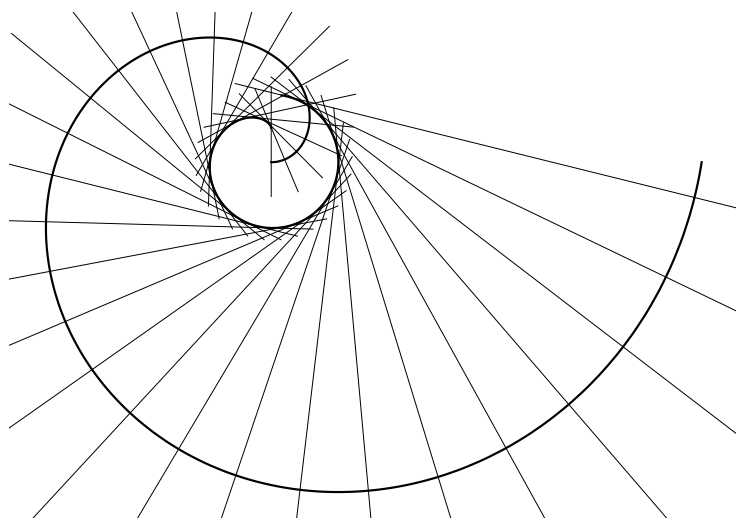


КОМПЬЮТЕРНАЯ ГЕОМЕТРИЯ И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

(задачи)

П.И. ТРОШИН



Казанский федеральный университет
Казань 2015

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

П.И. Трошин

**КОМПЬЮТЕРНАЯ ГЕОМЕТРИЯ
И
ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ**

(задачи)

Учебно-методическое пособие

Казань — 2015

УДК 004.925.8

ББК 22.151

Т 70

Трошин П. И.

Компьютерная геометрия и геометрическое моделирование (задачи).

Учебно-методическое пособие / П. И. Трошин. — Казань: Казанский федеральный университет, 2015. — 55 с.

Учебно-методическое пособие предназначено для проведения занятий со студентами Института математики и механики им. Н. И. Лобачевского Казанского (Приволжского) федерального университета.

©Казанский федеральный университет, 2015

©Трошин П. И., 2015

Оглавление

Введение	5
1 Кривые	7
1.1 Разные способы задания кривых	7
1.2 Кардиоида	9
2 Дифференциальная теория кривых	10
2.1 Репер Френе, сопровождающий трехгранник	10
2.2 Натуральные уравнения кривой: восстановление кривой по ее кривизне и кручению	11
2.3 Соприкасающаяся сфера	12
2.4 Эволюта и эвольвента	14
2.5 Локальное строение кривой	15
2.6 Параллельные кривые	15
2.7 Огибающие	16
2.8 Каустики	17
2.9 Кривая погони	18
3 Поверхности	19
3.1 Параметризованные и неявно заданные поверхности	19
3.2 Трубочатая, циклическая и каналовая поверхности	22
3.3 Тор	23
3.4 Лист Мёбиуса	25
3.5 Гиперболический параболоид как линейчатая поверхность	26
3.6 Однополостный гиперболоид как линейчатая поверхность	28
3.7 Стереографическая проекция	30
4 Дифференциальная теория поверхностей	31
4.1 Натуральный репер, кривые на поверхности	31
4.2 Кривизны, главные направления	31
4.3 Векторное поле, абсолютная производная, абсолютно \parallel перенос, геодезические	32
5 Кривые и поверхности второго порядка	34

6	Основы программирования: фракталы	35
6.1	Множества Жюлиа и Мандельброта для квадратичного полинома	35
	Заполненное множество Жюлиа $J(f_c)$	35
	Множество Жюлиа $J(f_c)$	36
	Множество Мандельброта M	38
6.2	Итерации на плоскости. Игра в хаос	41
6.3	Итерации на плоскости. Хаотические аттракторы	44
7	Основы пакета Матнематика для построения изображений	45
7.1	Графика	45
	Графические примитивы	45
	Построение графиков	46
	Совмещение изображений	47
	Экспорт и импорт графических объектов	48
7.2	Функции, модули, процедуры	49
7.3	Подстановки	49
7.4	Решение уравнений	50
7.5	Дифференцирование вектор-функций	51
7.6	Манипулирование, динамическое представление	52
7.7	Векторная алгебра	52
7.8	Операторы программирования	53
7.9	Хитрости	53
	Список литературы	54

Введение

Чем абстрактнее истина, которую ты хочешь преподать, тем сильнее ты должен обольстить ее еще и чувства.

Фридрих Ницше

Какой толк в книге без картинок или разговоров?

Алиса

Построение геометрических объектов на компьютере имеет большое значение как для создания иллюстративного материала, так и для большего понимания свойств этих объектов и принципов геометрии, лежащей в их основе. Кроме того, при этом развиваются навыки программирования.

Данное пособие содержит задачи на построение и изучение свойств различных геометрических объектов, изучаемых в курсах аналитической, дифференциальной и фрактальной геометрии. В последней главе кратко описаны основные навыки владения системой Матнематика, относящиеся к задачам представления графики.

Не оспаривая удобство использования и интерфейс различного программного обеспечения компьютерной математики, в данном пособии было решено в качестве основного средства моделирования использовать систему компьютерной алгебры и символьных вычислений Матнематика. Среди преимуществ этой системы — удобство работы, поддержка символьных вычислений, широкие графические возможности, удобная интерактивная справка.

Ввиду сделанного выбора данное пособие может использоваться при изучении применения Матнематика в курсе высшей математики. Однако все предложенные задачи читатель легко сможет реализовать на других языках математических вычислений.



Данное учебно-методическое пособие предназначено для проведения практических занятий студентов-математиков по курсам, изучающим программирование, дифференциальную и фрактальную геометрию.

Решая каждую из предложенных задач, читателю рекомендуется:

- решать задачу в максимально общем виде, чтобы легко можно было менять входные данные и пересчитывать результат;
- представлять графическое решение и ответы в наиболее информативном виде:
 - выбирать для построения кривые и поверхности «общего положения»;
 - подбирать точку обзора, размеры объектов и область для отображения необходимым образом;
 - отмечать и выделять на рисунке все точки и линии, использовать векторы, менять цвет объектов, их прозрачность;
 - снабжать графические объекты подписями;
- решать задачу преимущественно в векторном виде, не переходя непосредственно к координатам;
- предварительно делать схематические рисунки на бумаге и выводить соответственные расчетные формулы;
- варьировать параметры, входящие в задачу, по возможности, используя динамическое представление.



Рекомендуемое распределение задач для студентов:

- задачи части 1.1 — всем;
- задачи части 1.2 — по одной на выбор (по вариантам);
- задачи части 2.1 — всем;
- темы 2.2–2.9 — по одной на выбор (по вариантам);
- задачи части 3.1 — всем;
- темы 3.2–3.7 — по одной на выбор (по вариантам);
- задачи части 4.1–4.2 — всем;
- задачи части 4.3 — по одной на выбор (по вариантам);
- задачи части 5 — по одной на выбор (по вариантам);
- задачи частей 6.1–6.3 — по одной-двум на выбор (по вариантам).

Кривые

1.1 Разные способы задания кривых

Задание

1.1.1. Нарисовать плоские кривые, заданные параметрически (a — параметр):

- $r(t) = \{t \sin t, t \cos t\}$;
- $$\begin{cases} x = 1 - a(2 \cos t - \cos 2t), \\ y = a(2 \sin t - \sin 2t) \end{cases}.$$

1.1.2. Нарисовать плоские кривые, заданные в полярной системе координат (a — параметр):

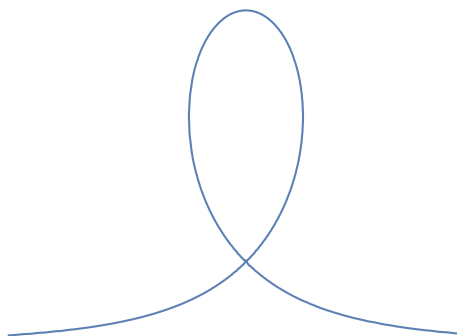
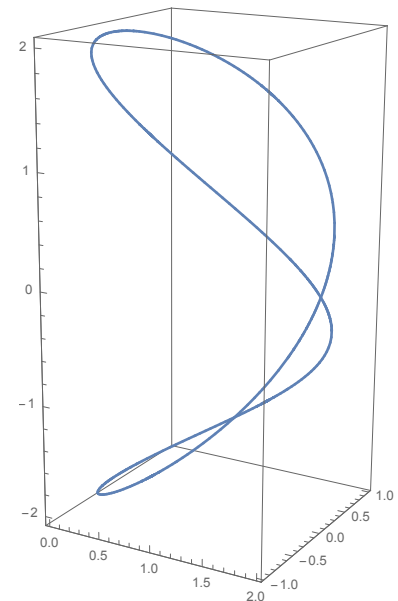
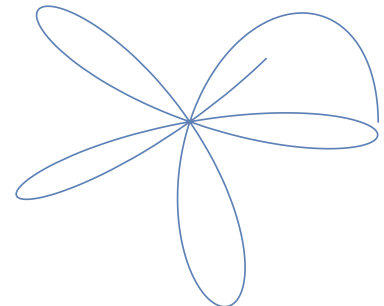
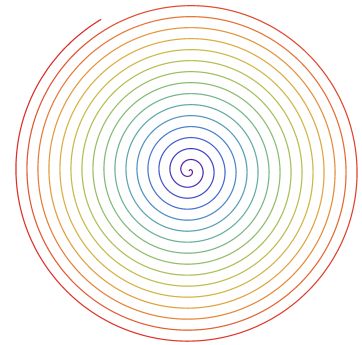
- $\rho = \cos(\phi^2)$;
- $\rho = 2a(1 + \cos \phi)$.

1.1.3. Нарисовать плоские кривые, заданные неявно (a — параметр):

- $x^3 + y^3 = 3axy$;
- $(x^2 + y^2 - 2ax)^2 = 4a^2(x^2 + y^2)$.

1.1.4. Нарисовать кривую, заданную пересечением двух поверхностей:

$$\begin{cases} x^2 + y^2 + z^2 = 4, \\ (x - 1)^2 + y^2 = 1. \end{cases}$$



1.1.5. Нарисовать семейство плоских кривых, являющихся решением системы дифференциальных уравнений

$$x'(t) = \sin(x(t)), \quad y'(t) = \cos(x(t)).$$

1.1.6. Нарисовать плоскую кривую, являющуюся решением системы дифференциальных уравнений

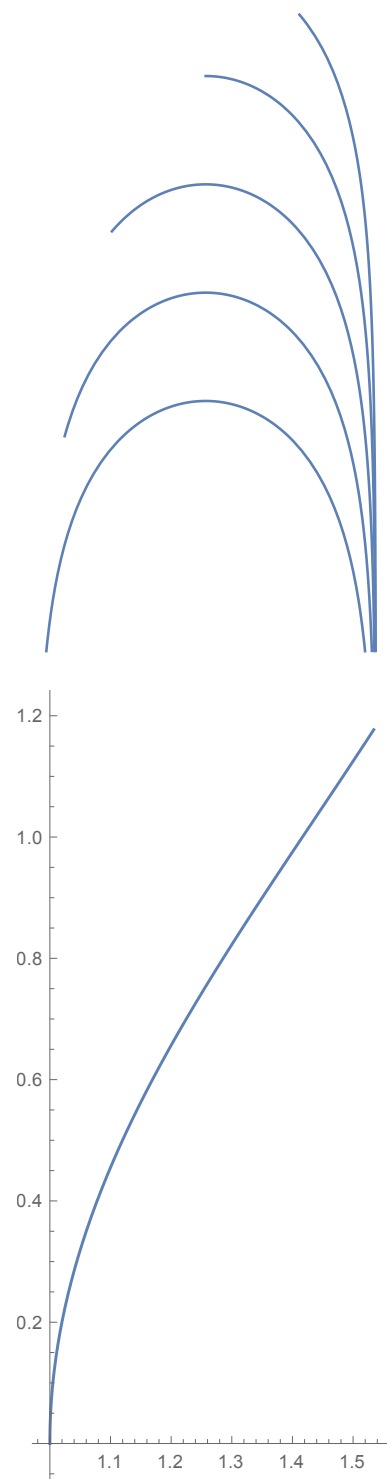
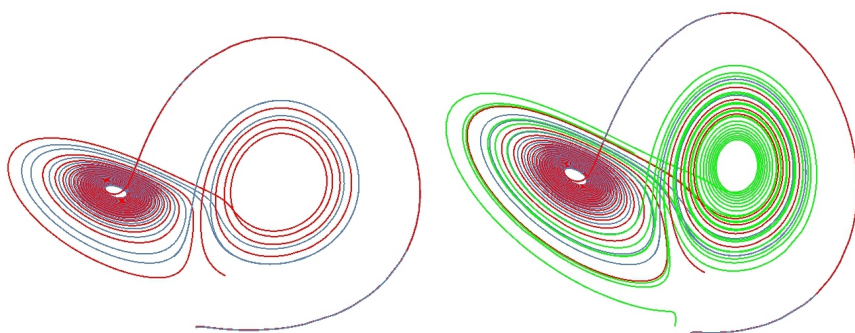
$$x'(t) = \sin(tx(t)), \quad y'(t) = \cos(x(t)),$$

проходящую через точку $\{x(0), y(0)\} = \{1, 0\}$.

1.1.7. Нарисовать пространственную кривую $r(t) = \{x(t), y(t), z(t)\}$, являющуюся решением системы дифференциальных уравнений Лоренца:

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(r - z) - y, \\ \dot{z} = xy - bz, \end{cases} \quad \sigma = 10, r = 28, b = 8/3.$$

1.1.8. На одном графике нарисовать три траектории системы Лоренца: две слегка отличающиеся начальными условиями и одну с большим отличием в начальных условиях. Что можно заметить?



1.2 Кардиоида

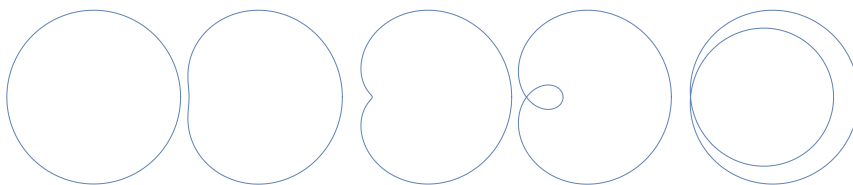
Задание

1.2.1. Получить кардиоиду как эпициклоиду при движении без скольжения одной окружности вдоль другой окружности того же радиуса a .

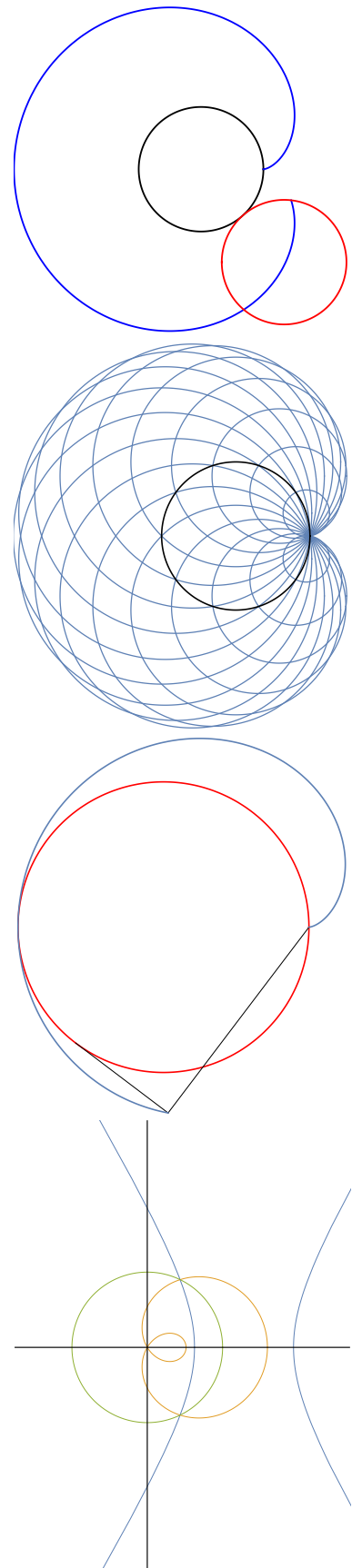
1.2.2. Получить кардиоиду как огибающую семейства окружностей с центрами на заданной окружности радиуса a , и проходящих через фиксированную точку на этой окружности. Для этого просто постройте это семейство¹.

1.2.3. Получить кардиоиду как «педальную кривую»: её точки являются проекциями фиксированной точки на данной окружности на касательные прямые к этой окружности.

1.2.4. Варьируя соотношение параметров, получить кардиоиду как промежуточное положение улитки Паскаля $r = b + a \cos \theta$ при $a = b$.



1.2.5. Вместе с улиткой Паскаля изобразить ее образ при инверсии $\rho(r, \phi) = 1/r$ относительно окружности с центром в вершине улитки (в точке самопересечения). Какие кривые получаются при разных параметрах a и b ?



¹Кардиоиду можно получить и как огибающую однопараметрического семейства прямых — каустику, см. с. 17.

Дифференциальная теория кривых

2.1 Репер Френе, сопровождающий трехгранник

Задание

2.1.1. Задать и построить бигулярную параметризованную кривую $\mathbf{r}(t)$ (рассмотрите замкнутые и незамкнутые кривые). Выбрать на ней точку M .

2.1.2. В точке M на кривой построить репер Френе. На одном рисунке отобразить кривую, точку на ней и исходящие из нее векторы: орты **касательной**, **главной нормали**, **бинормали**.

2.1.3. В точке M отобразить вектор ускорения.

2.1.4. Найти кривизну и кручение кривой в точке M .

2.1.5. Подсчитать длину дуги кривой от некоторой точки M_0 на кривой до точки M .

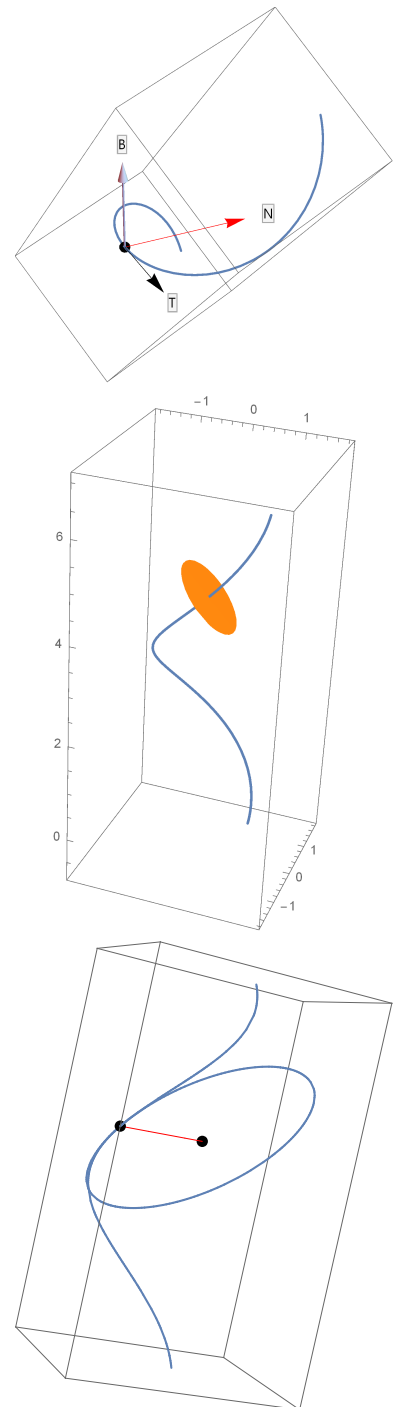
2.1.6. Изобразить центр кривизны кривой в точке M , соединив его с этой точкой.

2.1.7. Изобразить кривую, точку M и соприкасающуюся окружность в этой точке.

2.1.8. Построить плоскости трехгранника Френе. Отобразить кривую, точку на ней и следующие плоскости в этой точке (каждую на отдельном рисунке): **соприкасающуюся**, **нормальную**, **спрямляющую**.

2.1.9. Динамическая презентация:

точка $M(t)$ движется вдоль кривой $\mathbf{r}(t)$ вместе с объектами, построенными выше, выводится кривизна и кручение.



ПОДСКАЗКА: воспользуйтесь опцией `PlotRange->All`.

2.2 Натуральные уравнения кривой: восстановление кривой по ее кривизне и кручению

Известно, что своими натуральными уравнениями кривая задается единственным образом с точностью до движения в пространстве.

Задание

2.2.1. Задать кривизну и кручение скалярными функциями $k = k(s)$ и $\kappa = \kappa(s)$.

2.2.2. Решив систему дифференциальных уравнений Френе-Серре

$$\begin{cases} \vec{\tau}'(s) = k(s)\vec{\nu}(s), \\ \vec{\nu}'(s) = -k(s)\vec{\tau}(s) + \kappa(s)\vec{\beta}(s), \\ \vec{\beta}'(s) = -\kappa(s)\vec{\nu}(s), \end{cases}$$

найти уравнения кривой и построить ее.

Сколько скалярных неизвестных в этой системе? Все ли их нужно искать?

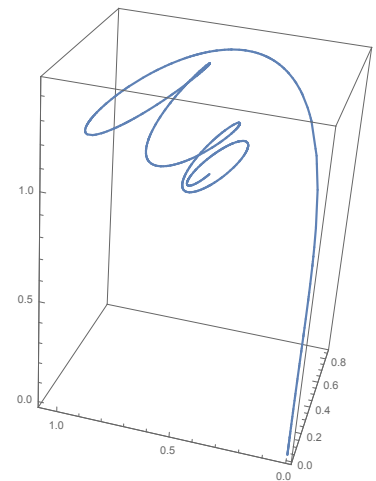
2.2.3. Какую роль играют константы интегрирования?

2.2.4. Вывести более простое дифференциальное уравнение для частного случая плоской кривой. Нарисуйте кривую.

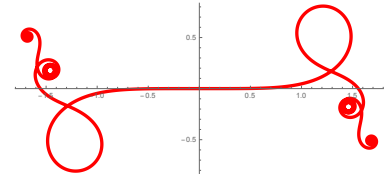
2.2.5. Воспользуйтесь также следующей системой дифференциальных уравнений для нахождения уравнения плоской кривой по ее кручению:

$$\begin{cases} x'(s) = \cos \theta(s), \\ y'(s) = \sin \theta(s), \\ \theta'(s) = k(s). \end{cases}$$

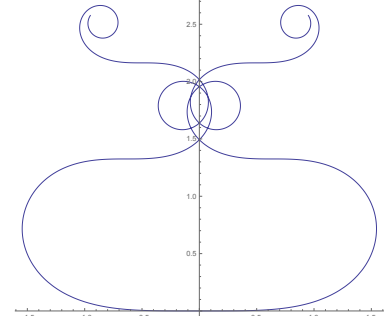
2.2.6. Что происходит, если $k < 0$?



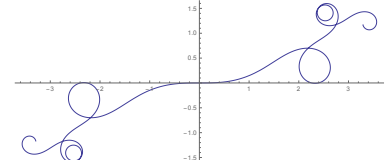
$$k = s, \kappa = \sin s$$



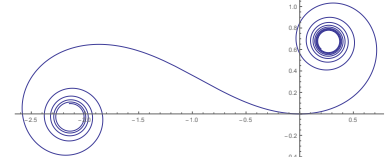
$$k = s^2 \sin s$$



$$k = s \sin s$$



$$k = s \cos s$$



$$k = s + 1$$

2.3 Соприкасающаяся сфера

Зафиксируем регулярную точку на кривой \mathbf{t}_0 . Рассмотрим произвольную точку \mathbf{t}_1 на кривой в окрестности \mathbf{t}_0 . Эти две точки определяют прямую. При стремлении $\mathbf{t}_1 \rightarrow \mathbf{t}_0$ мы получаем касательную прямую.

Если рассмотреть две точки $\mathbf{t}_1, \mathbf{t}_2$ в окрестности точки \mathbf{t}_0 , то они определяют окружность. При стремлении $\mathbf{t}_1, \mathbf{t}_2 \rightarrow \mathbf{t}_0$ окружность будет соприкасающейся.

Если рассмотреть три точки $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ в окрестности точки \mathbf{t}_0 , то они определяют сферу. В пределе при стремлении $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3 \rightarrow \mathbf{t}_0$ сфера называется *соприкасающейся*.

Соприкасающаяся сфера имеет 3-й порядок касания с кривой в соответствующей точке кривой; ее центр лежит на прямой, проходящей через центр кривизны параллельно бинормали. Все остальные сферы с центрами на этой прямой, касающиеся кривой, имеют касание порядка 2. все остальные сферы, касающиеся кривой в данной точке, имеют касание порядка 1.

Соприкасающаяся сфера в пересечении с соприкасающейся плоскостью дает соприкасающуюся окружность.

Задание

2.3.1. Построить соприкасающуюся окружность к выбранной вами кривой (в регулярной точке)

2.3.2. Построить соприкасающуюся сферу к этой кривой в этой точке. Её центр находится в точке

$$\mathbf{r}_C(s) = \mathbf{r}(s) + R(s)\vec{\mathbf{v}}(s) + \frac{R'(s)}{\kappa(s)}\vec{\mathbf{\beta}}(s),$$

где s — наутуральный параметр кривой $\mathbf{r}(t)$, $\vec{\mathbf{v}}$ — орт главной нормали, $\vec{\mathbf{\beta}}$ — орт бинормали, R — радиус кривизны, а κ — кручение.

Как перейти от натурального параметра к произвольному? Заметьте, что

$$R'(s) = \frac{dR}{ds} = \frac{dR}{dt} \frac{dt}{ds} = \frac{dR}{dt} : \frac{ds}{dt} = \frac{dR}{dt} : |\mathbf{r}'(t)| = \frac{R'(t)}{|\mathbf{r}'(t)|}.$$

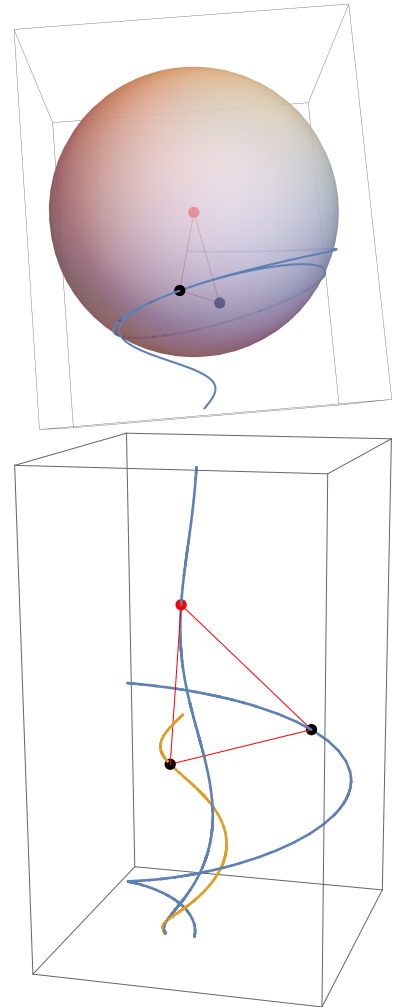
Радиус соприкасающейся сферы:

$$\tilde{R}(s) = \sqrt{R^2(s) + \left(\frac{R'(s)}{\kappa(s)}\right)^2}.$$

2.3.3. Динамическая презентация: соприкасающаяся сфера и окружность перемещаются вдоль кривой с изменением точки касания.

2.3.4. Вместе с кривой отобразить кривую центров соприкасающихся сфер и кривую центров кривизн. Для точки \mathbf{M} на кривой отобразить отрезок радиуса соприкасающейся окружности и отрезок, соединяющий центры соприкасающихся окружности и сферы (полярная прямая). Убедиться, что полярная прямая касается кривой центров сфер и является аналогом эволюты (в каком смысле?), а кривая центров кривизны в пространстве теряет свойство эволюты (какое?).

ПОДСКАЗКА: Используйте `Sqrt[r.r]` вместо `Norm[r]`. Почему это удобнее?

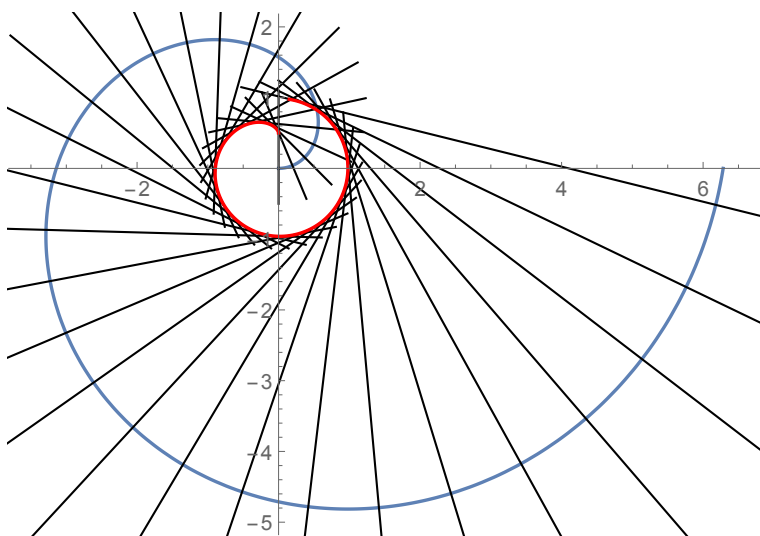


2.4 Эволюта и эвольвента

Задание

2.4.1. Построить плоскую кривую вместе с семейством ее нормалей.

Пронаблюдать огибающую этого семейства.



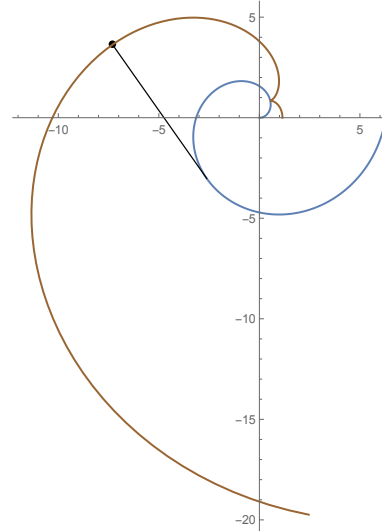
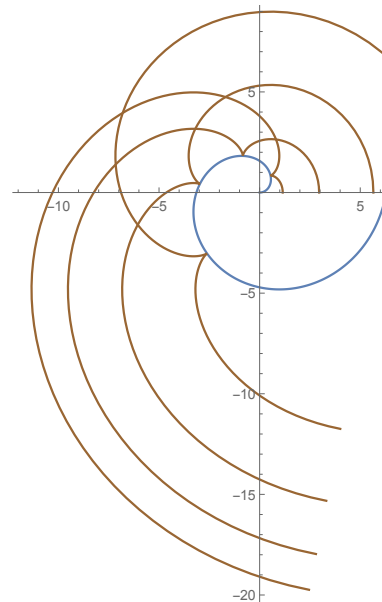
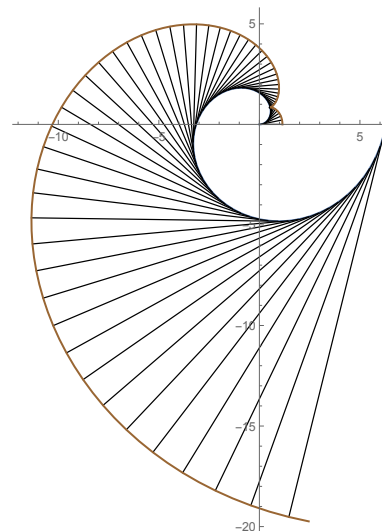
2.4.2. Построить эволюту данной кривой, как кривую центров кривизны.

Убедиться, что она совпадает с огибающей выше.

2.4.3. Для данной кривой построить ее эвольвенты при разных значениях параметра $s = 0, 1, 2, 3$. Почему получающиеся параллельные кривые отстоят при этом на разных расстояниях?

2.4.4. Динамическая презентация:

построить эвольвенту в виде «разматывающейся» с кривой нити (или катящейся прямой).



2.5 Локальное строение кривой

Задание

2.5.1. Задать и построить трирегулярную параметризованную кривую $\mathbf{r}(t)$.

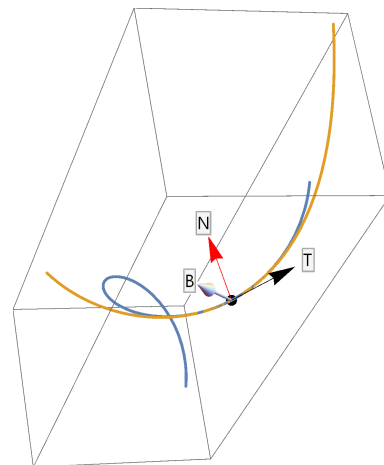
2.5.2. В точке M построить репер Френе: на одном рисунке отобразить кривую, точку на ней и исходящие из нее векторы репера Френе.

2.5.3. В точке M отобразить локальное приближение данной кривой до третьего порядка.

2.5.4. Динамическая презентация:

точка $M(t)$ движется вдоль кривой $\mathbf{r}(t)$ вместе с объектами, построенными выше.

2.5.5. Совпадают ли кривизна и кручение исходной кривой и ее локального приближения третьего порядка в точке M ?



2.6 Параллельные кривые

Задание

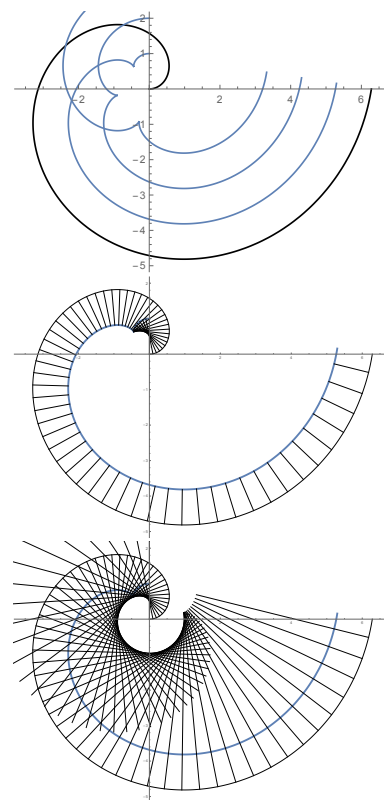
Для параметризованной кривой $\mathbf{r}(t)$ построить:

2.6.1. Семейство параллельных кривых с шагом 1.

2.6.2. Одну параллельную кривую и перпендикулярные отрезки между этими кривыми.

2.6.3. Одну параллельную кривую и лучи нормалей. Пронаблюдать, что параллельная кривая является эвольвентой для огибающей семейства нормалей (эволюты исходной кривой).

2.6.4. Построить параллельные кривые для параболы и эллипса.



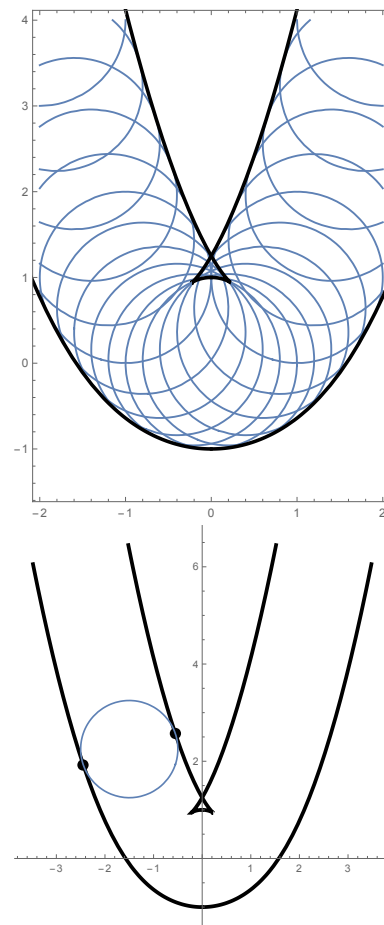
2.7 Огибающие

Задание

2.7.1. Построить огибающую (дискриминантную кривую) однопараметрического семейства плоских кривых вместе с изображением самого семейства.

2.7.2. Динамическая презентация:

отображается огибающая и одна из кривых семейства, а также точки касания. Кривая семейства движется вдоль огибающей.



2.8 Каустики

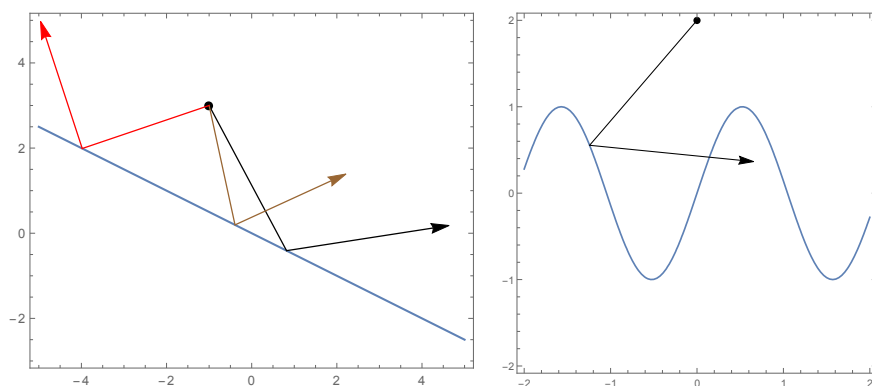
Каустика — огибающая однопараметрического семейства прямых линий.

Задание

2.8.1. Построить отраженный от прямой луч.

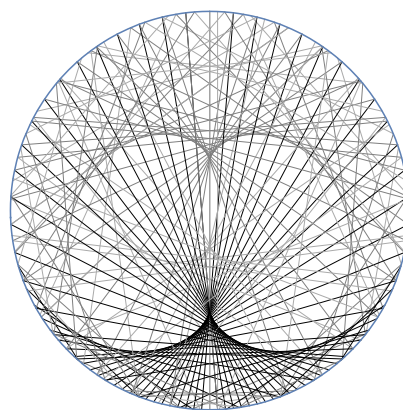
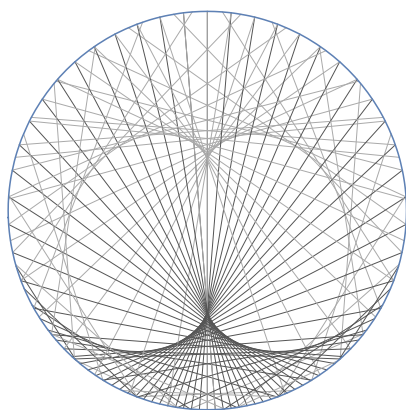
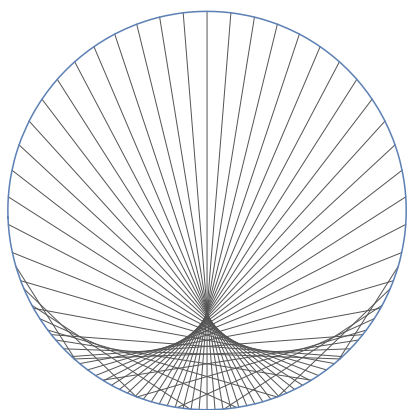
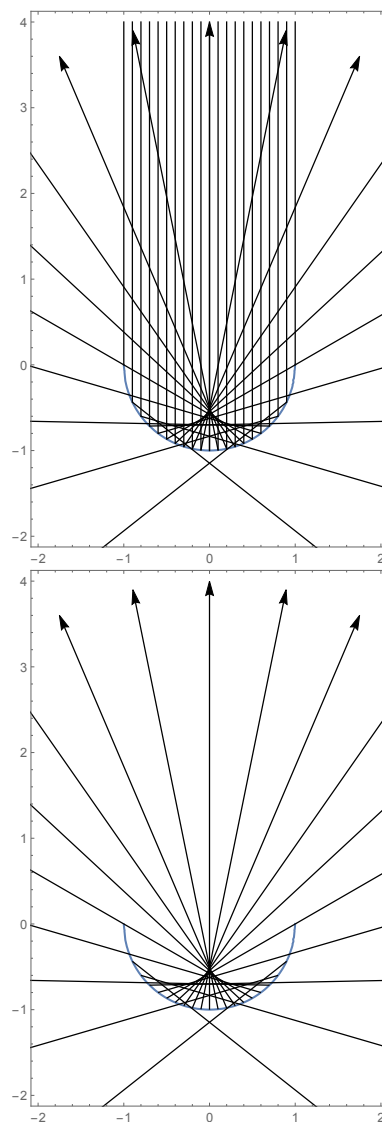
2.8.2. Построить отраженный от кривой луч (кривую можно задать неявным уравнением или параметрически).

Как задать поле «внешних» нормалей к кривой (не меняющих направление при прохождении точки перегиба)?



2.8.3. Построить семейство лучей, отраженных внутри от поверхности замкнутой кривой. Исходные лучи параллельны.

2.8.4. Построить каустику первого и второго порядков: лучи при отражении от поверхности снова отражаются.



2.9 Кривая погони

Кривая погони — траектория движения «жертвы» в следующей задаче.

Жертва движется на плоскости по заданной кривой $\mathbf{r}(t)$, $t \in [0, t_0]$ (иногда требуют, чтобы ее скорость была постоянной по модулю). Жертву преследует хищник. Условия погони: хищник стартует в момент времени $t = 0$ и движется с постоянной по модулю скоростью так, что его взгляд всегда направлен на жертву (он, в отличие от жертвы, постоянно корректирует свой путь).

Задание

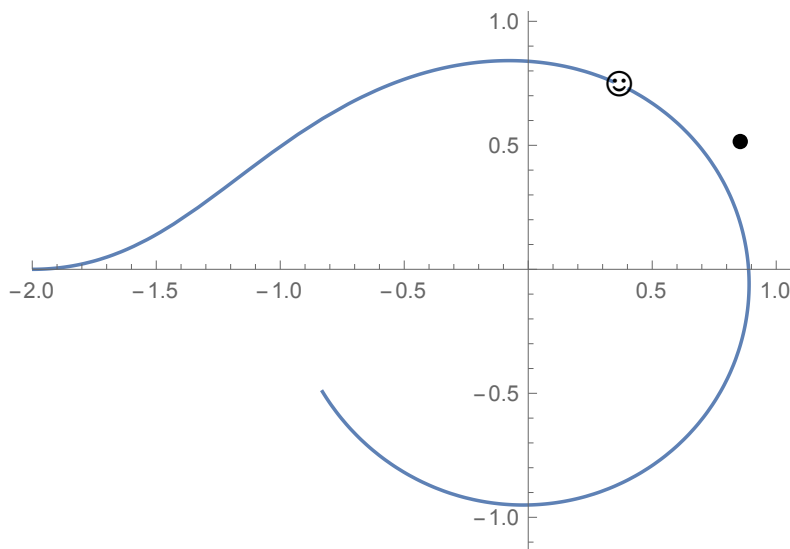
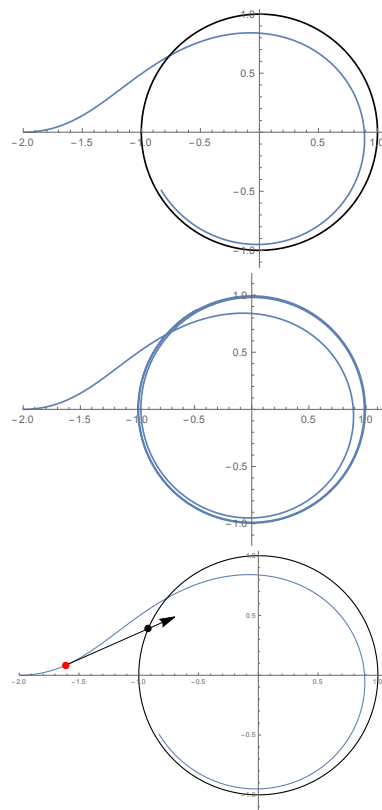
2.9.1. Вывести дифференциальное уравнение кривой погони.

2.9.2. Построить кривую погони вместе с траекторией жертвы.

2.9.3. В произвольной точке на траектории погони отметить хищника и отложить вектор его скорости, также отметить жертву в этот момент и убедиться, что вектор скорости действительно направлен на жертву.

2.9.4. Динамическая презентация: Отобразить кривую погони, жертву и хищника (отобразить процесс погони).

2.9.5. Догонит ли хищник жертву?



Поверхности

3.1 Параметризованные и неявно заданные поверхности

Для наглядности вместе с поверхностью отобразите ее **образующие, направляющие, оси вращения** и т.д.

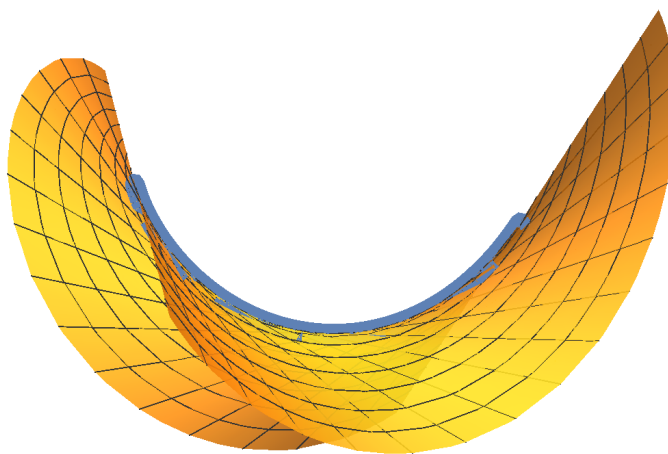
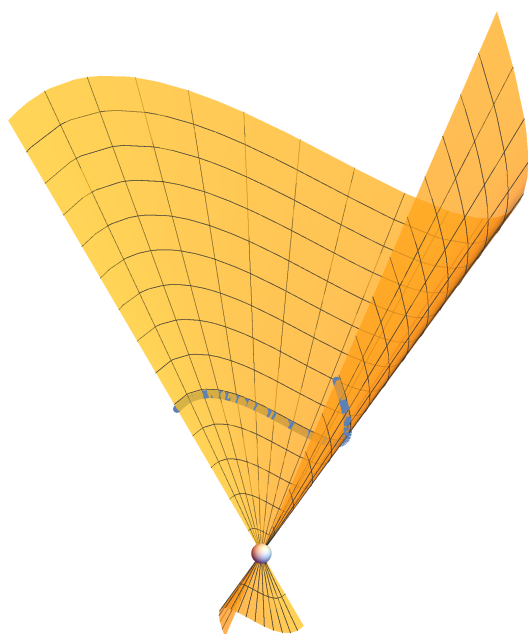
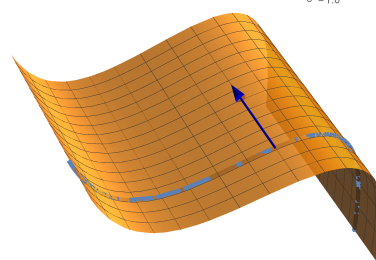
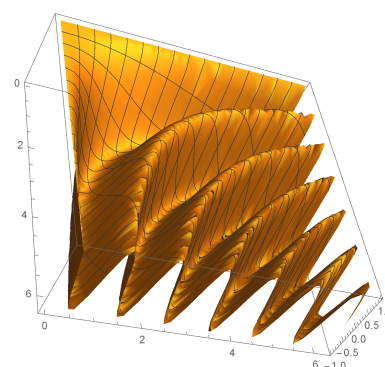
Задание

3.1.1. Построить параметризованную поверхность $\mathbf{r} = \mathbf{r}(u, v)$.

3.1.2. Построить цилиндрическую поверхность поверхность.

3.1.3. Построить коническую поверхность.

3.1.4. Построить торсовую поверхность: поверхность, образованную касательными к данной кривой.



3.1.5. Построить поверхность вращения.

Каким образом осуществить вращение вокруг произвольной оси?

3.1.6. Построить поверхность вращения прямой вокруг скрещивающейся с ней оси.

Как называется получающаяся поверхность?

3.1.7. Построить поверхность переноса.

В частности, перенос параболы вдоль другой параболы. Какие поверхности при этом получаются?

Динамическая презентация:

осуществить процесс параллельного переноса одной кривой вдоль другой.

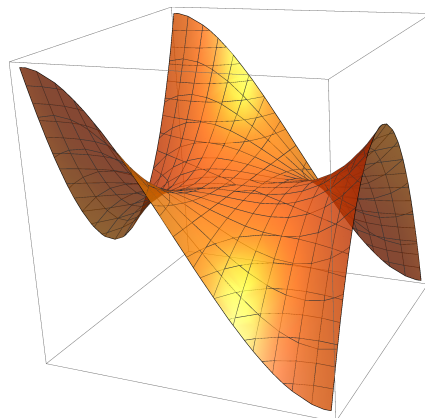
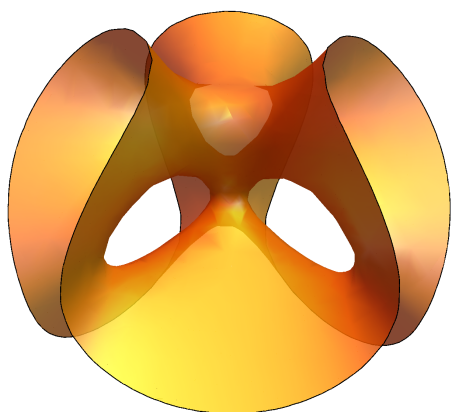
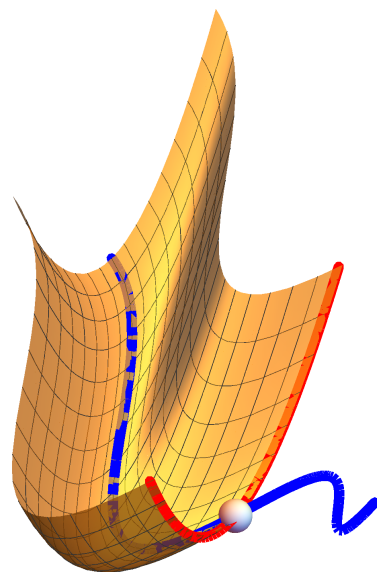
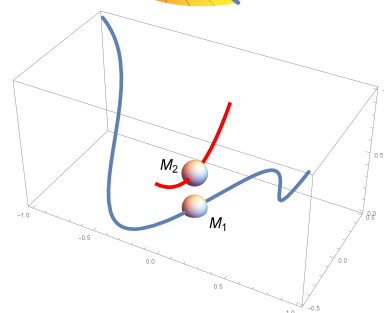
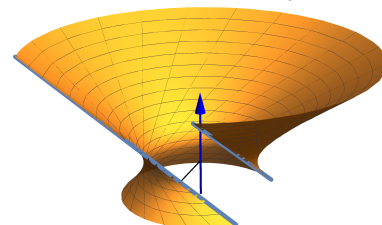
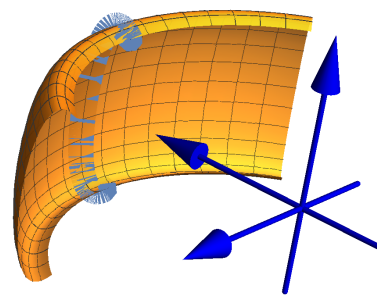
3.1.8. Построить поверхность, заданную неявно, например кубическую поверхность Клебша, содержащую ровно 27 прямых линий:

$$81(x^3 + y^3 + z^3) - 189(x^2y + x^2z + y^2x + y^2z + z^2x + z^2y) + 54xyz + 126(xy + xz + yz) - 9(x^2 + y^2 + z^2) - 9(x + y + z) + 1 = 0.$$

Или поверхность обезьяньего седла:

$$z = x^3 - 3xy^2$$

(в общем случае k спусков: $z(x, y) = \operatorname{Re}(x + iy)^k$).

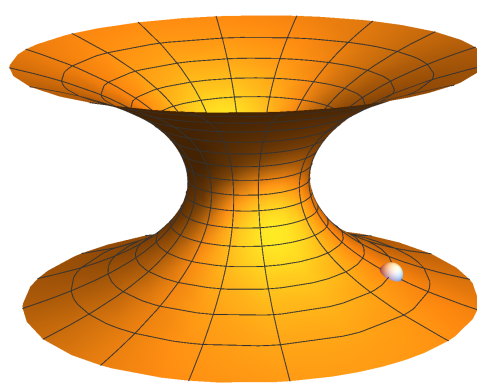
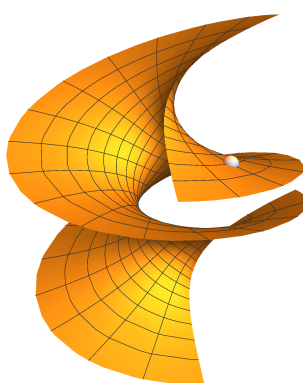
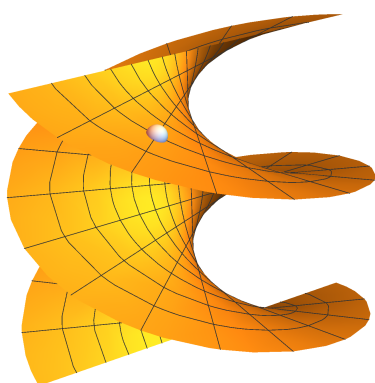
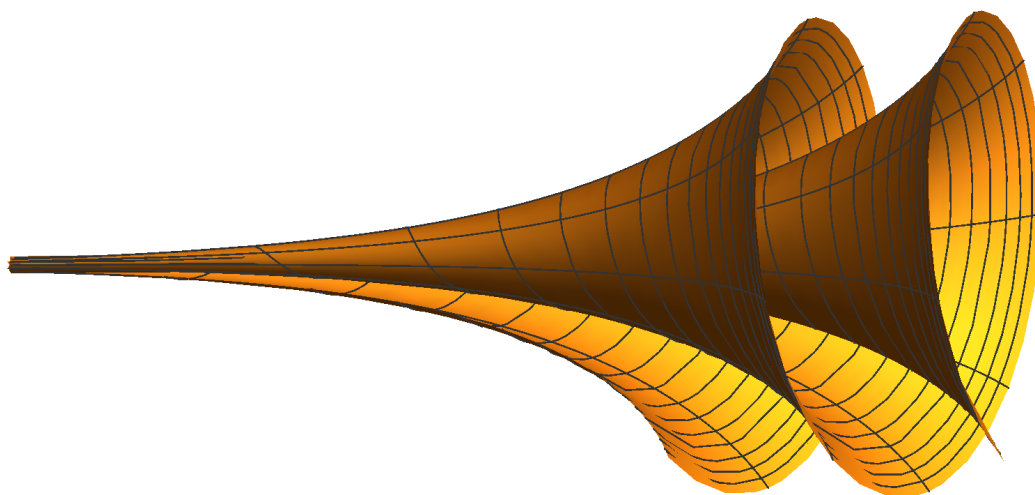
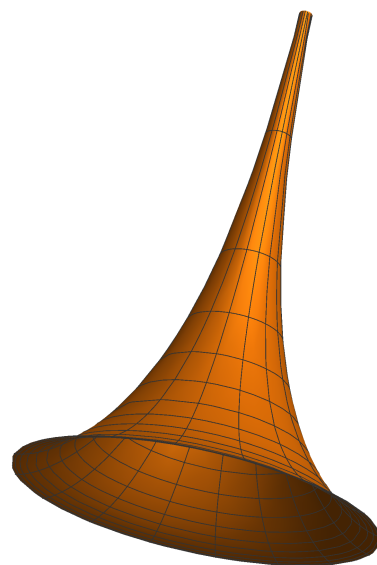


3.1.9. Построить псевдосферу.

3.1.10. Построить поверхность Дини постоянной кривизны $\mathcal{K} = -\frac{1}{a^2+b^2}$:

$$\mathbf{r}(u, v) = \begin{cases} x(u, v) = a \cos u \sin v, \\ y(u, v) = a \sin u \sin v, \\ z(u, v) = a(\cos v + \ln \operatorname{tg} \frac{v}{2}) + bu. \end{cases}$$

3.1.11. Динамическая презентация: построить деформацию геликоида в катеноид.



3.2 Трубчатая, циклическая и каналовая поверхности

Задание

3.2.1. Построить *трубчатую* поверхность: вдоль некоторой направляющей кривой движется образующая в форме окружности перпендикулярно кривой и с центром на этой кривой. Радиус окружности постоянный.

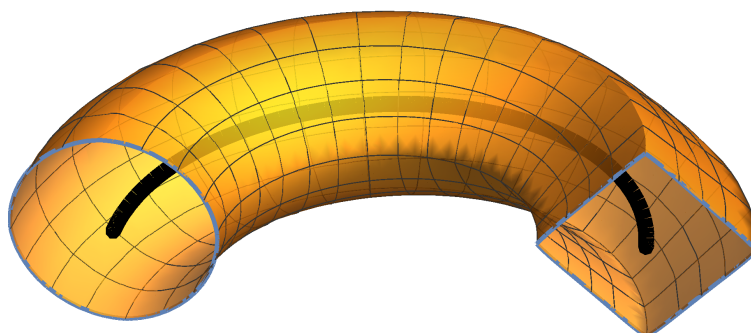
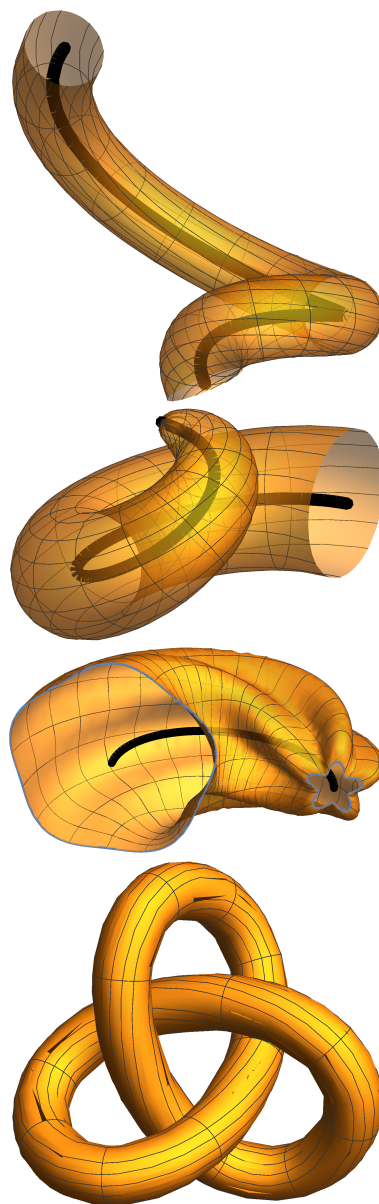
3.2.2. Построить *циклическую* поверхность: вдоль некоторой направляющей кривой движется образующая в форме окружности перпендикулярно кривой и с центром на этой кривой. Радиус окружности может быть переменным.

3.2.3. Построить *каналовую* поверхность: вдоль некоторой направляющей кривой движется образующая в форме плоской выпуклой замкнутой кривой, форма которой может непрерывно меняться, а плоскость — оставаться перпендикулярно направляющей кривой или образовывать с ней постоянный угол или быть параллельной некоторой фиксированной плоскости.

3.2.4. Построить трубчатую поверхность с образующей — кривой, заузленной в пространстве, например, трехлистник (его можно получить в виде замкнутой линии на поверхности тора):

$$\mathbf{r}(t) = \{\sin t + 2 \sin 2t, \cos t - 2 \cos 2t, -\sin 3t\}.$$

Похожий результат дает команда `KnotData["Trefoil"]`.



3.3 Тор

Тор $T^2 = S^1 \times S^1$ является примером поверхности вращения, а также — трубчатой поверхности. Тор получается вращением окружности вокруг оси, находящейся в плоскости окружности, но не пересекающей окружность. Тор можно получить как трубчатую поверхность с направляющей окружностью. С точки зрения алгебраической топологии тор — фактор множество $\mathbb{Z}_{2\pi} \times \mathbb{Z}_{2\pi}$.

Задание

3.3.1. Построить тор.

3.3.2. Построить семейство кривых на торе, пересекающих образующие под постоянным углом:

$$u(t) = mt, \quad v(t) = nt, \quad m, n, t \in \mathbb{R}.$$

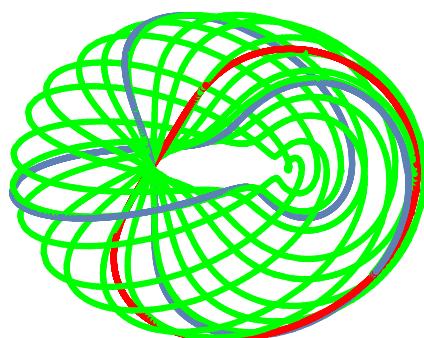
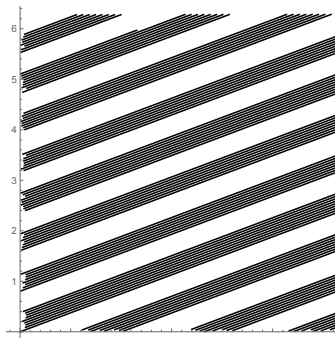
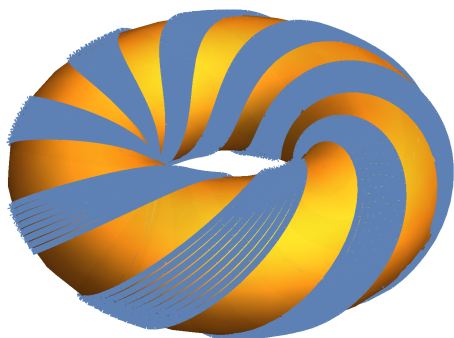
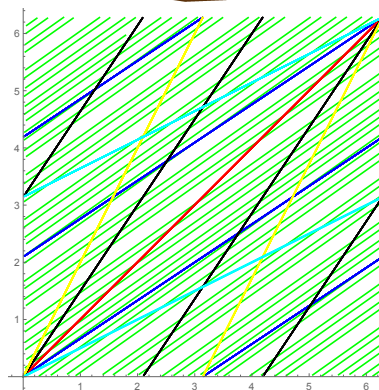
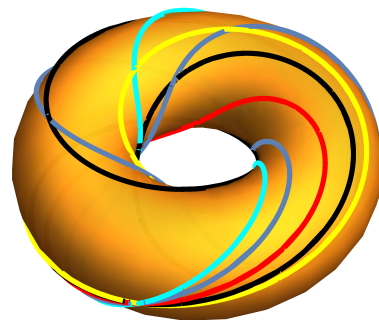
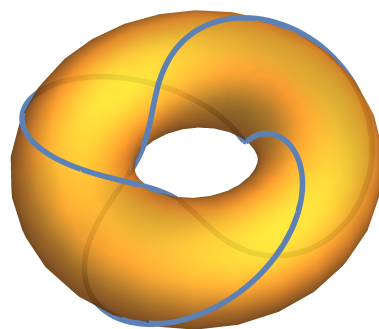
Заметить, при каких значениях угла получаются замкнутые кривые. Что происходит в случае незамкнутых обмоток тора?

Если $m, n \in \mathbb{Z}$, какую геометрическую интерпретацию можно дать этим числам?

3.3.3. Построить модель тора и кривых на нем в квадрате $[0, 2\pi] \times [0, 2\pi]$. Чем удобнее такая модель (обратите внимание на точки пересечения линий)?

3.3.4. Вывести из параметрических уравнений тора его неявное уравнение:

$$(x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) = 0.$$



Через любую точку тора можно провести четыре окружности, лежащие на нем: *меридиан*, *параллель* и две *окружности Вилларсо*.

3.3.5. Изобразить все четыре окружности, проходящие через одну точку на торе.

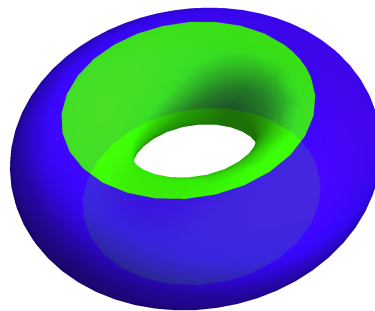
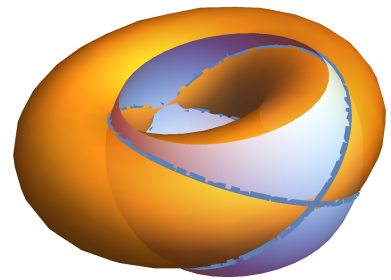
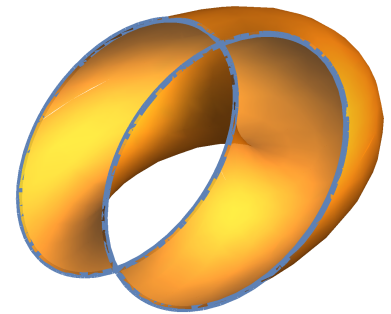
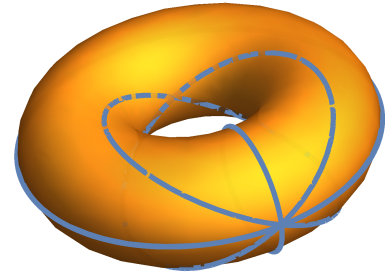
3.3.6. Построить окружности Вилларсо — сечения тора бикасательной плоскостью, проходящей через его центр.

3.3.7. Реализовать окружности Вилларсо как сечение тора сферой.

3.3.8. Зная уравнение одной окружности Вилларсо, получить уравнения любой другой:

$$\mathbf{r}(t) = \left\{ 2 \cos t + 1, \sqrt{R^2 - r^2} \sin t, r \sin t \right\}$$

3.3.9. Найти полную кривизну и главные кривизны тора. Найти асимптотические линии. Найти эллиптические, параболические и гиперболические точки и отметить их на торе.



3.4 Лист Мёбиуса

Лист (лента) Мёбиуса является примером линейчатой поверхности. Ее направляющей является окружность, а поле направляющих векторов прямолинейных образующих поворачивается в перпендикулярной к окружности плоскости на π .

Задание

3.4.1. Построить ленту Мёбиуса.

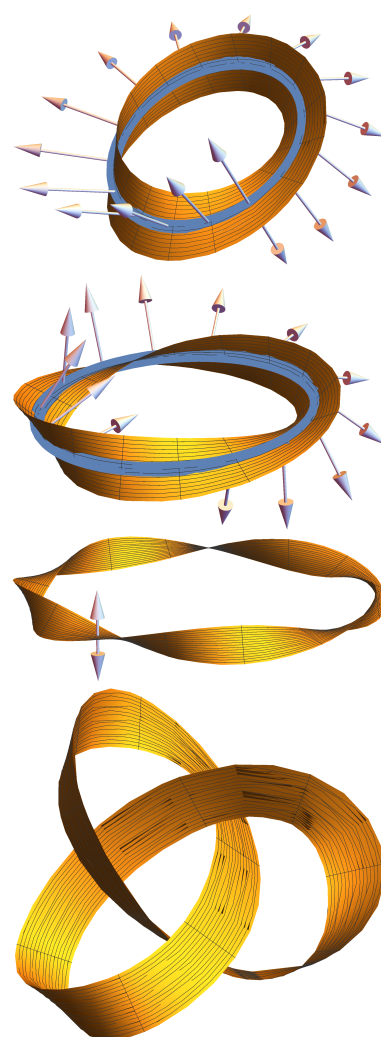
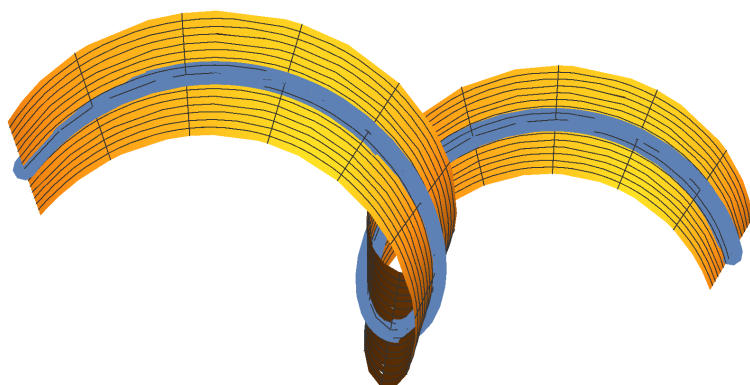
3.4.2. Построить поле ортов нормали к поверхности вдоль направляющей. Тем самым убедиться в неориентируемости поверхности.

3.4.3. Построить обобщение ленты Мёбиуса с n оборотами вокруг направляющей окружности.

3.4.4. Построить вариацию листа Мёбиуса с направляющей — трехлистником (или любой другой кривой). Вращение главной нормали направляющей кривой может мешать повороту ленты вокруг направляющей (можно рассмотреть винтовую линию).

3.4.5. Является ли полученная поверхность развертываемой?

3.4.6. Является ли полученная поверхность наложимой на плоскость?



3.5 Гиперболический параболоид как линейчатая поверхность

Каноническое уравнение гиперболического параболоида:

$$2z = \frac{x^2}{p} - \frac{y^2}{q} \quad (p, q > 0).$$

Гиперболический параболоид содержит два семейства прямолинейных образующих (каждое семейство задано пересечением плоскостей; параметр семейства $\lambda \in \mathbb{R}$):

$$(I) : \begin{cases} \lambda \left(\frac{x}{\sqrt{p}} + \frac{y}{\sqrt{q}} \right) = z, \\ \left(\frac{x}{\sqrt{p}} - \frac{y}{\sqrt{q}} \right) = 2\lambda. \end{cases} \quad (II) : \begin{cases} \lambda \left(\frac{x}{\sqrt{p}} - \frac{y}{\sqrt{q}} \right) = z, \\ \left(\frac{x}{\sqrt{p}} + \frac{y}{\sqrt{q}} \right) = 2\lambda. \end{cases}$$

Задание

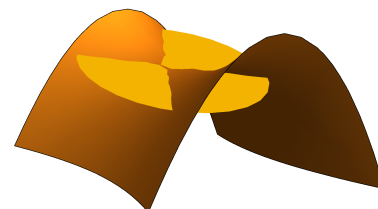
3.5.1. Построить Г.П. как поверхность, заданную неявно.

Построить касательную плоскость в выбранной точке на Г.П. Какие линии она высекает на поверхности?

3.5.2. Параметризовать Г.П. параметрами x и y и построить как параметризованную поверхность.

Параметризовать Г.П. параметрами, отличными от x и y (с помощью гиперболических тригонометрических функций).

Построить эту параметризованную поверхность и изучить, как выглядят координатные линии при такой параметризации.



3.5.3. Построить Г.П. как линейчатую поверхность: прямолинейная образующая движется вдоль направляющей кривой — параболы.

Параметризовать Г.П. двумя новыми параметрами: положением точки на образующей, и положением образующей на направляющей.

Динамическая презентация:

отобразить процесс образования поверхности.

Совместить два графика: параметризованной поверхности и поверхности, заданной неявно, убедиться, что они совпадают.

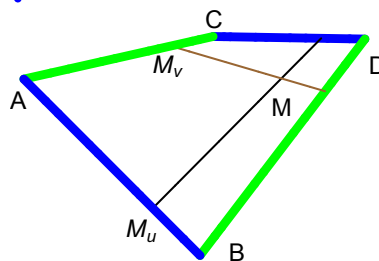
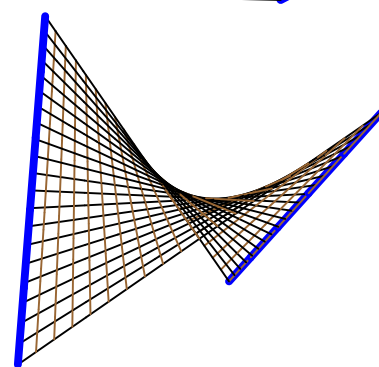
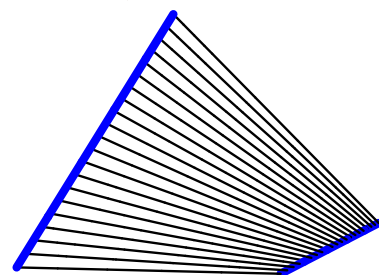
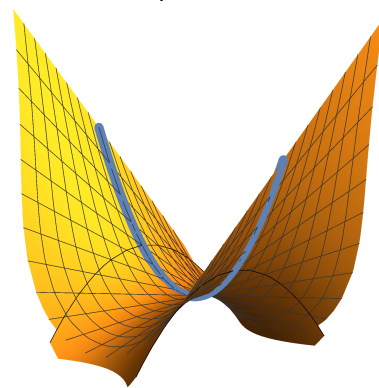
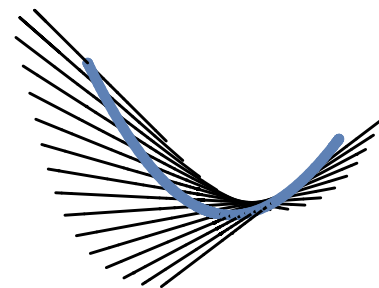
3.5.4. Построить часть поверхности Г.П. следующим образом: разбить два скрещивающихся отрезка AB и CD каждый на n точек и попарно соединить соответствующие точки одного отрезка с точками другого отрезка отрезками прямых.

Динамическая презентация:

показать, как меняется поверхность при изменении угла между скрещивающимися отрезками.

Построить также вторую серию отрезков, концы которых лежат на отрезках AC и BD . Заметить, что эти отрезки лежат на том же куске гиперboloида.

Параметризовать точки M на гиперboloиде новыми параметрами: простым отношением $u = (AB, M_u)$ и простым отношением $v = (AC, M_v)$.



3.6 Однополостный гиперболоид как линейчатая поверхность

Каноническое уравнение однополостного гиперболоида:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1 \quad (a, b, c > 0).$$

О.Г. содержит два семейства прямолинейных образующих (каждое семейство задано пересечением плоскостей; параметры семейства $\lambda, \mu \in \mathbb{R}, \lambda^2 + \mu^2 \neq 0$):

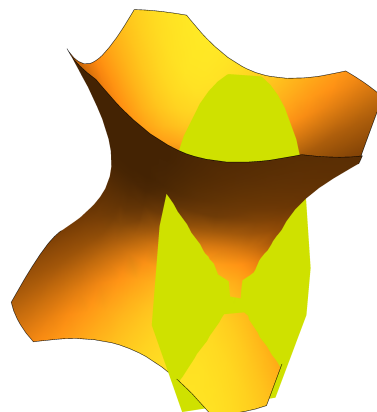
$$(I) : \begin{cases} \lambda \left(\frac{x}{a} + \frac{z}{c} \right) = \mu \left(1 + \frac{y}{b} \right), \\ \mu \left(\frac{x}{a} - \frac{z}{c} \right) = \lambda \left(1 - \frac{y}{b} \right). \end{cases} \quad (II) : \begin{cases} \lambda \left(\frac{x}{a} + \frac{z}{c} \right) = \mu \left(1 - \frac{y}{b} \right), \\ \mu \left(\frac{x}{a} - \frac{z}{c} \right) = \lambda \left(1 + \frac{y}{b} \right), \end{cases}$$

можно взять $\lambda = \cos t, \mu = \sin t, \quad t \in [0, 2\pi)$.

Задание

3.6.1. Построить О.Г. как поверхность, заданную неявно.

Построить касательную плоскость в выбранной точке на О.Г. Какие линии она высекает на поверхности?



3.6.2. Параметризовать О.Г. параметрами x и y и построить как параметризованную поверхность.

Параметризовать О.Г. параметрами, отличными от x и y (с помощью гиперболических тригонометрических функций).

Построить эту параметризованную поверхность и изучить, как выглядят координатные линии при такой параметризации.

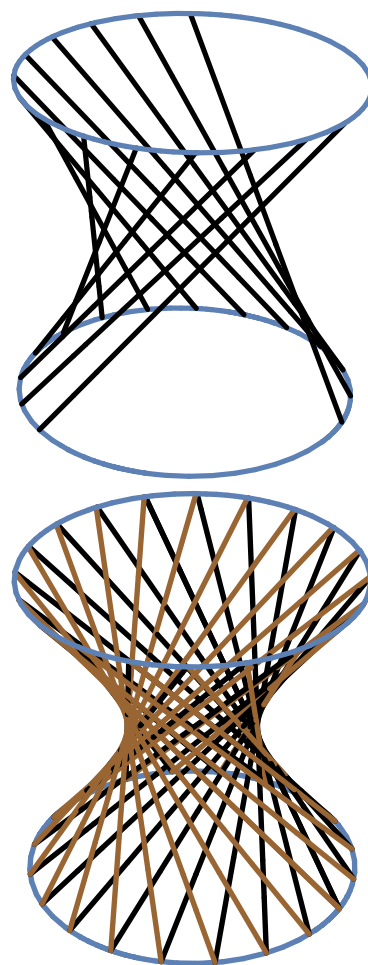
3.6.3. Построить О.Г. как линейчатую поверхность: прямолинейная образующая движется вдоль направляющей кривой — параболы.

Параметризовать О.Г. двумя новыми параметрами: положением точки на образующей, и положением образующей на направляющей.

Динамическая презентация:

отобразить процесс образования поверхности.

Совместить два графика: параметризованной поверхности и поверхности, заданной неявно; убедиться, что они совпадают.



3.7 Стереографическая проекция

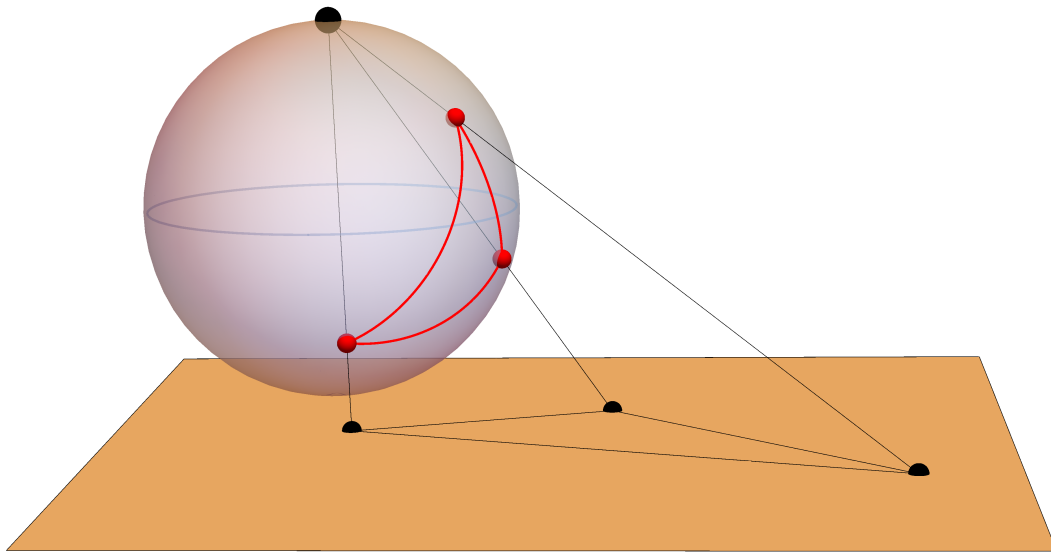
Задание

3.7.1. Задать операторы, осуществляющие стереографическую проекцию точек плоскости на сферу и обратно.

3.7.2. Нарисовать треугольник на плоскости и его образ на сфере.

3.7.3. Нарисовать кривую на сфере и ее образ на плоскости.

3.7.4. Каким свойством обладает стереографическая проекция? Во что переходят прямые линии и окружности на плоскости?



Дифференциальная теория поверхностей

4.1 Натуральный репер, кривые на поверхности

Задание

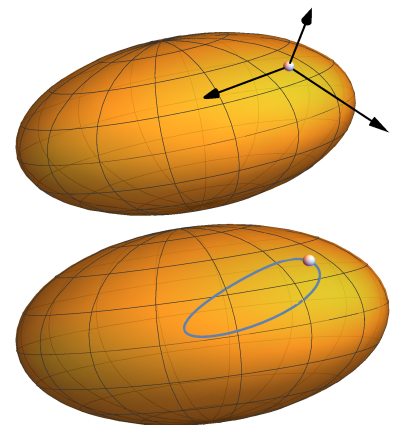
4.1.1. Построить параметризованную поверхность $\mathbf{r} = \mathbf{r}(u, v)$.

4.1.2. Отметить на ней точку M и построить подвижный репер $\{M; \mathbf{r}_u(u, v), \mathbf{r}_v(u, v), \mathbf{n}(u, v)\}$. Почему его векторы касаются сетки линий, построенной на поверхности программой МАТЕМАТИКА?

4.1.3. Написать оператор, находящий внутренние координаты касательного вектора по его пространственным координатам, и оператор, совершающий противоположную работу. Проверить правильность.

4.1.4. Через точку $M(u_0, v_0)$ провести произвольную кривую $u = u(\phi), v = v(\phi)$ на поверхности, например, образ окружности:

$$u(\phi) = r(\cos(\phi) - 1) + u_0; \quad v(\phi) = r \sin(\phi) + v_0.$$



4.2 Кривизны, главные направления

Задание

4.2.1. Подсчитать и вывести матрицы первой и второй квадратичных форм поверхности в точке M .

4.2.2. Подсчитать полную \mathcal{K} и среднюю $2\mathcal{H}$ кривизны поверхности в точке M с помощью матриц первой и второй квадратичной формы.

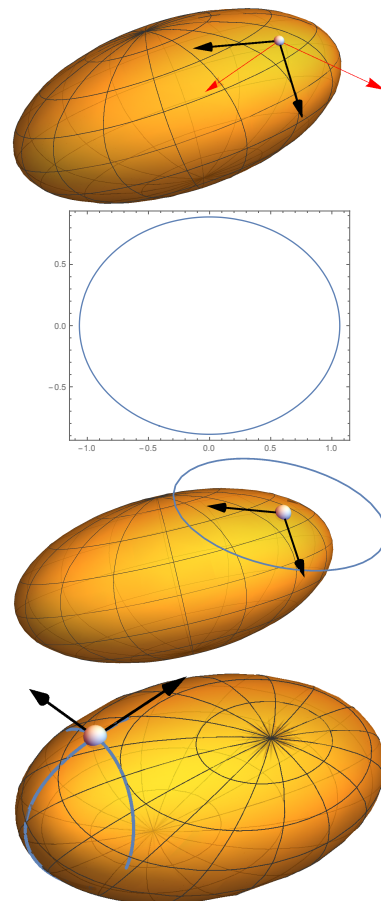
4.2.3. Найти нормальную кривизну поверхности в точке M в направлении вектора $\mathbf{a} = \{1, 2\}$, а также нормальную кривизну построенной выше кривой в точке M .

4.2.4. Найти главные кривизны и главные направления поверхности в точке M . Отобразить орты векторов главных направлений вместе с подвижным репером; убедиться, что они в общем случае не совпадают.

4.2.5. Написать уравнение для индикатрисы Дюпена с помощью коэффициентов второй квадратичной формы. Ее график задается неявным уравнением в осях главных направлений. Изобразите ее.

4.2.6. Изобразить индикатрису Дюпена в касательной плоскости к поверхности в точке M (для этого представьте индикатрису как параметризованную кривую, используя ее определение и формулу Эйлера).

4.2.7. Построить линии кривизны в окрестности точки M , убедиться что они касаются главных направлений.



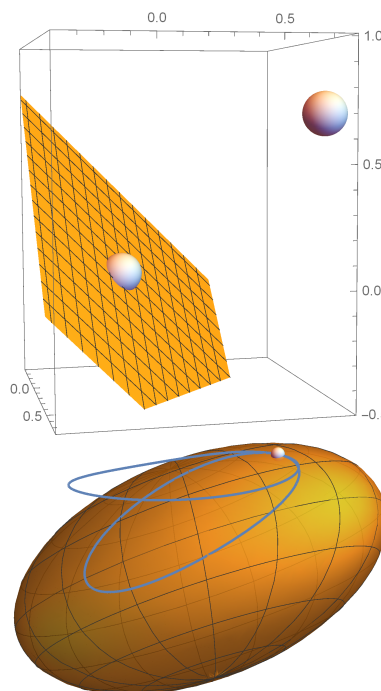
4.3 Векторное поле, абсолютная производная, абсолютно параллельный перенос, геодезические

Задание

4.3.1. Вывести аналитическую формулу для оператора ортогонального проектирования точки на произвольную плоскость (символьные вычисления в МАТНЕМАТИКА).

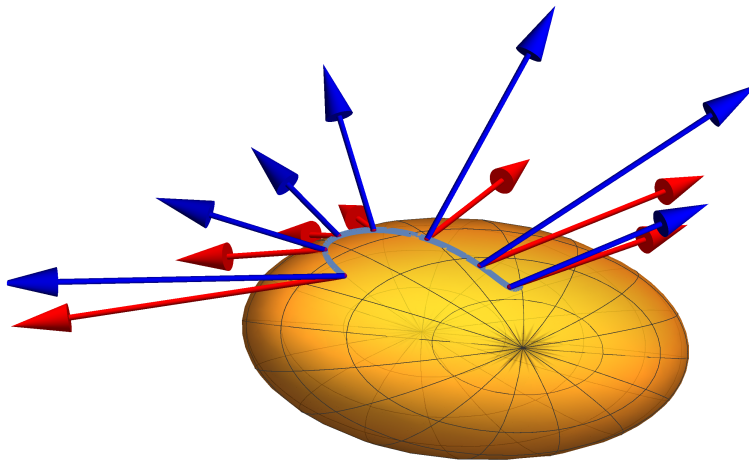
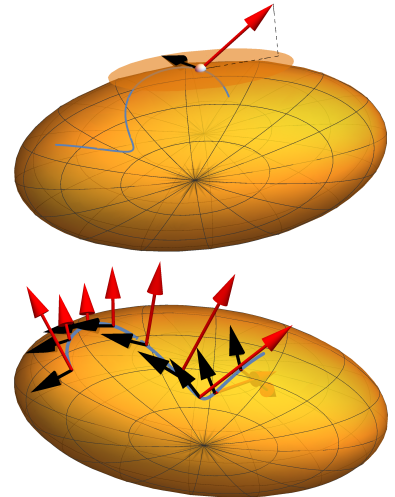
4.3.2. Построить проекцию кривой на поверхности на касательную плоскость к поверхности в некоторой точке этой кривой. Как связаны кривизна полученной плоской кривой, кривизна и геодезическая кривизна исходной кривой?

При проектировании кривой возникает проектирующий цилиндр. Как можно применить теорему Мёнье к данным кривым на нем? Изобразить.



4.3.3. Вдоль некоторой кривой на поверхности изобразить векторное поле и его производную. Убедиться, что в общем случае производная выходит из касательной плоскости и перестает быть векторным полем.

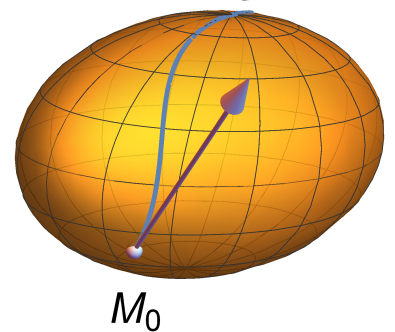
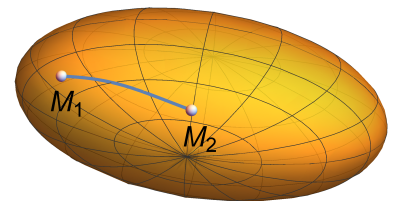
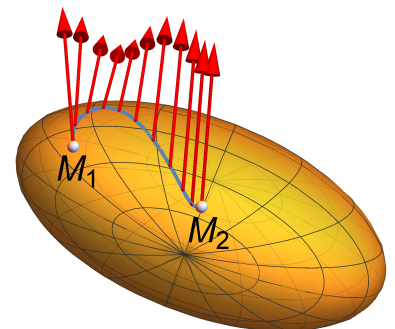
4.3.4. Вдоль некоторой кривой на поверхности изобразить векторное поле вместе с его абсолютной производной. Абсолютную производную следует задавать с помощью деривационных уравнений поверхности, а не оператором проектирования на плоскость.



4.3.5. Перенести вектор $\mathbf{a} = \{1, 2\}$ из точки $M_1(u_1, v_1)$ в точку $M_2(u_2, v_2)$, находящиеся на выбранной вами кривой, вдоль этой кривой.

4.3.6. Для данных точек $M_1(u_1, v_1)$ и $M_2(u_2, v_2)$ на поверхности найти и изобразить геодезическую линию, соединяющую эти точки.

4.3.7. Для данной точки $M_0(u_0, v_0)$ на поверхности найти и изобразить геодезическую линию, проходящую через эту точку в направлении $\mathbf{a} = \{1, 2\}$.



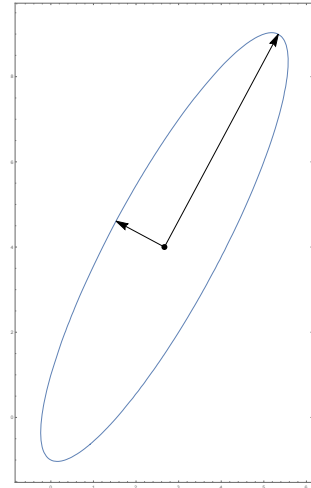
Кривые и поверхности второго порядка

Задание

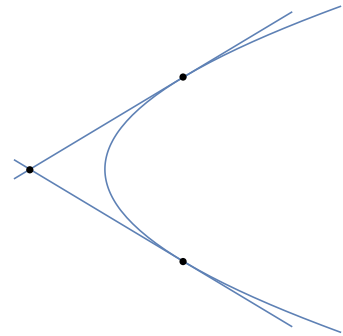
5.0.1. Кривая (второго порядка), заданная неявно.

Написать программу, которая позволяет по произвольному уравнению кривой второго порядка:

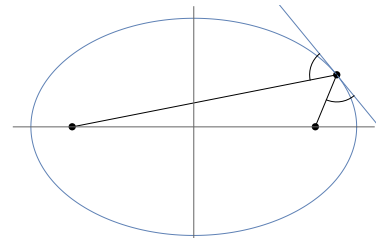
- Построить изображение кривой.
- Изобразить центр кривой (если есть) и оси ее канонической системы координат.
- Найти параметры канонического уравнения кривой (a , b , p).



5.0.2. Изобразить пересечение двух симметричных касательных к гиперболе. К чему стремится точка пересечения, когда точки касания неограниченно удаляются? Что происходит в случае параболы?



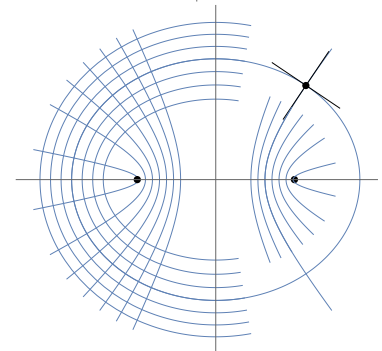
5.0.3. Иллюстрации оптических свойств эллипса, гиперболы, параболы, заданных в канонической системе координат.



5.0.4. Поверхность (второго порядка), заданная неявно.

Написать программу, которая позволяет по произвольному уравнению поверхности второго порядка:

- Построить изображение поверхности.
- Изобразить центр поверхности (если есть) и оси ее канонической системы координат.
- Найти параметры канонического уравнения поверхности (a , b , c , p , q).



ПОДСКАЗКА: Можно использовать команды `JordanDecomposition` или `Eigensystem`.

6.1 Итерации на комплексной плоскости. Множества Жюлиа и Мандельброта для квадратичного полинома

Рассмотрим некоторое отображение $f: \mathbb{C} \rightarrow \mathbb{C}$ (или равносильно $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$). Будем рассматривать динамику точек относительно этого отображения (орбиты точек):

$$z, f(z), f^2(z) = f(f(z)), \dots, f^k(z), \dots$$

Для каких-то точек z эта последовательность может сходиться к некоторой точке, для других — быть неограниченной, еще вариант — образовывать цикл (подумайте, какие другие варианты возможны).

Выделяя точки с общими свойствами их динамики, можно получить интересные геометрические объекты.

В дальнейшем мы будем рассматривать простейший квадратичный полином

$$f_c(z) = z^2 + c, \quad z, c \in \mathbb{C}, \quad c — \text{параметр}.$$

Использование комплексных чисел в данном случае сокращает обозначения (в декартовой системе координат это отображение запишется в виде $f(\{x, y\}) = \{x^2 - y^2 + x_c, 2xy + y_c\}$. Можно ли любое преобразование \mathbb{R}^2 записать в виде действия на комплексных числах?).

Заполненное множество Жюлиа $J(f_c)$

Заполненное множество Жюлиа $J(f_c)$ состоит из тех точек $z \in \mathbb{C}$, орбиты которых ограничены.

Для построения $J(f_c)$ нам помогут два свойства:

- $J(f_c) \subset D_R = \{|z| \leq R\}$, $R = 0.5 + \sqrt{0.5 + |c|}$,
- Если $\exists k : |f_c^k(z)| > R$, то $z \notin J(f_c)$.

Таким образом, нам достаточно разбить круг D_R на сетку из n узловых точек z_i , задаться предельным числом итераций $I \in \mathbb{N}$ и итерировать каждую точку z_i I раз. Если в процессе итерации точка выходит за пределы круга, мы ее не рисуем, если остается — считаем, что она лежит в множестве $J(f_c)$.

Алгоритм 6.1.1:

Заполненное множество Жюлиа для функции $f_c(z) = z^2 + c$.

Вход: $c \in \mathbb{C}$,

$n \in \mathbb{N}$,

$I \in \mathbb{N}$.


 комплексный параметр функции

 количество (плотность) испытываемых точек

 количество итераций для испытания точки

Выход: Изображение множества $J(f_c)$.


1: $path = \emptyset$;

 массив точек для отображения

2: $f_c(z) = z^2 + c$;

 инициализация итерируемой функции

3: $R = 0.5 + \sqrt{0.5 + |c|}$;

 радиус диска $D_R = \{z \mid |z| \leq R\}$, содержащего множество $J(f_c)$

4: для $z_i \in D_R$, $i = 1, \dots, n$

 разбиваем область D_R на сетку из n точек z_i

5: повторять I раз

6: $z_i = f_c(z_i)$;

 итерируем точку z_i

7: если $|z_i| > R$, то выход из цикла;

 точка z_i нам не подходит

8: если $|z_i| \leq R$, то $path = path \cup \{z_i\}$;

 z_i осталась в D_R после I итераций

9: Отобразить массив точек $path$.

Множество Жюлиа $J(f_c)$

Помимо заполненного множества Жюлиа интересной является также его граница $J(f_c) = \partial J(f_c)$, называемая множеством Жюлиа.

Существуют разные методы ее построения. Рассмотрим наиболее простой: метод обратных итераций. Нам понадобятся два свойства:

- Если z — наибольший по модулю из корней уравнения $f(z) = z$, то $z \in J(f_c)$;
- Если $z \in J(f_c)$, то орбита z относительно $f_c^{-1}(z) = \sqrt{z - c}$ плотна в $J(f_c)$ («плотно» ее заполняет). Более того, если $p^0 = z, p^{k+1} = f_c^{-1}(p^k)$, то $p^k \subset p^{k+1}$. Поэтому $J(f_c) \approx p^k$ при большом k .

Заметим, что $f_c^{-1}(z)$ принимает два значения $\sqrt{z-c}$ и $-\sqrt{z-c}$ (МАТЕМАТИКА выдает лишь одну из ветвей корня).

АЛГОРИТМ 6.1.2: Множество Жюлиа для функции $f_c = z^2 + c$.

МЕТОД ОБРАТНЫХ ИТЕРАЦИЙ.

Вход: $c \in \mathbb{C}$,

 комплексный параметр функции

$I \in \mathbb{N}$.

 количество итераций для испытания точки

Выход: Изображение множества $J(f_c)$.

1: $f_c(z) = z^2 + c$;

 итерируемая функция

2: $F_c(z) = +\sqrt{z-c} \cup -\sqrt{z-c}$;

 прообраз для f_c

3: Находим z_1, z_2 — корни уравнения $f_c(z) = z$;

4: Находим z — наибольший по модулю из этих корней;

5: $\text{path} = \{z\}$;

 массив точек для отображения, $z \in J(f_c)$

6: **повторять** I раз $\text{path} = F_c(\text{path})$;

7: Отобразить массив точек path .

Для простоты вычислений можно на каждом шаге итерации выбирать одну из ветвей корня случайным образом.

АЛГОРИТМ 6.1.3: Множество Жюлиа для функции $f_c = z^2 + c$.

РАНДОМИЗИРОВАННЫЙ МЕТОД ОБРАТНЫХ ИТЕРАЦИЙ.

Вход: $c \in \mathbb{C}$,

 комплексный параметр функции

$I \in \mathbb{N}$.

 количество итераций для испытания точки

Выход: Изображение множества $J(f_c)$.

1: $f_c(z) = z^2 + c$;

 итерируемая функция

2: $F_c^1(z) = +\sqrt{z-c}$;

 первая ветвь прообраза для f_c

3: $F_c^2(z) = -\sqrt{z-c}$;

 вторая ветвь прообраза для f_c

4: Находим z_1, z_2 — корни уравнения $f_c(z) = z$;

5: Находим z — наибольший по модулю из этих корней;

6: $\text{path} = \{z\}$;

 массив точек для отображения, $z \in J(f_c)$

7: **повторять** I раз

8: i — случайный выбор числа 1 или 2;

9: $z = F_c^i(z)$;

 строим случайную обратную орбиту исходной точки z

10: $\text{path} = \text{path} \cup \{z\}$;

11: Отобразить массив точек path .

Множество Мандельброта \mathcal{M}

Множество Мандельброта — это множество тех параметров $c \in \mathbb{C}$, при которых орбита нуля

$$0, f_c(0), f_c^2(0) = f_c(f_c(0)), \dots, f_c^k(0), \dots$$

ограничена: $\exists M_c \in \mathbb{R} \forall k \in \mathbb{N} \quad |f_c^k(0)| < M_c$. Для построения множества Мандельброта нам понадобятся следующие два свойства:

- $\mathcal{M} \subset D_2 = \{|c| \leq 2\}$;
- Если $\exists k : |f_c^k(0)| > 2$, то $c \notin \mathcal{M}$.

Таким образом, нам достаточно разбить круг D_2 на сетку из n узловых точек z_i , задаться предельным числом итераций $I \in \mathbb{N}$ и итерировать каждую точку z_i I раз. Если в процессе итерации точка выходит за пределы круга, мы ее не рисуем, если остается — считаем, что она лежит в множестве \mathcal{M} .

Алгоритм 6.1.4: Множество Мандельброта \mathcal{M} .

Вход: $c \in D_2 = \{|c| \leq 2\}$,

$n \in \mathbb{N}$,

I .


 комплексный параметр функции

 количество (плотность) испытываемых точек

 количество итераций для испытания точки

Выход: Изображение множества Мандельброта \mathcal{M} .

1: $path = \emptyset$;

 массив точек для отображения

2: $f(z, c) = z^2 + c$;

 инициализация итерируемой функции

3: для $c_i \in D_2, i = 1, \dots, n$

 разбиваем область D_2 на сетку из n точек c_i

4: $z = 0$;

5: повторять I раз

6: $z = f_{c_i}(z)$;

 итерируем точку $z = 0$

7: если $|z| > R$, то выход из цикла;

 точка c_i нам не подходит

8: если $|z| \leq R$, то $path = path \cup \{c_i\}$;

 $z = 0$ осталась в D_2 после I итераций

9: Отобразить массив точек $path$.

Задание

6.1.1. Зафиксировать параметр $c \in \mathbb{C}$ и построить заполненное множество Жюлиа $J(f_c)$.

6.1.2. На что влияют параметры n, I ?

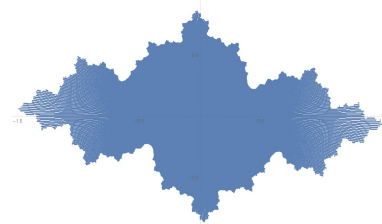
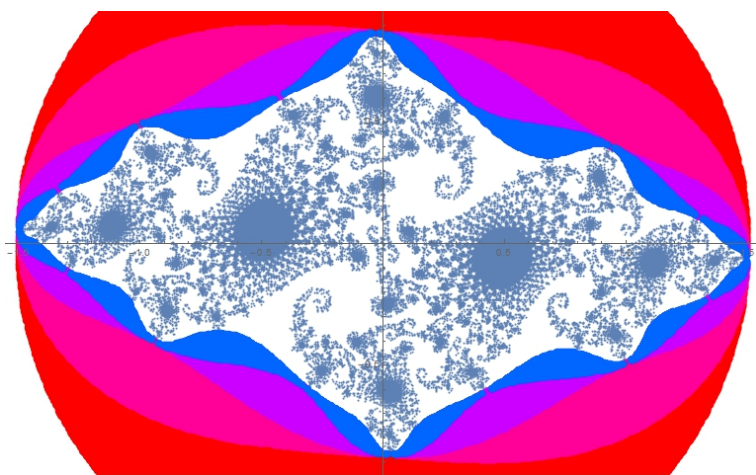
6.1.3. Использовать свойство симметричности $J(f_c)$ для сокращения вычислений.

6.1.4. Изменить программу так, чтобы вычислялся и рисовался только некоторый фрагмент множества («увеличение»).

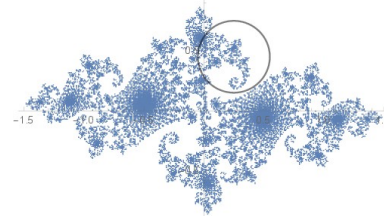
6.1.5. Изменить программу так, чтобы цвет точки вне $J(f_c)$ зависел от ее «скорости убегания», за меру которой можно взять количество итераций, необходимых для выхода из круга D_R .

6.1.6. Построить множество Жюлиа $J(f_c)$ методом обратных итераций или его рандомизированным вариантом.

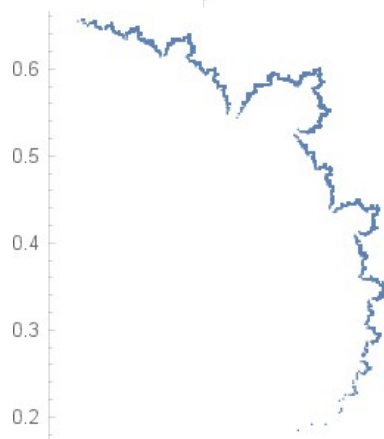
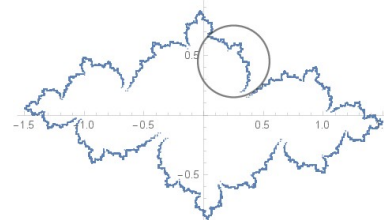
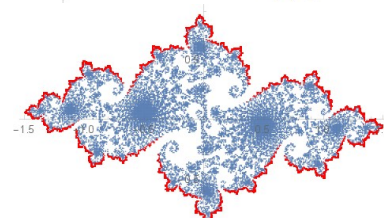
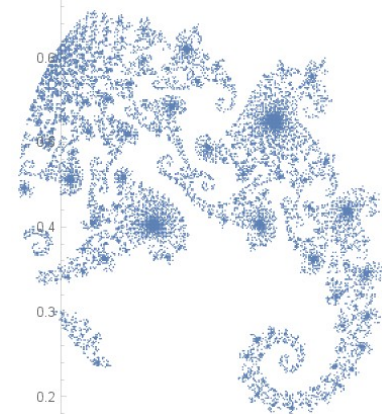
6.1.7. Изменить программу так, чтобы вычислялся и рисовался только некоторый фрагмент границы («увеличение»).



$I = 10$



$I = 400$



6.1.8. Построить множество Мандельброта.

6.1.9. Для сокращения вычислений использовать свойство симметричности, а также тот факт, что бóльшая по площади часть множества заключена в кардиоиду

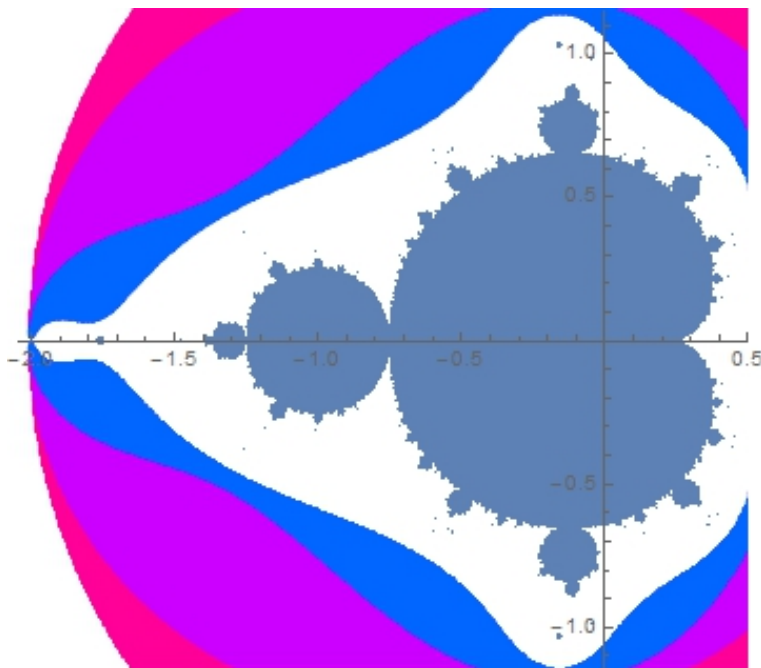
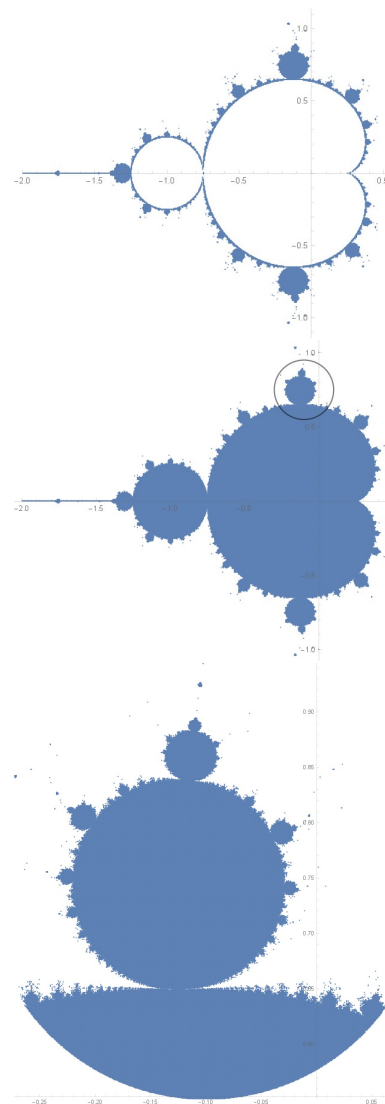
$$((x - 1/4)^2 + y^2)((x - 1/4)^2 + y^2 + x - 1/4) < y^2/4,$$

а вторая по площади часть — окружность

$$(x + 1)^2 + y^2 < 1/16.$$

6.1.10. Изменить программу так, чтобы вычислялся и рисовался только некоторый фрагмент границы («увеличение»).

6.1.11. Изменить программу так, чтобы цвет точки вне \mathcal{M} зависел от ее «скорости убегания», за меру которой можно взять количество итераций, необходимых для выхода из круга D_2 .



6.2 Итерации на плоскости. Игра в хаос

Рассмотрим $2 \leq N \in \mathbb{N}$ *сжимающих* отображений на плоскости $f_i: \mathbb{C} \rightarrow \mathbb{C}$ (или $f_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$), $i = 1, \dots, N$. Достаточные условия сжатия:

- Если $f: \mathbb{C} \rightarrow \mathbb{C}$, $f = f(z)$, $(f'_z = 0)$, $\exists K: |f'(z)| \leq K < 1 \forall z \in \mathbb{C}$.
- Если $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $f(x) = Ax + b$ — аффинное отображение, и собственные значения матрицы A $|\lambda_i| < 1$.

Тогда случайная орбита произвольной точки $z \in \mathbb{C}$ притягивается к некоторому множеству A («аттрактору») и хорошо его приближает. При этом $A = \bigcup_{i=1}^N f_i(A)$, то есть это множество состоит из деформированных копий самого себя (самоподобие).

Отображения $\{f_i\}_{i=1}^N$ называют *системой итерированных функций*, а A — ее *аттрактором*. Для построения аттракторов, часто являющихся *фракталами*, используют *детерминированный* и *рандомизированный* алгоритмы. Последний — наиболее эффективен и иногда называется *игрой в хаос*.

АЛГОРИТМ 6.2.1: ИГРА В ХАОС.

Вход: f_1, \dots, f_m ,

`points`,

`skip`.

 *итерируемые функции*

 *количество точек*

 *количество пропущенных точек*

Выход: Изображение аттрактора.

1: `path` = \emptyset ;

 *инициализация пустого массива*

2: $x_0 = \vec{0}$;

 *начальная точка итераций*

3: **повторять** `skip` **раз**

4: $k = \text{случайное число от } 1 \text{ до } m$;

5: $x_0 = f_k(x_0)$;

6: **повторять** `points` **раз**

7: $k = \text{случайное число от } 1 \text{ до } m$;

8: $x_0 = f_k(x_0)$;

9: `path` = `path` $\cup \{x_0\}$.

10: Отобразить массив точек `path`.

Задание

6.2.1. Методом игры в хаос построить аттракторы следующих систем итерированных функций в \mathbb{C} :

$$\bullet \text{ a) } \begin{cases} f_1(z) = qz, \\ f_2(z) = qz + 1, \end{cases} \quad z \in \mathbb{C}.$$

Рассмотрите случаи $q \in \mathbb{C}$, $|q| < 1$. Например:

i $q_1 = 0.56 + 0.34i$,

ii $q_2 = 0.6 + 0.34i$,

iii $q_3 = 0.5 + 0.5i$,

iv $q_4 = \alpha = -0.419643 - 0.606291i$.

$$\bullet \text{ b) } \begin{cases} f_1(z) = 0.5(1 + i)z, \\ f_2(z) = 1 - 0.5(1 - i)z, \end{cases} \quad z \in \mathbb{C}.$$

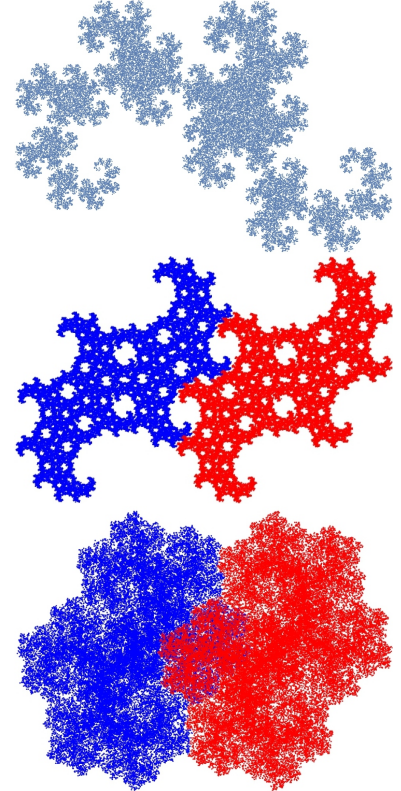
$$\bullet \text{ c) } \begin{cases} f_1(z) = (1/2 + i\sqrt{3}/6)(1 - z), \\ f_2(z) = (-1/2 + i\sqrt{3}/6)z + 1, \end{cases} \quad z \in \mathbb{C}.$$

$$\bullet \text{ d) } \begin{cases} f_1(z) = 0.5z, \\ f_2(z) = 0.5z + 0.5, \\ f_3(z) = 0.5z + (0.25 + 0.5\sqrt{3}i), \end{cases} \quad z \in \mathbb{C}.$$

$$\bullet \text{ e) } \begin{cases} f_1(z) = \alpha z, \\ f_2(z) = \alpha^2 z + 1, \\ f_3(z) = \alpha^3 z + (\alpha^2 + 1), \end{cases} \quad z \in \mathbb{C},$$

$\alpha = -0.419643 - 0.606291i$ — один из комплексных корней характеристического уравнения чисел трибоначчи $z^3 - z^2 - z - 1 = 0$.

6.2.2. Как можно истолковать геометрически умножение переменной z на комплексное число и сложение с комплексным числом?



6.2.3. Методом игры в хаос построить аттракторы следующих систем итерированных функций в \mathbb{R}^2 :

$$f_i \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = A_i \begin{bmatrix} x \\ y \end{bmatrix} + b_i, \quad \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2.$$

• а) $i = 1, 2$:

$$A_1 = \begin{bmatrix} 0.4 & -0.3733 \\ 0.06 & 0.6 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 0.3533 \\ 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -0.8 & -0.1867 \\ 0.1371 & 0.8 \end{bmatrix}.$$

$$b_2 = \begin{bmatrix} 1.1 \\ 0.1 \end{bmatrix}.$$

• б) $i = 1, 2, 3$:

$$A_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & -0.5 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix},$$

$$b_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

$$b_3 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}.$$

• в) $i = 1, 2, 3, 4, 5$:

$$A_1 = \begin{bmatrix} 0.1950 & -0.4880 \\ 0.3440 & 0.4430 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 0.4431 \\ 0.2452 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0.4620 & 0.4140 \\ -0.2520 & 0.3610 \end{bmatrix},$$

$$b_2 = \begin{bmatrix} 0.2511 \\ 0.5692 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} -0.0580 & -0.0700 \\ 0.4530 & -0.1110 \end{bmatrix},$$

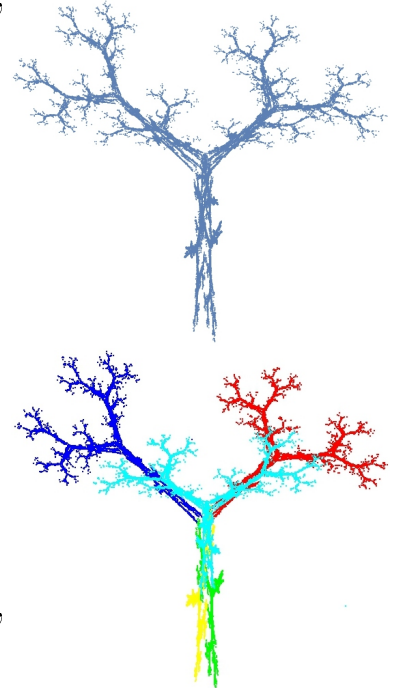
$$b_3 = \begin{bmatrix} 0.5976 \\ 0.0969 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} -0.0350 & 0.0700 \\ -0.4690 & 0.0220 \end{bmatrix},$$

$$b_4 = \begin{bmatrix} 0.4884 \\ 0.5069 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} -0.6370 & 0.0000 \\ 0.0000 & 0.5010 \end{bmatrix}.$$

$$b_5 = \begin{bmatrix} 0.8562 \\ 0.2513 \end{bmatrix}.$$



6.2.4. Изменить алгоритм программы так, чтобы каждая из деформированных копий аттрактора была окрашена в свой цвет. Для этого добавим к каждой точке в получающейся последовательности атрибут, — ее цвет, — в соответствии с номером того отображения i , каким она была получена.

Хранить цвет у каждой точки неэффективно, вместо этого для записи орбиты будем сортировать точки по \mathbf{N} массивам, которые при выводе на экран окрасим в свои цвета. Если точка \mathbf{z} получилась под действием отображения с номером i , добавим ее в массив с номером i .

6.3 Итерации на плоскости. Хаотические аттракторы

Простые отображения могут вести себя неожиданно сложным образом.

Задание

Постройте итерации (орбиту) $z_{k+1} = f(z_k)$ следующих отображений. Попробуйте поменять коэффициенты отображения, а также начальную точку. Соединив последовательно точки орбиты, убедитесь, что она заполняет аттрактор хаотически.

6.3.1. Аттрактор отображения Тинкербеллы¹

$$f(\{x, y\}) = \{x^2 - y^2 + ax + by, 2xy + cx + dy\},$$

$$\{a, b, c, d\} = \{0.9, -0.6013, 2, 0.5\}, \quad z_0 = \{-0.72, -0.64\}.$$

6.3.2. Аттрактор без названия

$$f(\{x, y\}) = \{0.6x(1 + 2x) + 0.8y(x - 1) - y^2 - 0.9,$$

$$0.1x(1 - 6x + 4y) + 0.1y(1 + 9y) - 0.4\}, \quad z_0 = \{-0.5, 0\}.$$

6.3.3. Аттрактор отображения Эно

$$f(\{x, y\}) = \{1 + 1.4x^2 + y, 0.3x\}, \quad z_0 = \{0, 0\}.$$

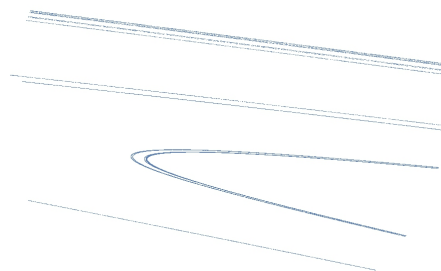
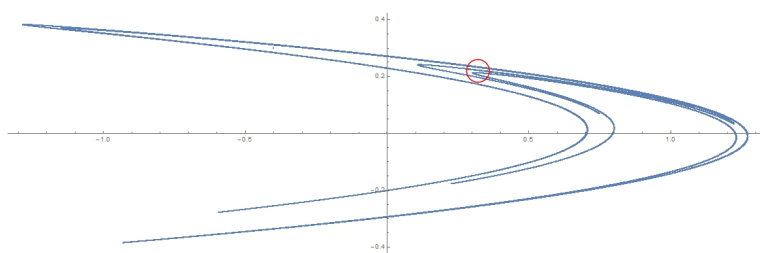
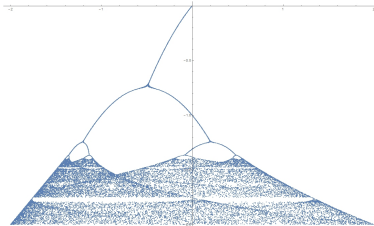
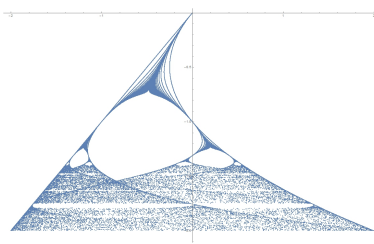
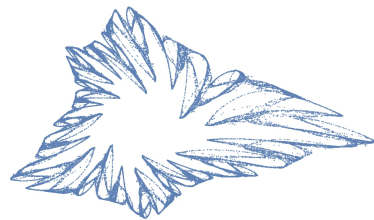
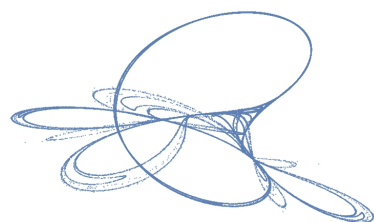
Сделайте также увеличенный фрагмент.

6.3.4. Бифуркационная диаграмма

Постройте итерации (орбиты) $z_{k+1} = f(z_k)$, $k = 1, \dots, 500$, квадратичного отображения

$$f(x) = x^2 + c, \quad z_0 = 0,$$

при различных значениях $c \in [-2, 0]$ на одном рисунке. Попробуйте отображать каждую орбиту, начиная с 50-го члена (что изменится при этом?).



¹Имя диснеевской феи из Питера Пена. Почему такое название аттрактора?

Основы пакета МАТНЕМАТИСА

для построения изображений

7.1 Графика

Графические примитивы

Простейший инструмент рисования — *графические примитивы*: точки, отрезки прямых, ломаные, многоугольники, стрелки, эллипсы (эллипсоиды)... и даже текст.

Чтобы «примитив» стал графическим объектом, его нужно поместить внутрь команды `Graphics` / `Graphics3D`. Кроме того, не забывайте использовать справку, в которой есть дополнительные опции к каждой команде.

Наиболее полезные графические примитивы:

- `Line[{{x1, y2}, {x2, y2}}]` — отрезок между точками (x_1, y_1) и (x_2, y_2) ;
- `Line[{{x1, y2}, ..., {xn, yn}}]` — ломаная, соединяющая точки $(x_1, y_1), \dots, (x_n, y_n)$;
- `Point[{x, y}]` — точка (x, y) ;
- `Sphere[{x, y}, r]` — сфера с центром (x, y) радиуса r (3D);
- `Arrow[{{x1, y2}, {x2, y2}}]` — вектор (стрелка) с началом в (x_1, y_1) и концом в (x_2, y_2) ;
- `Arrow[Tube[{{x1, y1, z1}, {x2, y2, z2}}]]` — объемный вектор с началом в (x_1, y_1, z_1) и концом в (x_2, y_2, z_2) (3D);
- `Circle[{x, y}, r]` — окружность с центром (x, y) и радиусом r (есть возможность нарисовать эллипс);
- `Disk[{x, y}, r]` — круг с центром (x, y) и радиусом r .

Цвет, толщина и стиль линий в графических примитивах задаются свойствами перед именем примитива:

- `Graphics[Red, Thickness[0.1], Opacity[0.5], Dashed, Line[{{0, 0}, {1, 1}}]]` — полупрозрачный отрезок пунктиром красного цвета толщины 0.1.

- `Graphics[{PointSize[0.1], Point[{0,0}]}]` — точка размера 0.1.
- Текст также используется как графический примитив (для надписей):
 - `Text["Text", {x,y}]` — текст `Text` на позиции (x, y) ;
 - `Rotate[Text["Text", {x,y}], k Degree]` — текст `Text` на позиции (x, y) под углом k градусов;
 - `Text[Style["Text", Large], {x,y}]` — текст `Text` на позиции (x, y) большим размером шрифта.
- `Graphics[Text[Style["Text", Blue, Italic, 24], {3,5}]]` — текст `Text` на позиции $(3, 5)$ голубым цветом наклонным шрифтом размера 24.

Построение графиков

Пакет MATHEMATICA позволяет эффективно строить графики функций и поверхностей (а также кривые и поверхности).

- `Plot[f(x), {x, a, b}]` — график функции (кривая) $y = f(x)$ на интервале $x \in [a, b]$;
- `Plot3D[f(x, y), {x, a, b}, {y, c, d}]` — график функции (поверхность) $y = f(x, y)$ в прямоугольнике $(x, y) \in [a, b] \times [c, d]$;
- `ParametricPlot[{x(t), y(t)}, {t, a, b}]` — график кривой, заданной параметрически $r(t) = \{x(t), y(t)\}$ на интервале $t \in [a, b]$;
- `ParametricPlot3D[{x(t), y(t), z(t)}, {t, a, b}]` — график кривой, заданной параметрически $r(t) = \{x(t), y(t), z(t)\}$ на интервале $t \in [a, b]$;
- `ParametricPlot3D[{x(u, v), y(u, v), z(u, v)}, {u, a, b}, {v, c, d}]` — график поверхности, заданной параметрически $r(u, v) = \{x(u, v), y(u, v), z(u, v)\}$ в прямоугольнике $(u, v) \in [a, b] \times [c, d]$;
- `ParametricPlot[ρ(φ), {t, a, b}]` — график кривой, заданной в полярной системе координат $\rho = \rho(\phi)$ на интервале $\phi \in [a, b]$;
- `ContourPlot[F(x, y)==0, {x, a, b}, {y, c, d}]` — график кривой, заданной неявно $F(x, y) = 0$ в прямоугольнике $(x, y) \in [a, b] \times [c, d]$;
- `ContourPlot3D[F(x, y, z)==0, {x, a, b}, {y, c, d}, {z, e, f}]` — график поверхности, заданной неявно $F(x, y, z) = 0$ в параллелепипеде $(x, y, z) \in [a, b] \times [c, d] \times [e, f]$.

Наиболее полезные графические опции:

- `Axes->False` — отключение изображения осей;
- `Boxed->False` — отключение изображения рамки;
- `PlotStyle->Red` — задание цвета графика;
- `AspectRatio->Automatic` — правильное соотношение сторон изображения;
- `PlotRangePadding->a` — отбивка изображаемой области графика на `a` единиц;
- `PlotRangeClipping->False` — отмена обрезки графика по его границам;
- `ContourStyle->Red` — изменение опций для `ContourPlot`;
- `Mesh->None` — отключение изображения координатных линий;
- `Ticks->None` — отключение изображения засечек на осях;
- `Opacity->k` — установка прозрачности графика, $k \in [0, 1]$;
- `RegionFunction` — функция, которая задает какая область (регион) из области определения графика отображается — полезна, когда нужно отобразить лишь кусок поверхности, см. примеры в справке.

Совмещение изображений

Часто возникает необходимость изобразить одновременно несколько графических объектов на одном рисунке. Команда `Show` выводит все графические объекты на одном рисунке, эффективно совмещая их и используя опции самого первого объекта.

`Show[g1, ..., gn]` — графические объекты `g1, ..., gn` на одном изображении.

Однако получившийся результат располагается в области самого первого графического объекта и при несовпадении областей определения и значений часть объектов может быть не видна. Решить эту проблему можно двумя путями: использовать опцию `PlotRangePadding` у тех объектов, которые не поместились на рисунке, или используя опцию `PlotRange->All` самой команды `Show`.

```
g1=Graphics[Circle[{1, 1}, 1]];
g2=Plot[Sin[x], {x, 0, Pi}];
Show[g2, g1, PlotRange->All] — попробуйте убрать опцию PlotRange!
```

Экспорт и импорт графических объектов

Пакет Mathematica позволяет экспортировать и импортировать для дальнейшей обработки различные типы объектов, в том числе и графические.

Импорт и экспорт (сохранение) графического объекта задаются командами `Import` и `Export`.

Также можно сохранить изображение, выбрав контекстное меню рисунка, и в нем — команду **Сохранить графику...**

Примеры:

- `plt=Plot[Sin[x], {x, 0, 2 Pi}]` — запись графического объекта в переменную `plt`.

- `Export["sinx.jpg", plt]` — сохранение в директорию по умолчанию.

Чтобы изменить директорию по умолчанию, на папку, в которой сохранен редактируемый ноутбук, воспользуйтесь командами

`SetDirectory[NotebookDirectory[]]` или `SetDirectory["путь к папке"]`.

- `Export["D:\\sinx.jpg", plt]` — сохранение в директорию `D:\\`

- `Export["sinx.jpg", plt, ImageSize->500]` — сохранение изображения с горизонтальным размером в 500 пикселей.

- `Export["sinx.jpg", plt, ImageResolution->200]` — сохранение изображения с плотностью пикселей 200.

Экспортировать в графику можно и другие объекты, например, формулы:

- `Export["sinx.jpg", Integrate[1/(x^4-1), x]]` — сохранение результата интегрирования в виде изображения.

- `Export["sinx.jpg", HoldForm[Integrate[1/(x^4-1), x]]]` — сохранение интеграла без вычисления в виде изображения.

Графические объекты и методы их представления и записи в файл делятся на два вида: *растровые* и *векторные* изображения.

- Растровое изображение: в памяти хранится цвет каждого пиксела в решетке фиксированного разрешения.
- Векторное изображение: такие объекты как прямые линии, векторы, гладкие кривые — хранятся в аналитическом виде, что позволяет существенно сократить размер файла и повысить точность воспроизведения (нет эффекта пикселизации).

Если изображение состоит из векторных объектов, его лучше сохранять в векторном формате (**pdf**, **eps**, **ps**...), а если оно состоит из большого множества отдельных точек, которые сложно описать аналитическими формулами, то следует сохранять в растровом формате (**jpg**, **bmp**, **png**, **gif**...). Заметим, что для публикации графики в статье журнала, в книге или при высокоточной печати следует хранить графику в достаточно большом разрешении или в векторном формате.

7.2 Функции, модули, процедуры

Для задания функции используется отложенное присвоение `:=`, а не `=`, а также не забывайте символ `_` у аргумента!

Примеры:

- `f[x_] := Sin[x]` — заданий функции.

- `f[x_] := (a = x^2; b = a^2; Print[a]; Sin[b])` — задание процедуры (несколько действий подряд).

Для избежания ошибок рекомендуется использовать модули: в них переменные локализуются.

- `f[x_] := Module[{a = 1}, Sin[a x]]` — здесь **a** — локальная переменная со значением по умолчанию 1.

- `f[xx_] := Module[{x}, D[Sin[x], x] /. x -> xx]` — правильное задание производной функции (см. подстановки), здесь **x** — локальная переменная без значения по умолчанию.

7.3 Подстановки

Полноценное использование пакета Mathematica подразумевает применение *подстановок*. Подстановка задается правилом **s=a->b**" замены каждого элемента **a** в некотором выражении на элемент **b**. Применение подстановки осуществляется командой `/.`

`c/.s` — применить подстановку **s** в выражении **c**.

Подстановка может состоять из нескольких правил:

s={a₁->b₁, ..., a_n->b_n}.

Элемент **a** может быть сложным выражением, например, функцией.

Также см. далее решение уравнений командами **Solve** и **DSolve** и модули.

Примеры:

- $x /. x \rightarrow 1$ — возвращает значение 1, не изменяя значение переменной x .
- $(x^2 + x + y) /. x \rightarrow 1$ — возвращает значение $2 + y$, не изменяя значение переменной x .
- $(x^2 + x + y) /. \{x \rightarrow 1, y \rightarrow 2\}$ — возвращает значение 4, не изменяя значение переменных x и y .
- $(y[x] + \text{Cos}[x]) /. y[x] \rightarrow \text{Sin}[x]$ — возвращает $\text{Sin}[x] + \text{Cos}[x]$.

7.4 Решение уравнений

При решении уравнений используется логическое равенство $==$, а не $=$!

- $\text{Solve}[F == 0]$ — решение уравнения $F(x) = 0$;
- $\text{Solve}[F == 0, x]$ — решение уравнения $F(x, y) = 0$ относительно x ;
- $\text{Solve}[F == 0, x, \text{Reals}]$ — вещественные корни уравнения $F(x) = 0$;
- $\text{Solve}[\{F == 0, G == 0\}]$ — решение системы уравнений $F(x) = 0, G(x) = 0$.

Не всегда решение может задаваться аналитической формулой. В таких случаях (или когда аналитическая формула слишком громоздка) проще воспользоваться численным решением:

- $\text{NSolve}[F == 0]$ — приближенное решение уравнения $F(x) = 0$;
- $\text{FindRoot}[F == 0, \{x, x_0\}]$ — приближенное решение уравнения $F(x) = 0$ в окрестности точки x_0 .

Решение всегда представляется массивом подстановок!

Примеры:

- $s = \text{Solve}[x^2 - 5x + 6 == 0]$ — запись решения (подстановки) в переменную s :
 $\text{Out}[1] = \{\{x \rightarrow 2\}, \{x \rightarrow 3\}\}$
- $x /. s$ — непосредственно получение массива решений:
 $\text{Out}[2] = \{2, 3\}$

При решении дифференциальных уравнений используются команды DSolve и NDSolve . Ответ представляется массивом подстановок на аналитические выражения или на интерполяционные функции.

Примеры:

Численное решение на интервале $x \in [0, 5]$

```
s=NDSolve[{y'[x]+y[x]==2Sin[x], y[0]==0}, y[x], {x,0,5}]
```

— представляется интерполяционной функцией:

```
Out[3]= {{y[x] -> InterpolatingFunction[{{0., 5.}}, <>][x]}}
```

Аналитическое решение

```
s=DSolve[y'[x]+y[x]==2Sin[x], y[x], x]
```

— дает ответ с константой:

```
Out[4]= {{y[x] -> e^{-x} C[1] - Cos[x] + Sin[x]}}
```

Как видно из формы ответа, **s** — это множество состоящее из массива подстановок. Чтобы вытащить этот массив подстановок, надо взять первый элемент **s[[1]]**.

```
Plot[y[x]/.s[[1]], {x, -0.01, 0.01}]
```

— график решения.

```
y[x]/.s[[1]]/.x->0.001
```

— вычисление значения в точке.

```
Y[xx_,c_]:= (y[x]/.s[[1]])/.{x->xx, C[1]->c}
```

— запись решения
в новую функцию.

7.5 Дифференцирование вектор-функций

Примеры:

Для функции одной переменной

```
f[t_]:= {t, t^2, t^3}
```

— достаточно символа '

```
f'[t]; f''[t]
```

— $\frac{df}{dt}, \frac{d^2f}{dt^2}$

```
F[t_]:=f'[t]
```

— производная как новая функция.

Для функции двух переменных нужно использовать оператор дифференцирования и модуль

```
D[f[u,v], u]
```

— $\frac{\partial f}{\partial u}$

```
F[uu_,v_]:=Module[{u,v}, D[f[u,v], u]/.u->uu]
```

— частная
производная как новая функция.

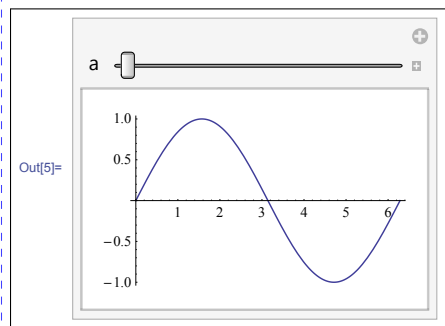
7.6 Манипулирование, динамическое представление

Введите параметр в вашу графику и динамически изменяйте его при помощи Manipulate.

Пример:

```
Manipulate[Plot[Sin[x+a],{x,0,2Pi}],{a,0,Pi}]
```

— динамическая презентация с параметром $a \in [0, \pi]$.



7.7 Векторная алгебра

Векторы и матрицы записываем массивами:

$v=\{x,y\}$, $w=\{x,y,z\}$, $A=\{\{x_{11},x_{12}\},\{x_{21},x_{22}\}\}$.

- Norm[v] — длина вектора v;
- v1.v2 (A1.A2) — скалярное произведение векторов (матриц);
- Cross[v] — повернутый на $\pi/2$ вектор v;
- Cross[w1.w2] — векторное произведение векторов $w1 \times w2$;
- Det[A] — определитель матрицы A;
- Inverse[A] — обратная матрица к A;
- Transpose[A] — транспонированная матрица A;
- Eigenvalues[A] — массив собственных значений матрицы A;
- Eigenvectors[A] — массив собственных векторов матрицы A;
- Eigensystem[A] — массив собственных значений и собственных векторов матрицы A.

7.8 Операторы программирования

Для программирования вам понадобятся операторы `Do`, `While`, `Break`, `For`, `If`. Их синтаксис см. в справке.

7.9 Хитрости

- В случаях, когда вектор не задан конкретными числами, лучше вместо `Norm[v]` использовать `Sqrt[v.v]`.
- После преобразования выражения (например, дифференцирование) упростите его командой `Simplify`.
- Делайте ваши рисунки максимально информативными, а текст программы комментируйте в символах (* комментарий *).

Литература

- [1] *Мищенко, А. С.* Курс дифференциальной геометрии и топологии / А. С. Мищенко, А. Т. Фоменко. — МГУ, 1980. — 439 с.
- [2] *Рашевский, П. К.* Курс дифференциальной геометрии / П. К. Рашевский. — 3 изд. — М. ; Л. : ГИТТЛ, 1950. — 428 с.
- [3] *Шануков, Б. Н.* Дифференциальная геометрия и основы тензорного анализа. курс лекций. — [Электронный ресурс]. <http://old.kpfu.ru/f5/index.php?id=7&num=2>.
- [4] *Шурыгин, В. В.* Аналитическая геометрия I. Учебное пособие к курсу / В. В. Шурыгин. — КГУ, 2006. — 98 с. <http://old.kpfu.ru/f5/shurygin/geometry1.pdf>.
- [5] *Шурыгин, В. В.* Аналитическая геометрия II. Учебное пособие к курсу / В. В. Шурыгин. — КГУ, 2008. — 70 с. <http://old.kpfu.ru/f5/shurygin/geometry2.pdf>.
- [6] *Шурыгин, В. В.* Аналитическая геометрия III. Учебное пособие к курсу / В. В. Шурыгин. — КГУ, 2007. — 101 с. <http://old.kpfu.ru/f5/shurygin/geom3.pdf>.
- [7] *Кроновер, Р. М.* Фракталы и хаос в динамических системах. Основы теории / Р. М. Кроновер. — М. : Постмаркет, 2000. — 352 с.
- [8] *Трошин, П. И.* Моделирование фракталов в среде Mathematica. Часть I. Фракталы на плоскости и в пространстве. Учебно-методическое пособие / П. И. Трошин. — Казань : КФУ, 2014. — 92 с. http://kpfu.ru/docs/F1416066913/main._1_.pdf.

Об авторе

ТРОШИН Павел Игоревич. Кандидат физико-математических наук, старший преподаватель кафедры геометрии Казанского (Приволжского) федерального университета. Основное направление научных исследований — фрактальная геометрия и динамические системы.

Замечания и пожелания можно отправлять автору по адресу:

Paul.Troshin@gmail.com

