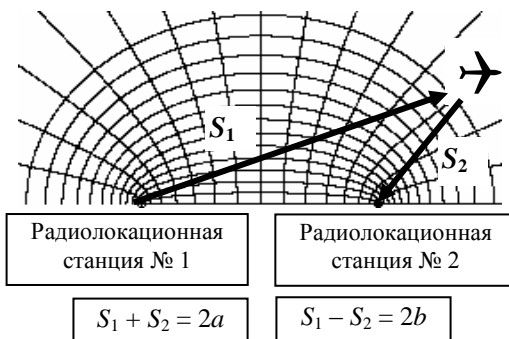


П.И. Совертков



ИССЛЕДОВАТЕЛЬСКИЕ ПРОЕКТЫ ПО МАТЕМАТИКЕ И ИНФОРМАТИКЕ

Практикум



**Издательство
Нижевартовского
государственного
университета
2013**

Печатается по постановлению Редакционно-издательского совета
Нижевартовского государственного университета

Рецензенты:

доктор физико-математических наук,
профессор Российского государственного
педагогического университета им. А.И.Герцена *А.Л.Вернер*;

доктор педагогических наук,
профессор Омского государственного
педагогического университета *В.А.Далингер*

Совертков П.И.

С 56 Исследовательские проекты по математике и информатике:
Практикум. — Нижневартовск: Изд-во Нижневарт. гос. ун-та,
2013. — 331 с.

ISBN 978–5–00047–042–8

Практикум предназначен для формирования практических навыков по математическому моделированию и разработке программ построения объектов компьютерной графики.

Тематика трех глав посвящена элективным курсам по математике и информатике: методам построения изображений линий и поверхностей в пространстве, поиску различных видов функциональной зависимости.

Представленные материалы могут служить основой для элективных курсов по информатике и математике в школах, гимназиях и лицеях; для дисциплин «Элементарная математика», «Компьютерное моделирование» и курсов по выбору в педагогических вузах по специальностям «Математика», «Информатика».

Для старшеклассников, готовящихся к участию в разработке исследовательских проектов по математике и информатике, для учителей математики и информатики, преподавателей, студентов и аспирантов педагогических университетов.

ISBN 978–5–00047–042–8

© Совертков П.И., 2013
© Издательство НВГУ, 2013

Оглавление

Введение	5
Глава 1. Приемы моделирования простейших объектов	9
§ 1. Геометрия многоугольных чисел и основы компьютерной графики	11
§ 2. Проект «Моделирование систем координат»	34
§ 3. Параметризация линий	52
§ 4. Эквидистантные линии и огибающая линия	61
§ 5. Построение аналогичных фигур	74
§ 6. Приемы изображения движущегося объекта	77
§ 7. Геометрия шарнирных механизмов	85
§ 8. Изображение пространственной фигуры на плоскости	105
§ 9. Закрашивание областей в разные цвета	114
§ 10. Пунктирная линия и радианная мера дуги	127
§ 11. Седловые поверхности в строительстве	132
§ 12. Построение диаграммы Вороного	150
Глава 2. Поиск функциональной зависимости	165
§ 13. Линии уровня функции	167
§ 14. Производящая функция	177
§ 15. Интерполяционные многочлены	186
§ 16. Булевы функции от трех переменных, удовлетворяющие условию коммутативности теста	197
§ 17. Компьютерное тестирование с помощью булевых функций от трех переменных	207
Глава 3. Математическая и компьютерная смесь	221
§ 18. Моделирование семейства педальных треугольников	223
§ 19. Последовательности квадратов и спирали	229
§ 20. Окружности Монжа и цветовая композиция от трех источников света	241
§ 21. Многозвенная периодическая траектория математического бильярда в треугольнике	251
§ 22. Моделирование замечательных точек T_i треугольника	261

§ 23. Метрика на поверхности и кратчайшие линии на кубе.....	272
§ 24. Опорная функция линии и радиус кривизны.....	297
§ 25. Матрица с доминирующей диагональю	311
Справочный материал	321
Ответы	326
Литература	331

Введение

Одной из важнейших форм подготовки исследователя является проектная деятельность, которая предполагает дополнительное предметное образование, формирование навыков исследовательской деятельности и разработку проекта. Сформулированные задачи должны осуществляться совместно или быть реализованы по направлениям, но подчинены единой цели.

В данном пособии предлагается несколько элективных курсов по математике, информатике и частично по математической картографии, в которых представлен дополнительный материал на уровне, доступном для учащихся старшего звена средней школы, сформулированы многочисленные задачи для самостоятельного исследования и предложены методы формирования навыков исследовательской работы. Подробно обсуждаются этапы разработки проекта.

После изучения первых двух параграфов первой главы практикума можно переходить к изучению любой главы.

Настоящее пособие имеет много точек пересечения с предметным материалом, представленным в других пособиях автора [19—21], но является полностью независимым от них.

Поиск функциональной зависимости представлен на различных задачах математики и информатики. Изучается переход от одного способа задания функции к другим способам, рассматривается переход от рекуррентного соотношения к явному заданию функции с помощью производящей функции.

Содержание каждой из глав 1, 2 можно выбрать для отдельного элективного интегративного курса по математике и информатике. В главе 3 представлены различные темы для исследования.

Методические рекомендации по реализации элективных курсов представлены в методическом пособии.

Предлагаемые элективные курсы построены по трехуровневой системе обогащения.

Первый уровень предусматривает знакомство учащихся с различными разделами математики и информатики, которые могут их заинтересовать.

Второй уровень направлен на специальное развитие мышления учащегося. Предлагаемые задания ориентированы на тренировку

наблюдательности и анализ закономерности в предлагаемой геометрической фигуре, способности выдвигать гипотезу, моделировать и экспериментировать с построенной моделью.

Третий уровень — самостоятельные исследования при решении творческих задач в новом проекте.

Пособие ориентирует на формирование пространственного восприятия и пространственной интуиции, что составляет важный компонент понимания и осмысления мира. Многие модели красиво и изящно изображаются на компьютере. Они способствуют развитию эстетического вкуса и пониманию красоты окружающего мира. Тем более, когда такие произведения представлены в своей голове и доведены до реализации на компьютере своими руками.

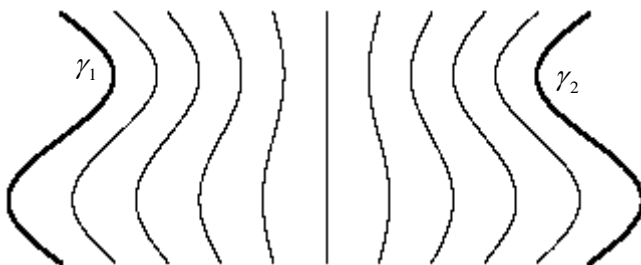
Для всех разработанных моделей предложены программы на языке Visual Basic 6. Читатель, не знакомый с этим языком, может познакомиться с ним, изучив несколько параграфов первой главы пособия. Если в тексте пособия или в ответе задачи название программы указано в кавычках, например, «**Многоугольные числа**», то эта программа представлена на компакт-диске, прилагаемом к учебному пособию. Операторы различных языков программирования для компьютерной графики незначительно отличаются, поэтому на компакт-диске приведены три программы для картографических проекций на языке Паскаль, программы закрашки области (§ 9, 10) на исполнителе Чертежник, изучаемом в курсе информатики, а также программы на языке Pascal ABC. Сопоставив программы на разных языках, можно легко «перевести» напечатанные в пособии программы на другой язык программирования. Если читатель не имеет программного продукта Visual Basic 6, то он может воспользоваться скомпилированными программами с расширением **.exe** для просмотра действия программы.

В конце данного пособия приведены справочные материалы по компьютерной графике. Справочные материалы по математическим основам компьютерной графики представлены в учебном пособии. Нумерация задач — двойная. Первое число указывает номер главы, а второе — номер задачи в главе. Это пособие содержит около 400 рисунков, большинство из которых напечатаны с помощью компьютерных программ.

Автор выражает благодарность ректору Нижневартковского государственного гуманитарного университета С.И.Горлову за поддержку инициативы подготовки пособий, направленных на развитие одаренных детей.

Глава 1

ПРИЕМЫ МОДЕЛИРОВАНИЯ ПРОСТЕЙШИХ ОБЪЕКТОВ



Что нас ожидает в этой главе?

1. Составление параметрических уравнений линии и поверхности.
2. Приемы изображения пунктирной линии.
3. Приемы изображения движущегося объекта.
4. Изображение пространственной фигуры на плоскости.
5. Построение линии, отстоящей на постоянном расстоянии от данной линии.
6. Прием «пропорционального смешивания» двух линий.
7. Приемы раскраски областей.

§ 1. Геометрия многоугольных чисел и основы компьютерной графики

Для большинства объектов, изучаемых в данном пособии, разработано математическое и компьютерное моделирование, поэтому познакомимся с простейшими методами моделирования геометрических объектов и подготовкой проекта для компьютерного моделирования.

Все программы в пособии разработаны на языке Visual Basic 6.0, но в этом параграфе несколько программ представлены на двух языках: Visual Basic 6.0 и Pascal. Сравнив эти программы, читатель может в дальнейшем переписать представленные в пособии программы с одного языка на другой.

Рассмотрим этапы математического и компьютерного моделирования задачи при разработке проекта.

1^й этап — Постановка и решение математической задачи

Многоугольные числа определяют число кругов, расположенных на плоскости внутри правильного многоугольника, поэтому их иногда называют плоскими фигурными числами. Аналогом многоугольных чисел являются пространственные фигурные числа. Определение многоугольных (фигурных) чисел сформулировано в различных книгах. В большинстве книг приведено только определение и представлены иллюстрации этих чисел на сторонах семейства многоугольников. Математическая теория многоугольных чисел достаточно полно представлена в элективном курсе [10]. Настоящее исследование посвящено разработке нового направления в теории фигурных чисел — моделированию изображения многоугольных чисел на компьютере. Изобразить многоугольные числа на экране компьютера можно простым повторением точек, изображаемых кругами определенного цвета. Несмотря на то, что в этом случае почти не требуется математического моделирования, этот метод не является рациональным, так как расстояния между кругами визуально трудно определить точно равными между собой. Построить, например, правильный одиннадцатиугольник также трудно с соблюдением метрических размеров. В этом случае лучше выбрать метод математического и компьютерного моделирования. Затратив один раз время

на моделирование, можно затем изображать многоугольные числа с различными параметрами.

Наибольшая трудность решения этой проблемы заключается в математическом моделировании. Такой аспект этой тематики не рассматривался в литературе. Следует заметить, что теория пространственных фигурных чисел в книгах излагается без соответствующих рисунков ввиду их сложности или приводятся схематические рисунки, не отражающие метрические размеры.

Многоугольные числа, связанные с геометрическими образами, возникли в древней Индии и Вавилоне на базе практических задач строительства сложных архитектурных деталей из прямоугольных блоков. Книги по истории математики отмечают, что, вероятно, во время своих путешествий ими заинтересовался Пифагор, который построил первую теорию фигурных (многоугольных) чисел. Многоугольные числа пользовались большой популярностью в конце эпохи Возрождения, после того как греческая математика проникла в Западную Европу. Ими занимались видные ученые — Ферма, Эйлер, Гаусс, Коши.

Приведем примеры простейших многоугольных чисел.

На рис. 1 представлены некоторые треугольные числа. Название связано с тем, что треугольные числа $S_3(n)$ т.е. числа 1, 3, 6, 10, 15, 21, 28, ... можно рассматривать как число шаров, уложенных в виде правильного треугольника, в основании которого находится n шаров.

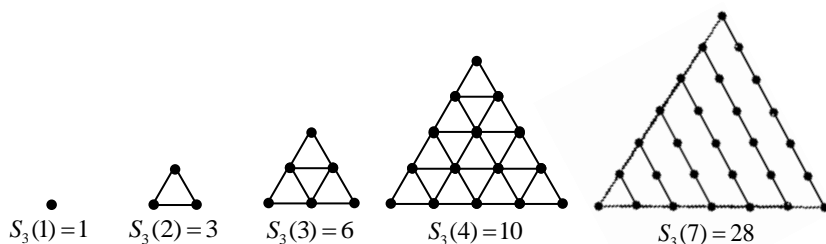


Рис. 1

Треугольные числа можно использовать для подсчета числа труб, уложенных в ярусы (рис. 2).

Для треугольных чисел выполняется рекуррентное равенство $S_3(n+1) = S_3(n) + n + 1$, где $S_3(1) = 1$. Действительно (рис. 3), чтобы получить следующее треугольное число $S_3(n+1)$ из предыдущего треугольного числа $S_3(n)$, нужно продолжить две стороны треугольника, выходящие из одной вершины, и на новой стороне дополнительно построить $(n+1)$ точку.

Треугольные числа вычисляются по формуле:

$$S_3(n) = \frac{n(n+1)}{2}.$$

Для доказательства этой формулы приложим к построенной фигуре равную ей фигуру (рис. 4). Получим n горизонтальных рядов точек, причем в каждом горизонтальном ряду находится $(n+1)$ точка, поэтому

$$2S_3(n) = n(n+1).$$

Замечание. В главе «Поиск функциональной зависимости» будут рассмотрены три способа перехода от рекуррентного соотношения для функции к явному заданию функции.

На рис. 5 представлена последовательность квадратов. Подсчитывая число точек в них, получим квадратные числа 1, 4, 9, 25, 36, ...

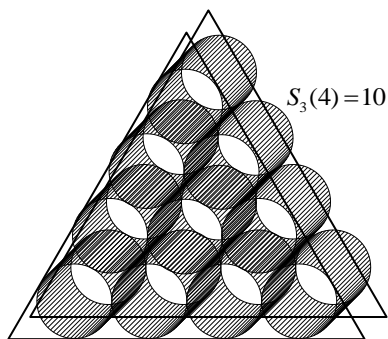


Рис. 2

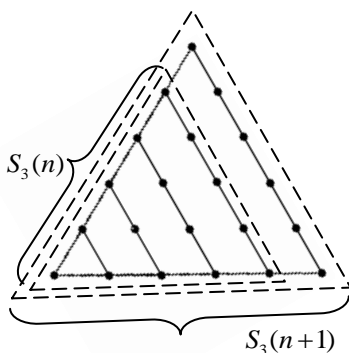


Рис. 3

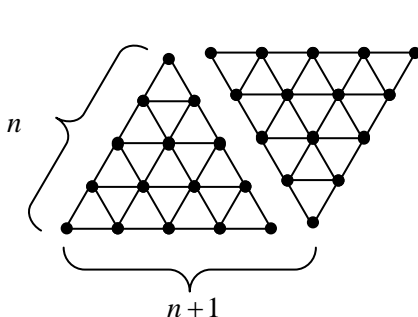


Рис. 4

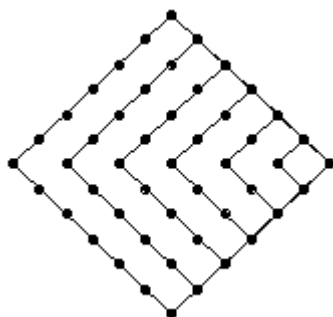


Рис. 5

Для квадратных чисел выполняется рекуррентное равенство:
 $S_4(n+1) = S_4(n) + 2n + 1$, где $S_4(1) = 1$.

Квадратные числа определяются формулой $S_4(n) = n^2$.

На рис. 6 представлена последовательность правильных пятиугольников. Подсчитывая число точек в них, получим пятиугольные числа 1, 5, 12, 22, 35, ...

Для пятиугольных чисел выполняется рекуррентное равенство:
 $S_5(n+1) = S_5(n) + 3n + 1$, где $S_5(1) = 1$.

Найдем формулу для определения пятиугольного числа $S_5(n)$.

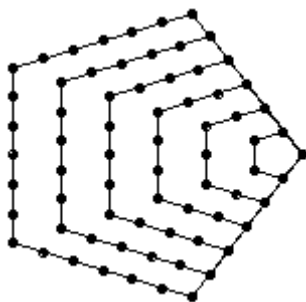


Рис. 6

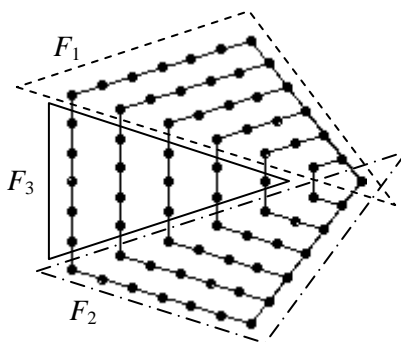


Рис. 7

Первый способ. Выделим на фигуре, состоящей из $S_5(n)$ точек (рис. 7), фигуры, содержащие треугольное число точек, т.е. две равных фигуры F_1 и F_2 , содержащие по $S_3(n)$ точек, и фигуру F_3 , содержащую $S_3(n-2)$ точек.

Если сложить из фигур F_1 , F_2 и F_3 исходную фигуру, то одна вершина пятиугольника окажется накрытой дважды, поэтому

$$S_3(n) + S_3(n) + S_3(n-2) = S_5(n) + 1,$$

$$S_5(n) = S_3(n) + S_3(n) + S_3(n-2) - 1.$$

Заменяя в формуле $S_3(n) = \frac{n(n+1)}{2}$ число n на число $n-2$, получим:

$$S_5(n) = 2 \frac{n(n+1)}{2} + \frac{(n-2)(n-1)}{2} - 1. \quad S_5(n) = \frac{n(3n-1)}{2}.$$

Второй способ, который можно распространить на любой тип многоугольных чисел.

Проведем из вершины внешнего пятиугольника две диагонали (рис. 8), тогда можно заметить три треугольные области F_1 , F_2 и F_3 , каждая из которых содержит $S_3(n)$ точек.

Если сложить из фигур F_1 , F_2 и F_3 исходную фигуру, то точки на этих двух диагоналях окажутся накрытыми дважды. Каждая диагональ содержит n точек, поэтому

$$3S_3(n) = S_5(n) + 2n,$$

$$S_5(n) = 3S_3(n) - 2n,$$

$$S_5(n) = \frac{n(3n-1)}{2}.$$

На рис. 9 представлена последовательность правильных шестиугольников. Подсчитывая число точек в них, получим шестиугольные числа 1, 6, 15, 28, 45, ...

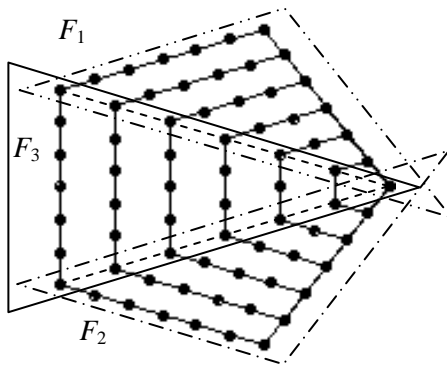


Рис. 8

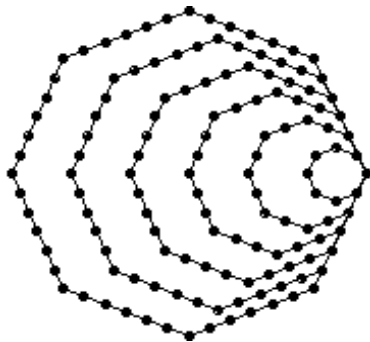


Рис. 9

Докажите, что для шестиугольных чисел выполняется рекуррентное равенство $S_6(n+1) = S_6(n) + 4n + 1$, где $S_6(1) = 1$ и шестиугольные числа определяются формулой $S_6(n) = n(2n - 1)$.

Определите аналогично m -угольные числа, используя последовательность правильных m -угольников (рис. 10).

Докажите, что для m -угольных чисел выполняются равенства: $S_m(n+1) = S_m(n) + (m-2)n + 1$, где $S_m(1) = 1$.

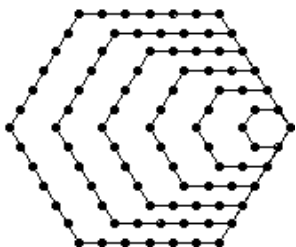


Рис. 10

$$S_m(n) = \frac{n[(m-2)n - m + 4]}{2}.$$

Изучите другие свойства фигурных чисел в статье [10].

В методическом пособии будет рассмотрен метод математической индукции, который применяется для доказательства различных равенств.

2^й этап — Разработка математической модели

Рассмотрим математическое моделирование фигуры, иллюстрирующей фигурные числа.

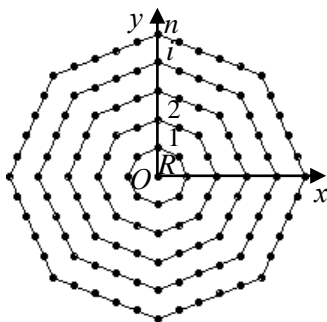


Рис. 11

Вначале упростим задачу и построим последовательность из n правильных m -угольников, имеющих общий центр (рис. 11).

На рисунке указана нумерация многоугольников от 1 до n . Числу 1 соответствует наименьший m -угольник.

Исходные параметры для моделирования:

n — число многоугольников,

m — число сторон многоугольника,

R — радиус окружности, описанной около многоугольника с номером 1.

Поставленную задачу сведем к трем более простым задачам:

А) Построить правильный m -угольник с номером i , имеющий радиус описанной окружности iR .

В) Построить на этом многоугольнике точки на сторонах.

С) Изменяя номер i , построить последовательность подобных многоугольников с общим центром.

Для решения задачи А) необходимо определить координаты вершин многоугольника.

Занумеруем вершины правильного m -угольника согласно рис. 12. Центральный угол, A_0OA_1 опирающийся на сторону многоугольника, равен $2\pi/m$.

Для вершины A_j получаем угол $A_0OA_j = 2\pi j/m$, тогда

$$A_j \left(iR \cos \frac{2\pi j}{m}; iR \sin \frac{2\pi j}{m} \right).$$

$$\text{Аналогично } A_{j+1} \left(iR \cos \frac{2\pi(j+1)}{m}; iR \sin \frac{2\pi(j+1)}{m} \right).$$

Для решения задачи В) необходимо разделить каждую сторону многоугольника с номером i на i равных частей.

Пусть K — середина стороны A_0A_1 многоугольника, тогда из прямоугольного треугольника OA_0K получаем $A_0K = OA_0 \sin(\pi/m)$.

Сторона правильного многоугольника равна $A_0A_1 = 2iR \sin(\pi/m)$.

Рассмотрим точки $B_0 = A_j$, $B_1, \dots, B_j = A_{j+1}$ на стороне A_jA_{j+1} (рис. 13). Для произвольной точки

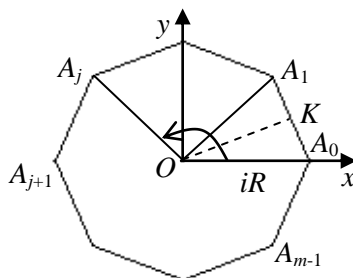


Рис. 12

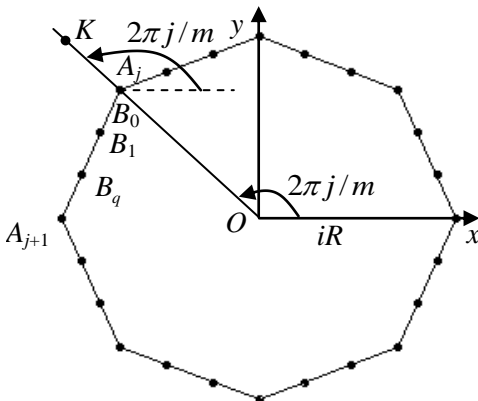


Рис. 13

$B_q \quad \overline{OB_q} = \overline{OA_j} + \overline{A_jB_q}$. Для произвольной точки K на продолжении радиуса OA_j получаем $\angle KB_0B_q = \pi/2 + \pi/m$, следовательно, вектор $\overline{A_jB_q}$ образует с осью Ox угол, равный $u = 2\pi j/m + \pi/2 + \pi/m$.

Для вектора $\overline{A_jB_q}$ имеем координаты:

$$\overline{A_jB_q} = (2iR \sin(\pi/m) \cos u; 2iR \sin(\pi/m) \sin u).$$

Переходим к компьютерному моделированию.

3^й этап — Подготовка инструментария для программы

Чтобы создать компьютерную программу, необходимо четко понимать, как будет реализован программно каждый шаг алгоритма.

Для высокой точности вычислений константу π в дальнейшем будем задавать равной $4\arctg 1$, т.к. $\operatorname{tg}(\pi/4) = 1$.

Для изображения отрезка и окружности используем операторы:

LINE $(x_1, y_1) - (x_2, y_2)$, B , BF — оператор
коорд. противоположных вершин цвет прямоугольн. закрашка прямоугольн.

рисования отрезка (прямоугольника при указании B или заливки прямоугольника при указании BF).

Для построения отрезка часто используем оператор

LINE $(x_1, y_1) - (x_2, y_2)$.
коорд. противоположных вершин

CIRCLE (x , y), \dots , \dots , \dots , \dots , \dots —
координаты центра радиус цвет дуги начало дуги конец дуги коэффициент сжатия

оператор рисования окружности (дуги эллипса при указании начала и конца дуги).

PSET (x, y) — оператор, рисующий точку (x, y) на экране.

Для создания циклов будем использовать следующий блок операторов

FOR $i = 1$ TO 10 STEP 0.1 — начинается цикл

.....

NEXT i — закрывается цикл

— оператор цикла по переменной i , принимающей значение от 0 до 10 с шагом изменения 0,1.

IF $x \geq 0$ THEN $a = \sqrt{x}$

ELSE

PRINT *значение квадратного корня не существует*

— условный оператор, который при выполнении условия $x \geq 0$ вычисляет значение квадратного корня из неотрицательного числа или сообщает о невозможности вычисления квадратного корня из отрицательного числа.

Таблица значений простейших символов и функций

Символ	Значение	Символ	Значение
+	сложение	COS (x)	$\cos x$
—	вычитание	SIN (x)	$\sin x$
*	умножение	TAN (x)	$\operatorname{tg} x$
/	деление	ATN (x)	$\operatorname{arc} \operatorname{tg} x$
\	целочисленное деление	EXP (x)	e^x
^	возведение в степень	LOG (x)	$\ln x$
MOD	нахождение остатка при делении	SQR (x)	\sqrt{x}
		ABS (x)	$ x $

4^й шаг — Создание экранной формы

Открываем Visual Basic 6.0 и находим на экранном окне изображение (рис. 14).

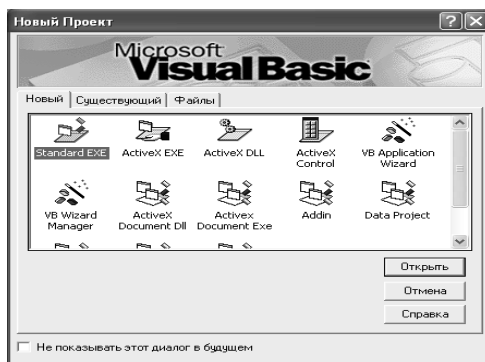


Рис. 14

Выделяем Standard EXE и получаем окно (рис. 15).

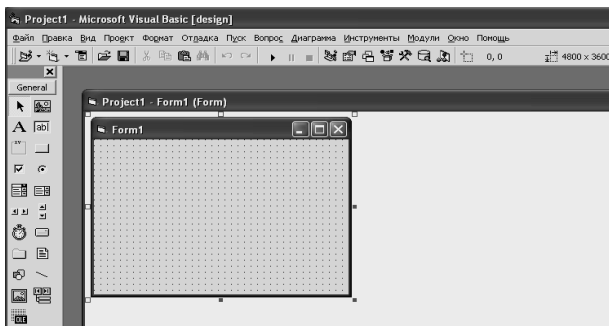


Рис. 15

Устанавливаем указатель мышки на край формы (form 1) и увеличиваем размеры формы до необходимых. В правом верхнем углу формы можно следить за их изменением в цифровом формате.

С левой стороны от экранной формы находим Окно инструментов (General) и на нем Командную кнопку (Command Button) (на рис. 16 она указана стрелкой).

После ее выделения и двукратного нажатия на экранной форме появляется надпись Command 1. Перетаскиваем ее в правый нижний угол формы.

С правой стороны главной панели инструментов находим окно Параметры (рис. 17). В левом столбце появилась выделенная надпись Caption. В правом столбце название Command 1 можно исправить на русифицированный вариант — Пуск.



Рис. 16

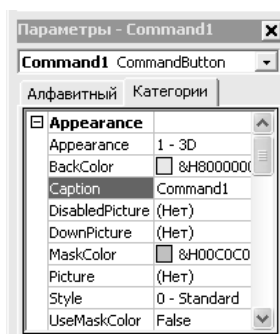


Рис. 17

Становимся указателем на форму и щелчком фиксируем это изменение.

Переходим к написанию программы.

Щелкнув дважды по кнопке *Пуск* на форме, видим на экране появившееся окно для написания кода (рис. 18).

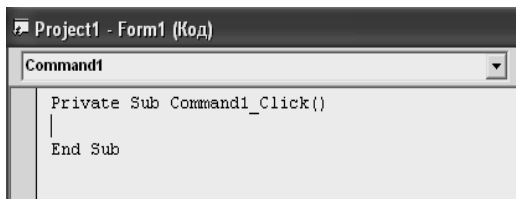


Рис. 18

Форму для кода можно вызвать другим путем, выделив *Вид* и далее *Код* на главной панели инструментов.

Между двумя заготовленными строками на экране фиксируем необходимые константы, операторы построения отрезков и окружностей, элементы цикла.

5^й шаг — Тренинг по разработке простейших программ

Прежде чем написать программу, содержащую несколько вложенных циклов, полезно сформировать простейшие навыки и увидеть результаты своей деятельности на экране компьютера.

Напишем программу построения нескольких горизонтальных отрезков, применяя для каждого отрезка свой стиль рисования (рис. 19).

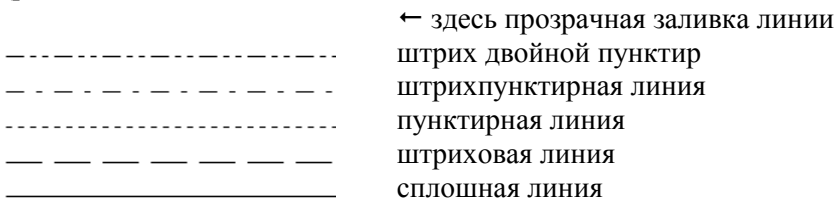


Рис. 19

Программа на языке **Visual Basic 6.0** имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 2000: y0 = 2000: a = 2500: h = 400
```

```

For i = 0 To 5
DrawStyle = i      'выбор типа линии
Line (x0, y0 + i * h)-(x0 + a, y0 + i * h)
Next i
End Sub

```

Пояснения к программе:

Private Sub Command1_Click() — начало программы;

End Sub — завершение программы;

Form1.Scale (0, Form1.Height)-(Form1.Width, 0) — оператор вводит на экране декартову систему координат, т.е. ось Ox направлена слева направо, а ось Oy — снизу вверх;

x_0, y_0 — координаты точки на экране, из которой начинаем построения;

a — длина отрезка;

h — расстояние между отрезками;

i — переменная цикла для построения нескольких отрезков.

Изменение ординат концевых точек отрезка происходит по формуле $y_0 + ih$. Тип линии изменяется в программе с помощью переменной i .

Значения свойств DrawStyle:

DrawStyle = 0 — сплошная заливка;

DrawStyle = 1 — штриховая линия;

DrawStyle = 2 — пунктирная линия;

DrawStyle = 3 — штрихпунктирная линия;

DrawStyle = 4 — штрих — двойной пунктир;

DrawStyle = 5 — прозрачная заливка линии.

Кроме шести стандартных типов линий можно самостоятельно конструировать моделирование типа линии. Такая задача будет рассмотрена в § 8.

Чтобы запустить программу, достаточно нажать кнопку с изображением треугольника (поясняющая надпись — Старт) на главной панели и кнопку *Пуск* на экранной форме.

После просмотра рисунка необходимо нажать кнопку с изображением заштрихованного квадрата (Конец). Экранная форма отойдет на второй план и только после этого можно вносить исправления в код.

Первый результат получен. Его нужно сохранить. Начинаем закрывать экранное окно. Компьютер спрашивает: Сохранить ли проект? Отвечаем: Да и указываем имя, под которым сохраняем.

При первом написании имени программы в окне желательно скопировать написанное имя, чтобы вставить его при втором запросе и при переносе кода программы в текст.

Рассмотрим обращение к написанной программе. Открываем Visual Basic 6.0, находим надпись *Существующий* и открываем свой файл. Первые попытки усовершенствовать этот файл могут оказаться не всегда успешными. Сохраните этот файл под другим именем и начинайте экспериментировать.

Программа на языке **Pascal**:

```
program otreski;
uses graph, crt;
var a,h,x0,y0,i : integer;
gd,gm: integer;
Begin
gd:=detect;
initgraph(gd,gm,"");
a:=200; h:=10; x0:=100; y0:=100;
For i:=0 to 4 do
Begin
SetLineStyle(i,1,1); 'выбор типа линии
Line(x0,y0+i*h,x0+a,y0+i*h);
end;
readkey;
CloseGRAPH;
end.
```

Типы линий в этом случае несколько отличаются. Запустите программу и определите типы линий.

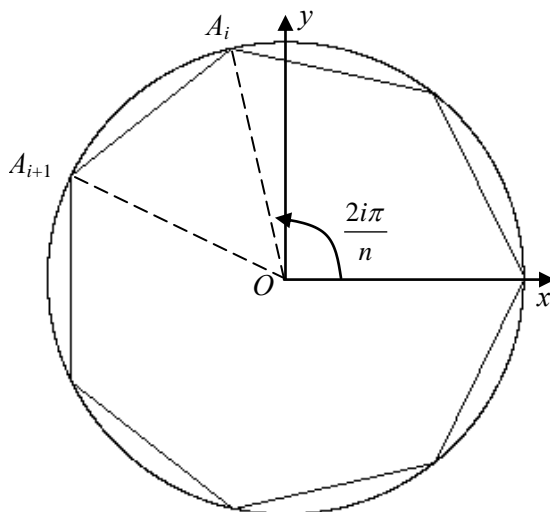


Рис. 20

Для построения окружности и вписанного правильного многоугольника в окружность (рис. 20) приведем две программы.

Программа «Многоугольник» на языке **Visual Basic 6.0**:

```
Private Sub Command1_Click()
R = 2000: n = 7: x0 = 5000: y0 = 4000: pi = 3.14
For t = 0 To 2 * pi Step 0.001
PSet (R * Cos(t) + x0, R * Sin(t) + y0)
Next t
For i = 0 To n - 1
X1 = R * Cos(i * 2 * pi / n): Y1 = R * Sin(i * 2 * pi / n)
X2 = R * Cos((i + 1) * 2 * pi / n): Y2 = R * Sin((i + 1) * 2 * pi / n)
Line (X1 + x0, Y1 + y0)-(X2 + x0, Y2 + y0)
Next i
End Sub
```

Программа на языке **Pascal**:

```
program polygon;
uses graph, crt;
var R,x,y,t,x1,y1,x2,y2,i,n,x0,y0: integer;
    gd,gm: integer;
    pi: real;
Begin
R:=150; n:=7; x0:=200; y0:=200; pi:=3.14;
gd:=detect;
```



```

initgraph(gd,gm,"");
For t:=0 To 628 do
  Begin
    x:=Trunc (R*cos(t/100)); y:=Trunc (R*sin(t/100));
    PutPixel (x+x0, y+y0,15);
  end;
For i:=0 To n do
  Begin
    x1:=Trunc (R*cos (i*2*pi/n)); y1:=Trunc (R*sin (i*2*pi/n));
    x2:=Trunc (R*cos ((i+1)*2*pi/n)); y2:=Trunc (R*sin ((i+1)*2*pi/n));
    Line (x1+x0,y1+y0,x2+x0,y2+y0);
  end;
readkey;
CloseGRAPH;
end.

```

Закрепите полученные сведения для следующего рис. 21, используя один из языков программирования. Сравните свою программу с программой «**Линия квадратов**».

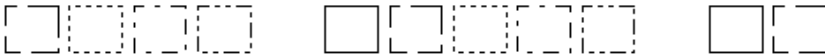


Рис. 21

Изменение координат объекта можно осуществить в цикле добавлением приращения, например, $x = x + h$, где h — величина сдвига по горизонтальной прямой. Заливку квадратов (рис. 22) также осуществим с помощью переменной цикла в программе «**Линия квадратов с заливкой**».

```

Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 1000: y0 = 2000: a = 500: h = 600: x = x0
For i = 0 To 10
  FillStyle = i Mod 8 'тип заливки
  Line (x0 + x, y0)-(x0 + x + a, y0 + a), , B
  x = x + h
Next i
End Sub

```



Рис. 22

Свойства FillStyle:

FillStyle=0 — сплошная заливка цветом;

FillStyle=1 — прозрачная заливка;

FillStyle=2 — горизонтальная штриховка;

FillStyle=3 — вертикальная штриховка;

FillStyle=4 — диагональная штриховка прямыми из верхнего
левого угла в нижний правый угол;

FillStyle=5 — диагональная штриховка прямыми из нижнего
левого угла в верхний правый угол;

FillStyle=6 — горизонтально-вертикальная штриховка;

FillStyle=7 — диагональная пересекающаяся штриховка.

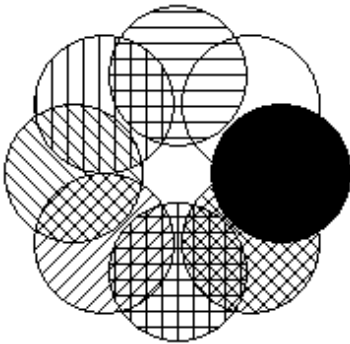


Рис. 23

Центры восьми заполненных (залитых) окружностей расположены на окружности радиуса R .



Рис. 24

На рис. 23 представлено наложение стандартных заливок, заданных для различных окружностей.

```
Private Sub Command1_Click()  
    Form1.Scale (0, Form1.Height)-  
        (Form1.Width, 0)  
    x0 = 4000: y0 = 4000: pi = 4 * Atn(1): R =  
        800  
    For i = 0 To 7  
        FillStyle = i Mod 8  
        Circle (x0 + R * Cos(i * 2 * pi / 8), y0 + R  
            * Sin(i * 2 * pi / 8)), 2 * R / 3  
    Next i  
End Sub
```

Для фиксации размеров точки (рис. 24) на экране компьютера можно воспользоваться свойством DrawWidth. По умолчанию DrawWidth равно 1.

В следующей программе построена последовательность точек различного размера.

```
Private Sub Command1_Click()  
    Line (0, 0)-(12000, 10000), QBColor(15)  
    For i = 1 To 10  
        DrawWidth = i  
        PSet (400 + i * 300, 400)  
    Next i  
End Sub
```

Замечание. Во второй строке программы установлена заливка экранной формы размером 12000×10000 белым цветом, чтобы потом, при вставке скопированного рисунка в текст, фон рисунка был белым.

Замечание. Фон рисунка можно сделать белым другим способом, без использования этого оператора. Щелкнув мышкой по форме, активизируем форму. В разделе Параметры находим BackColor и задаем соответствующий фон заливки формы.

Напишите программу построения квадратов с различной шириной контура (рис. 25).



Рис. 25

6^й шаг — Разработка кода для программы

Перейдем к компьютерному моделированию рассмотренной ранее задачи о центральных многоугольных числах.

Для решения поставленной задачи нужно применить вложенные циклы.

Вначале программируем решение задачи В) — построить точки на стороне многоугольника.

Предположим, что номер i многоугольника и номер j стороны многоугольника фиксированы, тогда необходимо построить $i + 1$ точку на стороне $A_j A_{j+1}$. Для лучшей видимости точек применим изображение точек в виде расширяющейся совокупности окружностей с центрами в найденной точке B_q на стороне $A_j A_{j+1}$.

```
For q = 0 To i
  x = i * r * Cos(j * 2 * pi / m) + q * a * Cos(j * 2 * pi / m + (m + 2) * pi / 2 / m)
  y = i * r * Sin(j * 2 * pi / m) + q * a * Sin(j * 2 * pi / m + (m + 2) * pi / 2 / m)
  For s = 2 To 30
    Circle (x0 + x, y0 + y), s
  Next s
Next q
```

Замечание 1. В программе пока не указаны атрибуты внешней оболочки цикла, т.е. объявление программы, значения параметров.

Пока основное внимание сосредоточено на компьютерном моделировании точек на стороне с помощью цикла.

Замечание 2. Вместо цикла для построения совокупности окружностей можно применить оператор, изменяющий толщину изображаемого объекта — `DrawWidth = 3`. Итак, цикл по переменной s можно заменить операторами:

```
DrawWidth = 3
Pset (x0+x,y0+y)
```

В дальнейшем придется строить стороны многоугольника тонкой линией, поэтому при выходе из цикла с переменной q в этом случае необходимо вставить оператор `DrawWidth = 1`.

Включаем задачу В) в задачу А) — изменяя номер j стороны, построим правильный m -угольник с фиксированным номером i и радиусом описанной окружности iR .

```
For j = 0 To m - 1
Line (x0 + i * r * Cos(j * 2 * pi / m), y0 + i * r * Sin(j * 2 * pi / m))-_
(x0 + i * r * Cos((j + 1) * 2 * pi / m), y0 + i * r * Sin((j + 1) * 2 * pi / m))
For q = 0 To i
x = i * r * Cos(j * 2 * pi / m) + q * a * Cos(j * 2 * pi / m + (m + 2) * pi / 2 / m)
y = i * r * Sin(j * 2 * pi / m) + q * a * Sin(j * 2 * pi / m + (m + 2) * pi / 2 / m)
For s = 2 To 30
Circle (x0 + x, y0 + y), s
Next s, q
Next j
```

Переходим к завершающей стадии — решению задачи С) — изменяя номер i , построим последовательность подобных многоугольников с общим центром.

Включаем задачу В) с задачей А) в задачу С), т.е. вставляем ранее рассмотренные циклы как внутренние в цикл с переменной i .

```
r = 300: m = 8: n = 5: x0 = 4000: y0 = 4000: pi = 4 * Atn(1)
For i = 0 To n
For j = 0 To m - 1
Line (x0 + i * r * Cos(j * 2 * pi / m), y0 + i * r * Sin(j * 2 * pi / m))-_
(x0 + i * r * Cos((j + 1) * 2 * pi / m), y0 + i * r * Sin((j + 1) * 2 * pi / m))
For q = 0 To i
x = i * r * Cos(j * 2 * pi / m) + q * a * Cos(j * 2 * pi / m + (m + 2) * pi / 2 / m)
y = i * r * Sin(j * 2 * pi / m) + q * a * Sin(j * 2 * pi / m + (m + 2) * pi / 2 / m)
For s = 2 To 30
Circle (x0 + x, y0 + y), s
Next s, q, j, i
```

Запустите программу «**Центральные многоугольные числа**» и проведите ее тестирование с различными значениями m сторон многоугольника и значениями n числа многоугольников.

7^й этап — Модификация модели

Рассмотрим окончательное решение рассмотренной ранее задачи о наглядном изображении многоугольных чисел.

Вначале добавим к построенному рисунку оси прямоугольной декартовой системы координат с помощью операторов:

Line (x0, y0)-(x0 + 1.1 *n*R, y0)

Line (x0, y0)-(x0, y0 + 1.1 **n*R).

Для изображения многоугольных чисел необходимо все правильные многоугольники перенести параллельным переносом так, чтобы одна вершина у них была общей и две стороны каждого многоугольника, выходящие из этой точки, были сторонами следующего многоугольника.

В программе «**Многоугольные числа**» найдите изменение в операторе, которое реализует параллельный перенос в одну точку.

Проект, представленный на школьную конференцию, должен содержать достаточно интересную информацию с новым содержанием. Начинаем расширять тему нашей задачи.

1.1. Напишите компьютерную программу, которая иллюстрирует укладку труб (рис. 26) (вид вдоль оси трубы). Сравните с программой «**Укладка кругов в треугольник**».

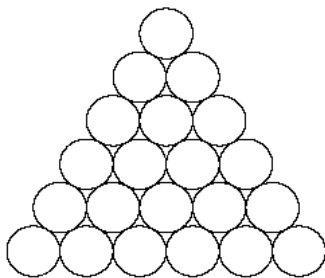


Рис. 26

1.2. Если квадрат деформировать в ромб, то квадратные числа можно иллюстрировать как число кругов, расположенных в ромбе (рис. 27), поэтому квадратные числа можно назвать ромбическими. Напишите программу построения этого рисунка.

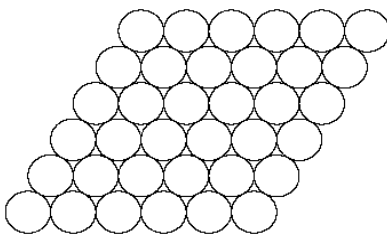


Рис. 27

1.3. На рис. 28 изображены круги в правильном треугольнике и представлено наглядно треугольное число $S_3(8) = 36$. На этом рисунке в ромбе изображено квадратное число $S_4(6) = 36$. Рисунок иллюстрирует возможность перекачать 6 кругов из правильного треугольника в ромб. В этом случае треугольное число оказалось равным квадратному числу. Напишите компьютерную программу, которая перечисляет квадратно-треугольные числа, т.е. числа, являющиеся одновременно квадратными и треугольными. Напишите компьютерную программу построения рис. 28.

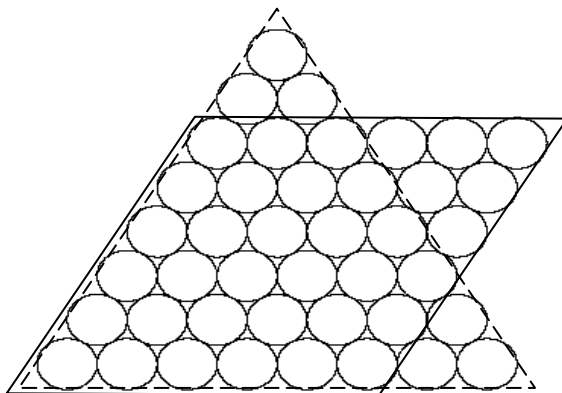


Рис. 28

1.4. Многоугольные числа связаны с важной практической задачей по определению максимального числа равных фигур, которые можно расположить внутри данной фигуры. Познакомьтесь в Интернете с проблемой плотной упаковки шаров внутри данной фигуры.

1.5. Многоугольные числа определяют число кругов, расположенных на плоскости внутри правильного многоугольника, поэтому их иногда называют плоскими фигурными числами. Аналогом многоугольных чисел являются пространственные фигурные числа. Изучите аналитическое задание пространственных фигурных чисел в [10], затем познакомьтесь с методами компьютерной графики в пространстве (глава 2, 3 настоящего пособия) и напишите программу изображения пирамидальных чисел на экране компьютера.

8^й этап — Оформление проекта

При оформлении проекта рекомендуется следующая схема:

1. Актуальность темы.
2. Выдвижение гипотезы.
3. Цель и задачи исследования.
4. Исторический обзор.
5. Используемые методы.
6. Собственный результат.
7. Выводы.
8. Используемая литература.

Рассмотрим новое направление о расположении окружностей вокруг данной окружности.

Окружности S_1, S_2, \dots, S_n образуют венок вокруг окружности S , если каждая окружность S_i касается окружностей S, S_{i-1} и S_{i+1} (Будем считать, что $S_{n+1} = S_1$) (рис. 29).

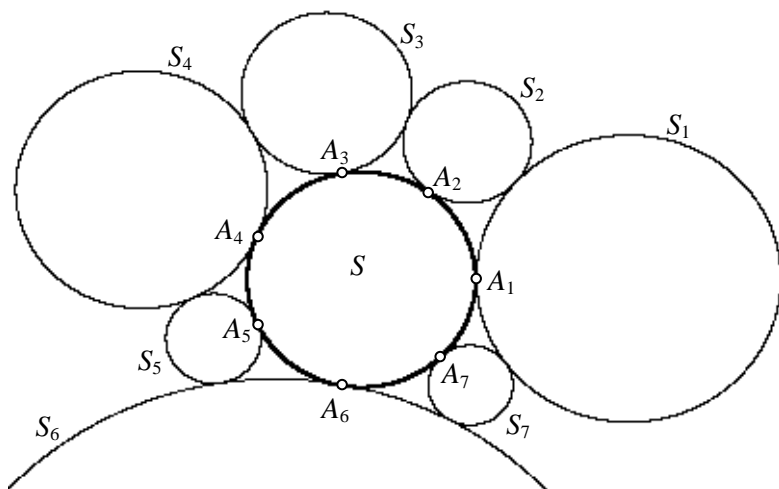


Рис. 29

Вокруг любой окружности всегда можно образовать венок из шести равных ей окружностей (рис. 30).

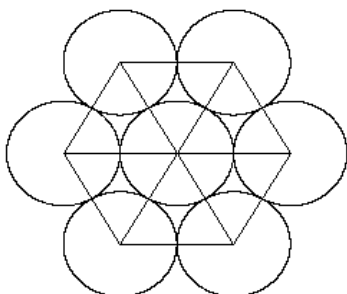


Рис. 30

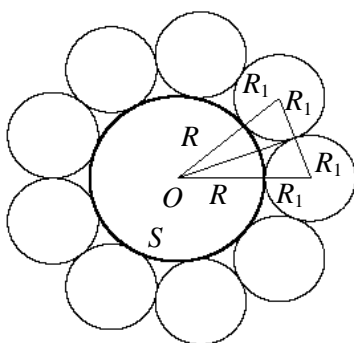


Рис. 31

Вокруг любой окружности всегда можно построить венок из произвольного числа n равных окружностей, где $n \geq 3$. Для радиуса R базовой окружности и радиуса R_1 окружностей венка из прямоугольного треугольника получаем

$$\frac{R_1}{R + R_1} = \sin \frac{\pi}{n} \quad \text{или} \quad R_1 = \frac{R \sin(\pi/n)}{1 - \sin(\pi/n)}.$$

1.6. Напишите программу для построения венка на рис. 31 и сравните с программой «Венок с данным числом окружностей».

Если радиусы окружностей венка произвольные или некоторые из них равны, то познакомьтесь с алгоритмом построением венка в пособии [20].

Пусть A_1, A_2, \dots, A_n — произвольные n точек на окружности S радиуса R (рис. 29). Если n — нечетное число, то всегда можно построить окружности S_1, S_2, \dots, S_n , касающиеся окружности S , соответственно, в точках A_1, A_2, \dots, A_n и такие, что S_2 касается S_1 и S_3 , S_3 касается S_2 и S_4 , ..., S_n касается S_{n-1} и S_1 , S_1 касается S_n и S_2 . Радиус окружности S_1 равен:

$$R_1 = \frac{1}{\frac{1}{R} \pm 2 \frac{A_2 A_3 \cdot A_4 A_5 \cdot \dots \cdot A_{n-1} A_n}{A_1 A_2 \cdot A_3 A_4 \cdot \dots \cdot A_{n-2} A_{n-1} \cdot A_n A_1}} \quad (1)$$

(знак плюс соответствует внутреннему касанию S_1, S_2, \dots, S_n с S , а знак минус — внешнему касанию) [23].

1.7. Напишите программу построения венка окружности, задав в начале программы радиус базовой окружности и параметры для точек касания окружностей.

Сравните ее с программой «**Окружности вокруг окружности**».

1.8. Используя программу, постройте венок из окружностей, расположенных вне базовой окружности (рис. 29), внутри базовой окружности (рис. 32), внутри и вне базовой окружности (рис. 33).

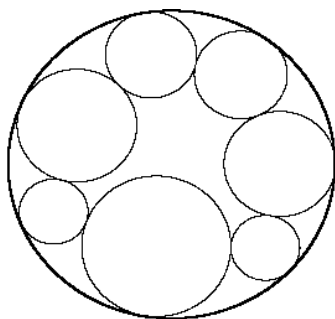


Рис. 32

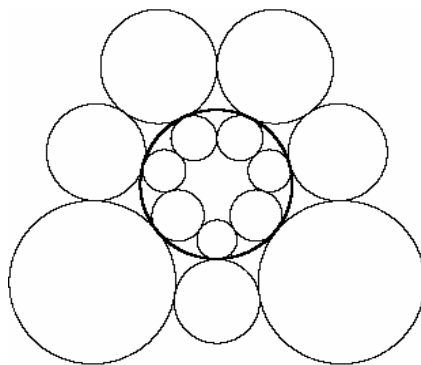


Рис. 33

Некоторые окружности венка касаются не только двух соседних окружностей, но и других окружностей венка (рис. 34, рис. 35).

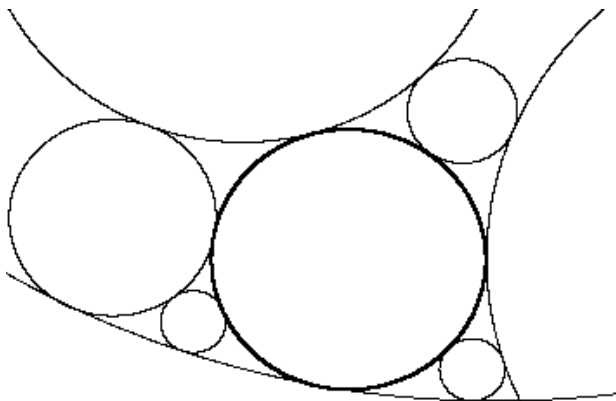


Рис. 34

Если расстояния между точками касания окружностей венка с базовой окружностью существенно различаются, то при одном знаке в формуле (1) могут появиться окружности, касающиеся базовой окружности внутренним и внешним образом (рис. 34—35).

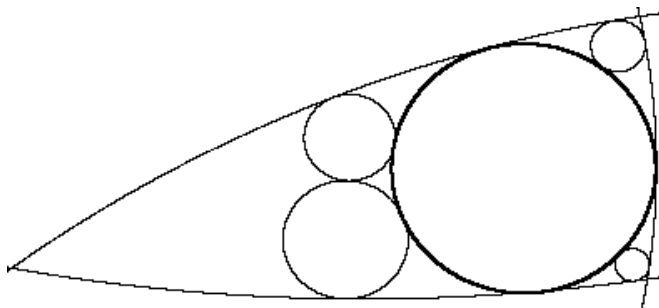


Рис. 35

1.9. Изучите проблему расположения окружностей вокруг данной окружности в пособии [12] и напишите соответствующие программы.

§ 2. Проект «Моделирование систем координат»

Суть любого проекта — предложить решение новой проблемы или решение известной проблемы новым методом. В начале этой книги мы разработаем учебный проект, в котором рассмотрим формирование этапов разработки проекта.

Поиск темы проекта может быть самостоятельным (среди объявленных конкурсов, грантов) либо под влиянием научного руководителя.

Методы решения поставленной проблемы в проекте — изучение литературы по поставленной проблеме (вначале в школьных учебниках других авторов, затем в научно-популярной литературе и, наконец, в специальной литературе), поиск аналогий от известных методов к поставленной задаче, постановка эксперимента по проверке гипотезы, обоснование, т.е. доказательство гипотезы.

Рассмотрим сформулированные выше этапы в конкретной задаче.

В школе подробно изучается прямоугольная декартова система координат. Основное назначение системы координат — каждой точке плоскости ставится в соответствие упорядоченная пара чисел и, наоборот, каждой упорядоченной паре чисел соответствует точка на плоскости. Зачем нужны декартовы координаты? Если в модели некоторой реальной задачи получены условия для нахождения декартовых координат в виде уравнения, системы уравнений или неравенства, то можно изучить свойства множества точек, удовлетворяющих этим условиям, или построить это множество.

Сетка координатной системы видна на экране осциллографа и используется для визуального изучения электрических процессов. В последнее время с экранов телевизоров много говорится о системах GPS и ГЛОНАСС, которые позволяют быстро определить местоположение объекта, снабженного соответствующим приемником, и проложить маршрут между двумя точками. Для пользователя на экране приемника представлена графическая информация в декартовой системе координат, хотя сама информационная система использует более сложную пространственную систему координат, с которой мы познакомимся в одной из глав пособия.

Декартову сетку на экране компьютера можно быстро построить различными способами.

Рассмотрим построение на языке Visual Basic 6.0.

Напомним основные операторы, известные из базового курса по информатике.

Line (x1, y1)-(x2, y2) — оператор построения отрезка с концевыми точками $(x_1; y_1), (x_2; y_2)$.

Иногда мы будем применять этот оператор в более развернутой форме:

LINE $(x_1, y_1)-(x_2, y_2)$, , B | BF — оператор
коорд. противоположных вершин цвет прямоугольник закрашка прямоугольника
 рисования отрезка (прямоугольника при указании B)

FOR i =0 To 10 STEP 0.1 — начинается цикл

.....
 NEXT i — закрывается цикл

— оператор цикла по переменной i , принимающей значение от 0 до 10 с шагом изменения 0,1.

Circle (x, y), R — оператор построения окружности с центром в точке (x, y) , радиусом R .

PSet (x, y) — оператор построения точки (x, y) на экране компьютера.

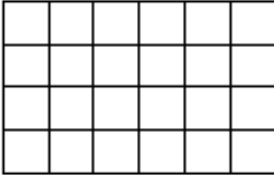


Рис. 1

Для лучшей видимости фигур на экране начало координат лучше сместить в точку (x_0, y_0) .

В следующей программе вначале строится горизонтальный ряд равных отрезков с помощью одного цикла (рис. 1), а затем вертикальный ряд с помощью другого цикла. В циклах применяется параллельный перенос отрезков.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 1000: y0 = 1000: a = 300: n = 4: k = 6
For i = 0 To n
Line (x0, y0)-(x0 + k * a, y0)
y0 = y0 + a: Next i
y0 = 1000
For j = 0 To k
Line (x0, y0)-(x0, y0 + n * a): x0 = x0 + a: Next j
End Sub
```

Число операторов в программе можно сократить, если использовать вложенные циклы и строить оператором не отрезки, а маленькие квадраты.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 1000: y0 = 1000: a = 300: n = 4: k = 6
For i = 0 To n - 1
For j = 0 To k - 1
Line (x0 + j * a, y0 + i * a)-(x0 + (j + 1) * a, y0 + (i + 1) * a), B
Next j, i
End Sub
```

Положение воздушной цели на экране радиолокационной станции (рис. 2) также определяется упорядоченной парой чисел (r, φ) , но смысл чисел в этой паре уже другой.

Первое число r означает расстояние от станции до воздушной цели, второе число φ означает угол, образованный направлением на цель и фиксированным направлением луча, который называется *полярным лучом* [17, с. 203; 5]. Полярная система координат на плоскости широко используется в радиолокации.

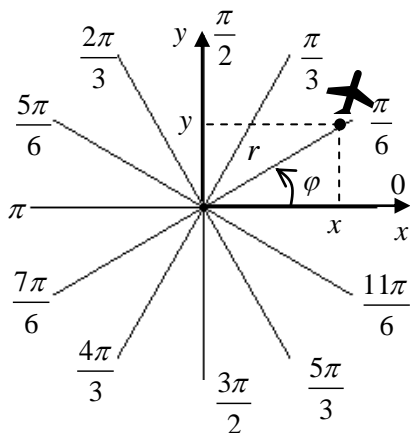


Рис. 2

На радиолокационных станциях также используются эллиптические координаты. Вначале познакомимся с эллипсом и гиперболой, которые представлены в школьном учебнике в качестве дополнительного материала повышенной трудности.

Эллипсом (рис. 3) называется геометрическое место точек плоскости, сумма расстояний от которых до двух данных точек F_1, F_2 , называемых *фокусами*, есть величина постоянная, большая, чем расстояние между фокусами [20, с. 69].

Пусть $F_1 F_2 = 2c$, $MF_1 + MF_2 = 2a$, $a > c$ и система координат выбрана согласно рис. 3, тогда каноническое уравнение эллипса [17, с. 197]

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \text{ где } b^2 = a^2 - c^2.$$

Эллипсы, имеющие общие фокусы (рис. 4), называются *софокусными*. Семейство софокусных эллипсов с фокусами $F_1(c, 0)$, $F_2(-c, 0)$ можно задать аналитически одним уравнением

$$\frac{x^2}{a_1^2} + \frac{y^2}{a_1^2 - c^2} = 1, \quad (1)$$

где $a_1 > c$ и параметр a_1 меняется.

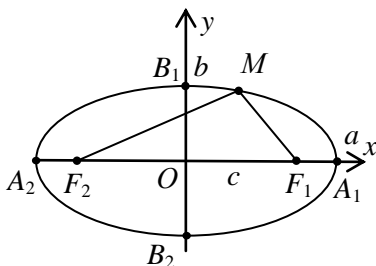


Рис. 3

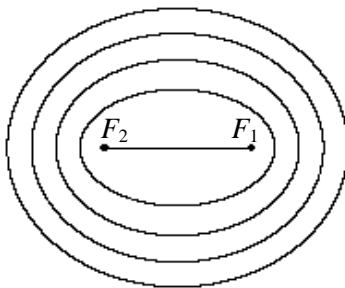


Рис. 4

Программа построения софокусных эллипсов:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 4000: y0 = 3000: c = 600: a1 = c + 200
For i = 1 To 4
For x = 0 To a1 Step 0.1
y = Sqr((a1 ^ 2 - c ^ 2) * (1 - x ^ 2 / a1 ^ 2))
PSet (x + x0, y + y0): PSet (x + x0, -y + y0):
PSet (-x + x0, y + y0): PSet (-x + x0, -y + y0)
Line (x0 - c, y0)-(x0 + c, y0)
Next x
a1 = a1 + 200
Next i
For R = 2 To 24 Step 2
Circle (x0 + c, y0), R: Circle (x0 - c, y0), R
Next R: End Sub
```

Гиперболой называется геометрическое место точек плоскости, модуль разности расстояний от которых до двух данных точек F_1 , F_2 , называемых фокусами, есть величина постоянная, меньшая, чем расстояние между фокусами [17, с. 72].

Пусть $F_1F_2 = 2c$, $|MF_1 - MF_2| = 2a$, $a < c$, тогда для системы координат, выбранной на рис. 5, каноническое уравнение гиперболы [17, с. 197] имеет вид:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \text{ где } b^2 = c^2 - a^2.$$

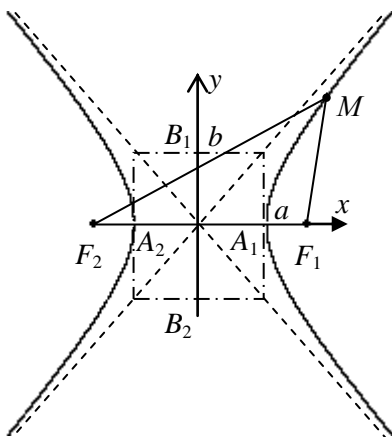


Рис. 5

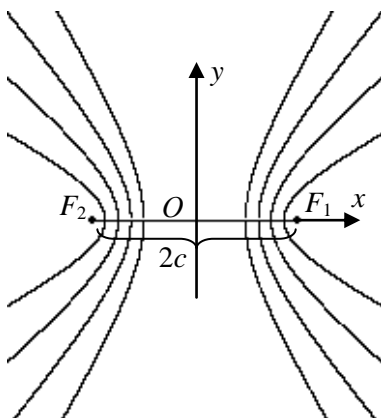


Рис. 6

Гиперболы, имеющие общие фокусы (рис. 6), называются софокусными. Семейство софокусных гипербол с фокусами $F_1(c, 0)$, $F_2(-c, 0)$ можно задать аналитически одним уравнением

$$\frac{x^2}{a_2^2} - \frac{y^2}{c^2 - a_2^2} = 1, \quad (2)$$

где $0 < a_2 < c$ и параметр a_2 меняется.

1.10. Напишите программу построения софокусных гипербол (рис. 6).

В радиолокации большое значение имеет использование системы координат, связанной с семейством эллипсов и семейством гипербол, имеющих одни и те же фокусы (рис. 7, 8).

Пусть радиолокационные станции расположены в фокусах, тогда, посылая сигнал из одного фокуса до воздушной цели и принимая отраженный сигнал во втором фокусе, можно определить сумму расстояний до воздушной цели и разность расстояний.

Покажем, что по этим двум параметрам координаты точки определяются однозначно.

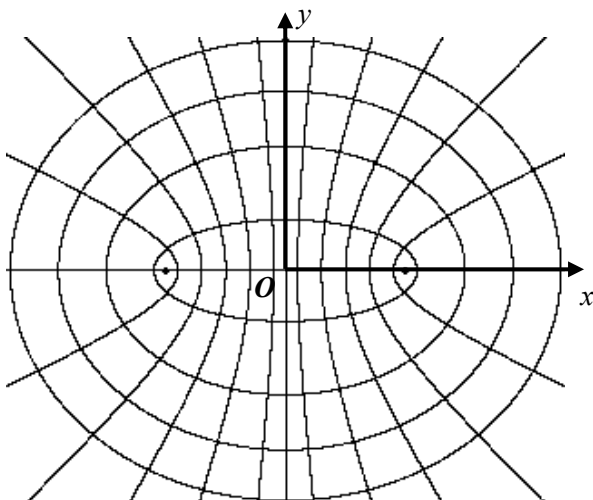


Рис. 7

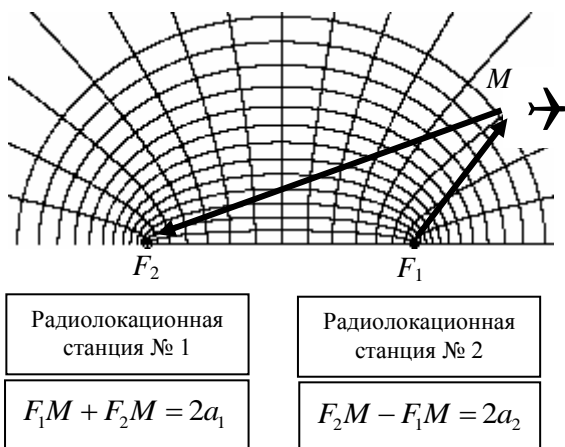


Рис. 8

Умножая уравнение (1) на $\frac{1}{c^2 - a_2^2}$, а уравнение (2) на $\frac{1}{a_1^2 - c^2}$

и складывая полученные уравнения, найдем $x^2 = \frac{a_1^2 a_2^2}{c^2}$.

Умножая уравнение (1) на $\frac{1}{a_2^2}$, а уравнение (2) на $-\frac{1}{a_1^2}$ и складывая полученные уравнения, найдем

$$y^2 = \frac{(a_1^2 - c^2)(c^2 - a_2^2)}{c^2}.$$

Для двух параметров a_1, a_2 при фиксированном значении расстояния между фокусами $F_1F_2 = 2c$ получим четыре точки на плоскости при пересечении эллипса и гиперболы. Например, в первой четверти получаем точку пересечения

$$M_1 \left(\frac{a_1 a_2}{c}; \frac{\sqrt{(a_1^2 - c^2)(c^2 - a_2^2)}}{c} \right).$$

Семейство софокусных эллипсов и гипербол можно задать одним уравнением $\frac{x^2}{\lambda} + \frac{y^2}{\lambda - c^2} = 1$, где c — расстояние от начала координат до фокуса, λ — параметр. При $\lambda > c^2$ уравнение определяет эллипс, при $0 < \lambda < c^2$ — гиперболу. Каждый эллипс ортогонален софокусной с ним гиперболе.

Семейство софокусных эллипсов и гипербол можно построить как два семейства координатных линий на плоскости по формулам $x = \operatorname{ch} v \cos u$, $y = \operatorname{sh} v \sin u$,

где $\operatorname{sh} v = \frac{e^v - e^{-v}}{2}$, $\operatorname{ch} v = \frac{e^v + e^{-v}}{2}$, $v \geq 0, u \in [0, 2\pi)$.

Фиксируя один из параметров u, v и изменяя второй параметр, построим семейство софокусных линий. Параметры u и v называются *эллиптическими координатами на плоскости*.

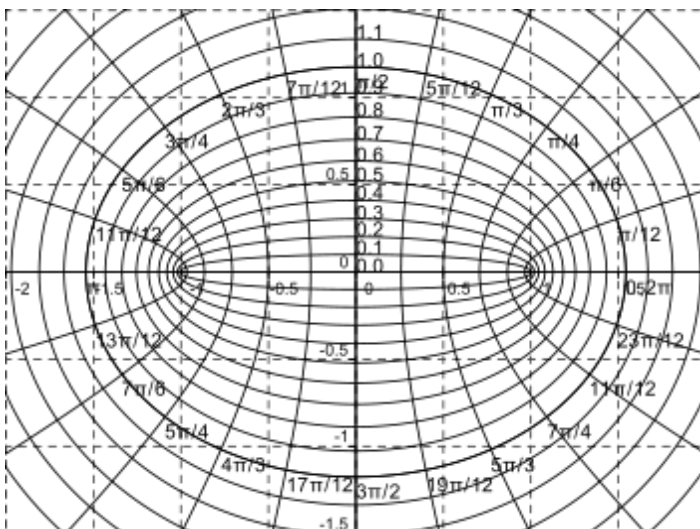


Рис. 9

Изучите рис. 9 из Википедии (свободно распространяемой энциклопедии) в Интернете и поясните значение цифр на рисунке.

1.11. Напишите программу построения софокусных эллипсов и гипербол и сравните ее с программой «**Софокусные эллипс и гипербола**».

Большой интерес представляют задачи с препятствиями.

Пусть на местности, на которой расположено озеро, поселились люди. Известно, что озеро имеет форму круга, но центр круга и радиус круга неизвестны. Поселившиеся люди знакомы с геометрией Евклида и имеют нитку (веревку) достаточно большой длины для геометрических построений. Вдоль границы озера, т.е. по окружности построен забор малой высоты, поэтому с помощью нитки можно проводить измерения на окружности, прикладывая нитку вдоль основания забора.

Как, используя нитку, провести простейшие геометрические построения на местности?

Рассмотрим задачи, решение которых обсуждается далее.

Задача № 1. Построить в произвольной точке границы круга отрезок, являющийся продолжением радиуса окружности, т.е. отрезок, перпендикулярный окружности (рис. 10).

Задача № 2. Построить координатную полярную сеть на плоскости, в которой полярные лучи являются продолжениями радиусов, а полюс совпадает с недоступным центром окружности. Написать компьютерную программу построения полярной системы координат.

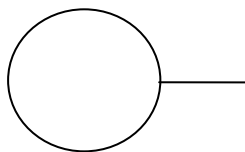


Рис. 10

Задача № 3. Определить радиус окружности и ввести единицу измерения длины, кратную радиусу окружности.

Задача № 4. Построить декартову систему координат, начало которой расположено в недоступном центре окружности. Написать компьютерную программу построения декартовой системы координат.

Задача № 5. Разработать математическое и компьютерное моделирование прокладки кратчайшего курса между двумя данными точками плоскости, если отрезок, соединяющий эти точки, пересекает озеро (рис. 11). Эта задача важна для судовождения, если некоторый морской район, имеющий форму круга, закрыт для плавания.

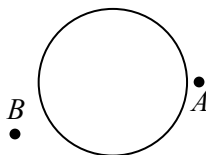


Рис. 11

Условие перпендикулярности отрезков легко реализовать в равнобедренном треугольнике, используя медиану треугольника.

Итак, первая идея. Расположить равнобедренный треугольник ABC так, чтобы медиана треугольника прошла через данную точку K на окружности.

Как это реализовать практически?

От точки K с помощью нитки построим две точки A и B таким образом, чтобы длины дуг KA и KB были равны (рис. 12). Натянем нить над озером из точки A в точку B . Сохраняя длину нити AB , вернем нить на местность, найдем середину M отрезка AB сложением нити пополам (рис. 13). В точках A и B привяжем нить, на которой отмечена середина C . Привяжем в точке M нить достаточной длины и вернем

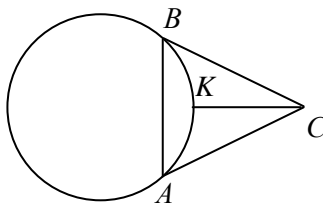


Рис. 12

над озером нить AB с привязанными кусками нитей. После натяжения нитей (рис. 14) получим искомый отрезок KC .

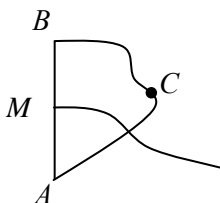


Рис. 13

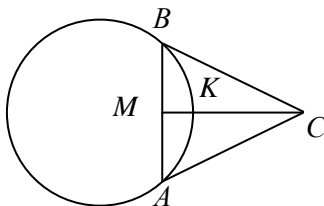


Рис. 14

Предложенный метод содержит много операций, причем натяжение нити над озером может привести к осложнениям при опускании нити в воду. Можно ли упростить этот метод?

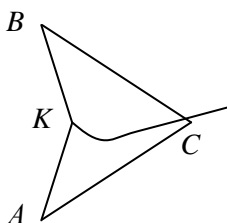


Рис. 15

Криволинейная фигура $AKBC$ содержит два равных отрезка и две равные дуги. Если изготовить такую фигуру из нити, то ее можно перестроить с помощью натяжения нитей в четырехугольник, который называется *дельтоидом* (рис. 15).

Дельтоид предварительно изготавливается, затем прикладывается к окружности и натяжением нити в точках A и B вдоль окружности, а также натяжением нити в точке C получаем криволинейную фигуру $AKBC$ (рис. 16).

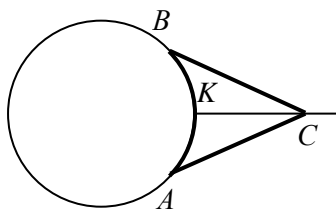


Рис. 16

Важный вывод. Построение перпендикуляра к окружности, а также перпендикуляра к отрезку можно выполнять с помощью дельтоида.

Задача № 2 решается следующим образом. Прикладывая нить вдоль забора, определим длину окружности. Если число равных дуг, на которые нужно разделить окружность, является степенью двойки, то сложением нити пополам определяем длину дуги между двумя соседними точками на окружности.

Если число отрезков n не является степенью двойки, то можно использовать теорему Фалеса для деления длины окружности на заданное число частей (рис. 17).

На одном луче n раз откладываем произвольный отрезок m , а на другом луче — отрезок, равный длине окружности. Соединяем полученные точки A и B .

На угле FAD строим четырехугольник со сторонами, равными отрезку m . Сторона FE ромба будет параллельной отрезку AB . Продолжая отрезок FE до пересечения с прямой OB , получим отрезок BG . Отложим n раз отрезок BG на отрезке OB . Прикладывая нитку, равную отрезку OB к окружности, получим точки, которые делят окружность на n равных частей.

Важный вывод. Построение прямой FG , проходящей через данную точку F и параллельной данной прямой AD , можно выполнить с помощью ромба.

Компьютерное решение задачи намного проще (рис. 18) и представлено в следующей программе.

```
Private Sub Command1_Click()
x0 = 4000: y0 = 4000: R = 600: R1 = 1200: pi = 4 * Atn(1): n = 24
For i = 1 To n
Circle (x0, y0), R
Line (R * Cos(u) + x0, R * Sin(u) + y0)-(R1 * Cos(u) + x0, R1 * Sin(u) + y0)
u = u + 2 * pi / n
Next i
End Sub
```

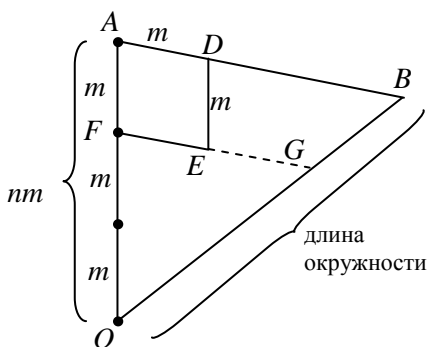


Рис. 17

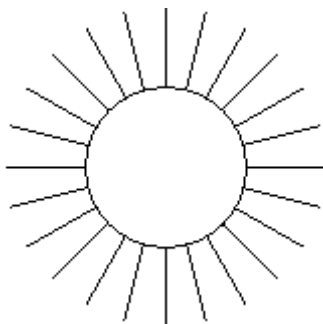


Рис. 18

Во всех предыдущих геометрических построениях не использовался единичный отрезок, т.е. единица измерения. Задача № 3 на измерение радиуса окружности может быть решена несколькими способами.

Первый способ. Если выбрана единица измерения на нитке, то можно определить численное значение длины окружности и, поделив его на число 2π , получим радиус окружности, выраженный в виде числа. Используя единичный отрезок, можно построить отрезок, равный радиусу окружности. Этот способ может привести к приближенным вычислениям.

Рассмотрим геометрические способы без использования единичного отрезка.

Второй способ. Разделим окружность на шесть равных частей. Натянем нить через озеро между двумя соседними точками деления. Полученный отрезок с концами в этих точках является стороной шестиугольника, вписанного в окружность, а значит, он равен радиусу окружности.

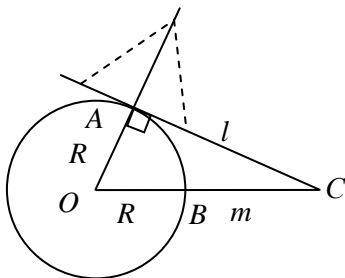


Рис. 19

Третий способ. Используя дельтоид, построим в произвольной точке A отрезок, являющийся продолжением радиуса OA (рис. 19). Используя равнобедренный треугольник, построим в точке A касательную к окружности. Используя снова дельтоид, построим в произвольной точке B отрезок, являющийся продолжением радиуса OB . Получим прямоугольный треугольник OAC , в котором обозначим $AC = l$, $BC = m$, тогда $OA^2 + AC^2 = OC^2$ или $R^2 + l^2 = (R + m)^2$, $2Rm = l^2 - m^2$.

Построение отрезка, равного радиусу (рис. 20), выполнено по формуле $\frac{l+m}{2m} = \frac{R}{l-m}$ на основе теоремы Фалеса.

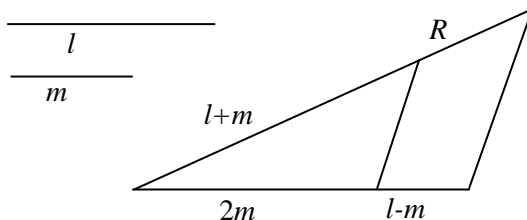


Рис. 20

После определения радиуса окружности можно построить план местности с изображением полярной системы координат или декартовой системы координат. Если радиус окружности получен как отрезок, то теперь естественно выбрать единичный отрезок на плоскости таким образом, чтобы радиус был кратным единичному отрезку (рис. 21).

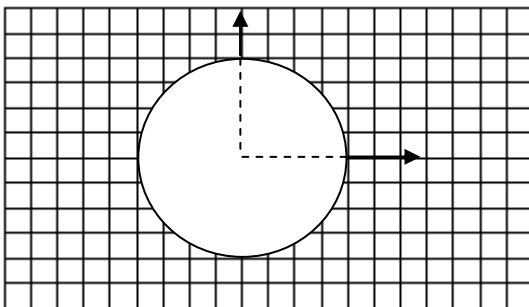


Рис. 21

Компьютерное моделирование этой модели можно осуществить разными способами.

Первый способ. Горизонтальные и вертикальные отрезки координатной сетки рисовать точками с помощью цикла. В программе предусмотреть условный оператор, который рисует точку только в том случае, когда она находится вне данного круга.

Второй способ. Построим операторами вертикальные и горизонтальные отрезки координатной сетки, не обращая внимание на точки круга. Затем на сетке построим круг с данным центром и данным радиусом, заливая его белым цветом.

Рассмотрим задачу о построении кратчайшего маршрута между двумя точками $A(r_1, u_1)$ и $B(r_2, u_2)$ с полярными координатами, если отрезок AB пересекает круг.

Эта задача имеет важное значение для судовождения, если некоторый морской район, имеющий форму круга, закрыт для плавания.

Произвольный маршрут между точками A и B можно сократить в некоторых случаях:

а) на рис. 22 $AC + CD + DB > AE + EF + FB$, где AE и BF — касательные к окружности;

б) на рис. 23 $AE + EF + FB > AG + GH + HB$;

в) на рис. 24 $AG + GH + HB > AK + KL + LM + MN + NB$.

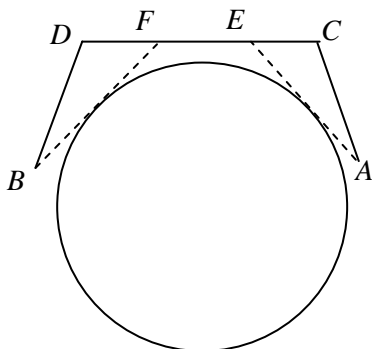


Рис. 22

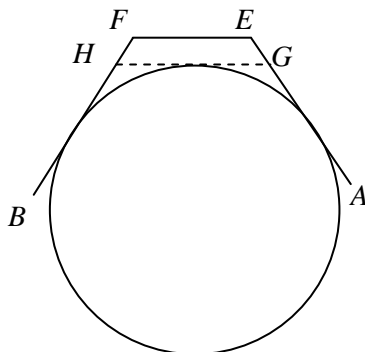


Рис. 23

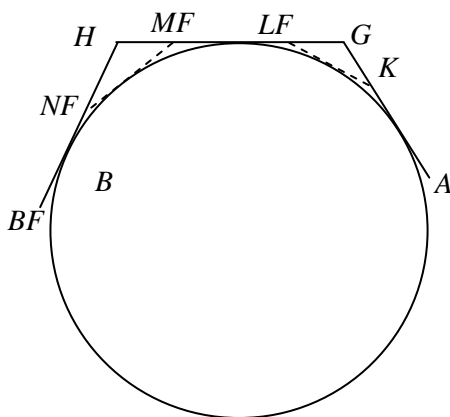


Рис. 24

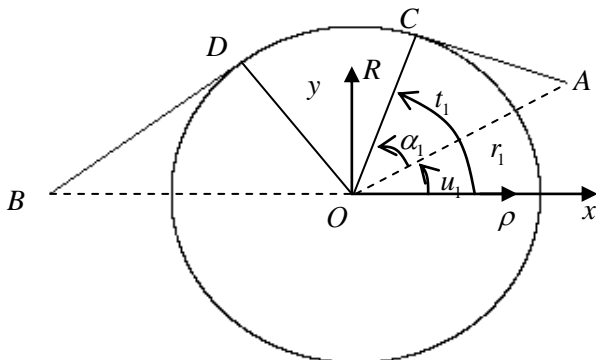


Рис. 25

После различных сокращений маршрута получим кратчайший маршрут $ACDB$, состоящий из двух касательных AC , BD , и дуги окружности CD , соединяющей точки касания (рис. 25).

Из прямоугольного треугольника OAC получаем

$$\operatorname{tg} \angle AOC = \operatorname{tg} \alpha_1 = \frac{AC}{OC} = \frac{\sqrt{r_1^2 - R^2}}{R}.$$

Для точки C на окружности получаем значение параметра

$$t_1 = u_1 + \alpha_1, \quad t_1 = u_1 + \operatorname{arctg} \frac{\sqrt{r_1^2 - R^2}}{R}.$$

Точка C имеет декартовы координаты $x = R \cos t_1, y = R \sin t_1$.

Аналогично для точки D на окружности получаем значение параметра

$$t_2 = u_2 - \alpha_2, \\ t_2 = u_2 - \operatorname{arctg} \frac{\sqrt{r_2^2 - R^2}}{R}.$$

Точка D имеет декартовы координаты $x = R \cos t_2, y = R \sin t_2$.

Следующая компьютерная программа «Маршруты вокруг озера» строит полярную сетку вокруг

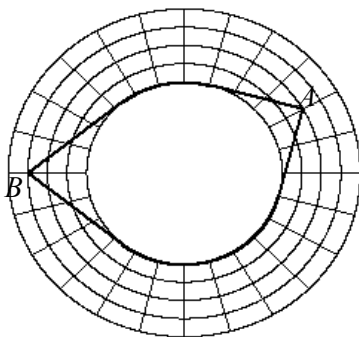


Рис. 26

озера (рис. 26), показывает два маршрута вокруг озера, соединяющие две данные точки *A* и *B*.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 4000: y0 = 4000: R = 1000: pi = 4 * Atn(1): n = 24
R1 = 1400: u1 = pi / 6: r2 = 1600: u2 = pi
For k = 0 To 4
For t = 0 To 2 * pi Step 0.001
PSet ((R + k * 200) * Cos(t) + x0, (R + k * 200) * Sin(t) + y0)
Next t, k
For i = 1 To n
Line (R * Cos(u) + x0, R * Sin(u) + y0)-(R + 800) * Cos(u) + x0, (R + 800) * Sin(u)
+ y0)
u = u + 2 * pi / n
Next i
DrawWidth = 2
t1 = u1 + Atn(Sqr(R1 ^ 2 - R ^ 2) / R)
Line (R * Cos(t1) + x0, R * Sin(t1) + y0)-(R1 * Cos(u1) + x0, R1 * Sin(u1) + y0)
t2 = u2 - Atn(Sqr(r2 ^ 2 - R ^ 2) / R)
Line (R * Cos(t2) + x0, R * Sin(t2) + y0)-(r2 * Cos(u2) + x0, r2 * Sin(u2) + y0)
For t = t1 To t2 Step 0.001
PSet (R * Cos(t) + x0, R * Sin(t) + y0)
Next t
t3 = u2 + Atn(Sqr(r2 ^ 2 - R ^ 2) / R)
Line (R * Cos(t3) + x0, R * Sin(t3) + y0)-(r2 * Cos(u2) + x0, r2 * Sin(u2) + y0)
t4 = u1 - Atn(Sqr(R1 ^ 2 - R ^ 2) / R)
Line (R * Cos(t4) + x0, R * Sin(t4) + y0)-(R1 * Cos(u1) + x0, R1 * Sin(u1) + y0)
For t = t3 To 2 * pi + t4 Step 0.001
PSet (R * Cos(t) + x0, R * Sin(t) + y0)
Next t
End Sub
```

1.12. Улучшите представленную программу, дополнив ее вычислением длин двух маршрутов. Длина дуги окружности равна произведению величины радиуса на величину угла в радианах. Программа должна напечатать на экране информацию о полученных длинах маршрутов.

Для вывода на экран значения переменной *s* используют оператор печати:

```
Print s.
```

Важная информация. Выше предложен простой способ компьютерного моделирования построения касательной из данной точки к данной окружности. Познакомьтесь с другими методами

компьютерного моделирования касательной из точки к окружности [19, с. 200].

Замечание. В научно-популярной литературе по математике и физике поясняется, что в точке $M(x, y)$ линии γ , заданной уравнением $F(x, y) = 0$, нормальный

вектор к линии, т.е. вектор, перпендикулярный к касательной (рис. 27), можно найти в виде $\overline{N}(F_x, F_y)$, где F_x — частная производная от функции $F(x, y)$,

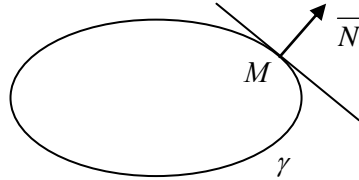


Рис. 27

вычисленная по переменной x при условии, что переменная y зафиксирована, т.е. считается постоянной. Аналогично, F_y — частная производная от функции $F(x, y)$, вычисленная по переменной y при условии, что переменная x зафиксирована, т.е. считается постоянной.

Рассмотрим точку пересечения эллипса и софокусной гиперболы, для которой выполняются равенства:

$$x^2 = \frac{a_1^2 a_2^2}{c^2}, \quad y^2 = \frac{(a_1^2 - c^2)(c^2 - a_2^2)}{c^2}.$$

Для нормального вектора эллипса (1) получаем

$$\overline{N}_1 \left(\frac{2x}{a_1^2}, \frac{2y}{a_1^2 - c^2} \right).$$

Для нормального вектора гиперболы (2) получаем

$$\overline{N}_2 \left(\frac{2x}{a_2^2}, \frac{2y}{c^2 - a_2^2} \right).$$

Найдем скалярное произведение этих векторов:

$$\overline{N}_1 \cdot \overline{N}_2 = \frac{4x^2}{a_1^2 a_2^2} - \frac{4y^2}{(a_1^2 - c^2)(c^2 - a_2^2)} = \frac{4}{c^2} - \frac{4}{c^2} = 0.$$

$$\overline{N}_1 \cdot \overline{N}_2 = 0 \Leftrightarrow \overline{N}_1 \perp \overline{N}_2.$$

Следовательно, касательная к эллипсу перпендикулярна касательной к гиперболе в точке пересечения эллипса и гиперболы.

Это означает, что в каждой точке сеть из эллипсов и гипербол является ортогональной.

Далее в пособии будут рассмотрены системы координат в пространстве и решены содержательные задачи. Например, математическое моделирование пространственных объектов для компьютерного изображения на плоскости экрана.

§ 3. Параметризация линий

Функциональные зависимости $y = kx + b$, $y = ax^2 + bx + c$, $y = a^x$, $y = A \sin(\omega x)$ можно определить единообразно как явное задание функции y от переменной x и записать в виде $y = f(x)$. Чтобы построить такие линии, необходимо для каждого значения переменной x вычислить значение функции $f(x)$ и построить все точки с координатами $(x; f(x))$.

Следующие зависимости $ax + by + c = 0$, $(x - x_0)^2 + (y - y_0)^2 = R^2$ между переменной x и переменной y можно единообразно определить как задание функции в общем виде и записывать в виде $F(x, y) = 0$.

Напоминаем, что уравнение $ax + by + c = 0$, где $a^2 + b^2 \neq 0$, задает на плоскости прямую, а уравнение $(x - x_0)^2 + (y - y_0)^2 = R^2$ задает окружность с центром в точке $M_0(x_0; y_0)$ и радиусом R .

Перейти от конкретного явного задания функции, например, от $y = \ln x$ к общему заданию очень просто, а от общего уравнения к явному выражению функции в некоторых случаях сложно, а иногда и невозможно.

Например, из уравнения $x^2 + y^2 = 9$ получаем $y = \pm \sqrt{9 - x^2}$. Фактически мы должны рассмотреть две функции: $y = \sqrt{9 - x^2}$ и $y = -\sqrt{9 - x^2}$.

Построение точек окружности в этом случае можно осуществить по отдельности, строя полуокружности с помощью двух циклов:

```

Private Sub Command1_Click()
For x = -300 To 300 Step 0.1
PSet (500 + x, 500 + Sqr(90000 - x ^ 2))
Next
For x = -300 To 300 Step 0.001
PSet (500 + x, 500 - Sqr(90000 - x ^ 2))
Next
End Sub

```

А можно реализовать одновременное построение точек окружностей в одном цикле:

```

Private Sub Command1_Click()
For x = -300 To 300 Step 0.1
PSet (500 + x, 500 + Sqr(90000 - x ^ 2))
PSet (500 + x, 500 - Sqr(90000 - x ^ 2))
Next
End Sub

```

Получилась экономия операторов в программе.

Существует третий способ аналитического задания линий на плоскости — с помощью параметрических уравнений $x = x(t)$, $y = y(t)$, где x и y являются функциями некоторого параметра t .

Рассмотрим параметризацию отрезка M_1M_2 , для которого известны координаты концевых точек $M_1(x_1; y_1)$ и $M_2(x_2; y_2)$. Для любой точки $M(x; y)$ на отрезке вектор $\overline{M_1M}$ коллинеарен ненулевому вектору $\overline{M_1M_2}$, поэтому он может быть выражен через этот

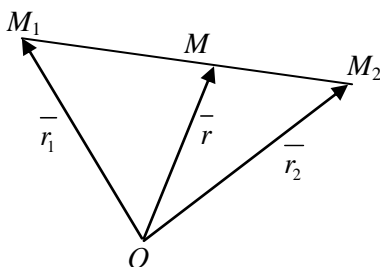


Рис. 1

вектор. Пусть $\overline{M_1M} = t\overline{M_1M_2}$. Обозначим $\overline{OM_1} = \overline{r_1}$, $\overline{OM_2} = \overline{r_2}$, $\overline{OM} = \overline{r}$, тогда $\overline{M_1M} = \overline{r} - \overline{r_1}$, $\overline{M_1M_2} = \overline{r_2} - \overline{r_1}$, $\overline{r} - \overline{r_1} = t(\overline{r_2} - \overline{r_1})$, $\overline{r} = (1-t)\overline{r_1} + t\overline{r_2}$, $t \in [0; 1]$ — векторное уравнение отрезка.

Параметрические уравнения отрезка:

$$x = (1-t)x_1 + tx_2, \quad y = (1-t)y_1 + ty_2, \quad t \in [0;1]. \quad (1)$$

Из векторного равенства $\overline{M_1M} = t\overline{M_1M_2}$ следует, что $t = \frac{M_1M}{M_1M_2}$.

Параметр t в параметрических уравнениях (1) имеет следующий геометрический смысл — это длина отрезка M_1M , измеренная с помощью отрезка M_1M_2 .

Рассмотрим другую параметризацию отрезка M_1M_2 .

Точка M делит отрезок M_1M_2 в отношении λ , если $M_1M/MM_2 = \lambda$. Из векторного равенства $\overline{M_1M} = \lambda\overline{MM_2}$ получаем $\bar{r} - \bar{r}_1 = \lambda(\bar{r}_2 - \bar{r})$, $(1+\lambda)\bar{r} = \bar{r}_1 + \lambda\bar{r}_2$.

$$x = \frac{x_1 + \lambda x_2}{1 + \lambda}, \quad y = \frac{y_1 + \lambda y_2}{1 + \lambda}, \quad \lambda \in (0; \infty). \quad 2$$

Параметр λ в параметрических уравнениях (2) имеет следующий геометрический смысл — это отношение длины отрезка M_1M к длине отрезка MM_2 .

В дальнейшем мы часто будем использовать движение точки по окружности.

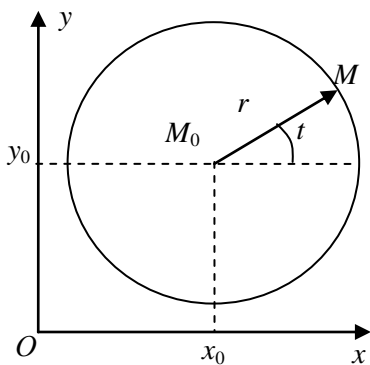


Рис. 2

Рассмотрим окружность $(x - x_0)^2 + (y - y_0)^2 = r^2$ в плоскости Oxy с центром $M_0(x_0; y_0)$ и радиусом r . Пусть произвольная точка $M(x; y)$ окружности образует с положительным направлением оси Ox угол t , тогда для координат вектора $\overline{M_0M}$ получаем (рис. 2): $x - x_0 = r \cos t$, $y - y_0 = r \sin t$.

Уравнения

$$x = x_0 + r \cos t, y = y_0 + r \sin t, t \in [0; 2\pi]$$

называются *параметрическими уравнениями окружности*.

Используя отрезок, можно создавать красивые геометрические фигуры (рис. 3). Поясним принцип построения этой фигуры.

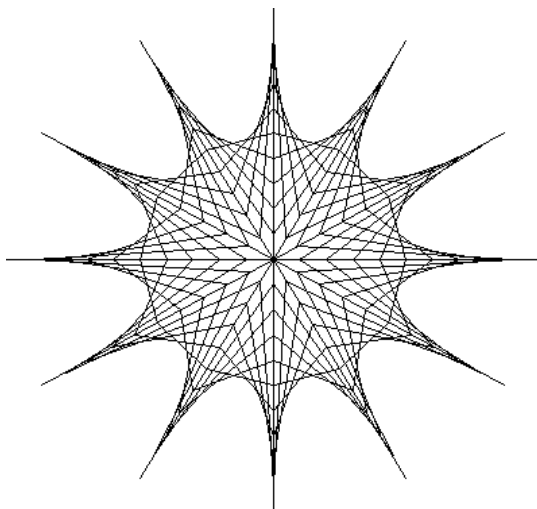


Рис. 3

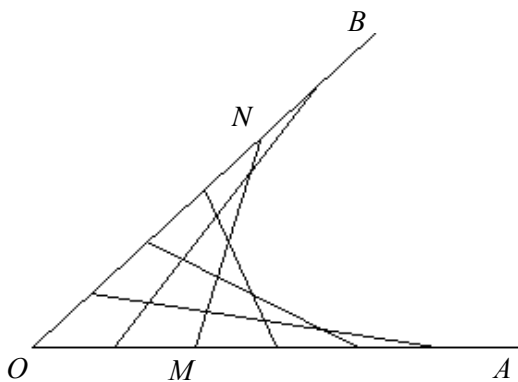


Рис. 4

Пусть $n \in \mathbb{N}, n > 2$. Разделим полный угол на n равных частей и рассмотрим $\angle AOB = 2\pi/n = t_0$, где $OA = OB = R$ (рис. 4).

Построим отрезок MN с концами на сторонах угла AOB таким образом, чтобы $OM = BN$. Изменяя длину отрезка OM , получим семейство отрезков с концами на сторонах угла.

Пусть отрезки OA и OB разделены на k равных частей. Если $OM = jR/k$, где j — натуральное число, $0 \leq j \leq k$, то $ON = R - jR/k$.

При моделировании всей фигуры необходимо представить вращение такого угла, поэтому будем считать, что луч OA образует с осью Ox угол t , тогда луч OB образует с осью Ox угол $t + t_0$. Концевые точки имеют координаты:

$$M = (jR/k \cos(t); jR/k \sin(t)),$$

$$N = (R - jR/k \cos(t + t_0); R - jR/k \sin(t + t_0)).$$

Программа построения фигуры, изображенной на рис. 3, имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): xc = 4000: yc = 4000: R = 3000: n = 12: k = 10
t0 = 2 * pi / n
For i = 0 To 2 * pi Step t0
For j = 0 To R Step R / k
Line (j * Cos(t) + xc, j * Sin(t) + yc)-((R - j) * Cos(t + t0) + xc, (R - j) * Sin(t + t0) + yc)
Next j
t = t + t0
Next i
End Sub
```

1.13. Рассмотрите другие параметризации отрезка, выполнив в формулах (1) следующие замены:

а) $1 - t = u$, б) $t = 2v$, в) $t = w/2$, г) $t = \frac{s}{1+s}$.

В каждом случае укажите область изменения параметра.

Какое множество на плоскости задают уравнения (1), если:

д) $t \in [0; +\infty)$, е) $t \in (-\infty; 0]$,

ж) $t \in (-\infty; +\infty)$?

1.14. Функции

$$x = a \cos t, \quad y = b \sin t, \quad t \in [0; 2\pi] \quad (3)$$

удовлетворяют каноническому уравнению эллипса (рис. 5)

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \text{ где } b^2 = a^2 - c^2.$$

Уравнения (3) называются *параметрическими уравнениями эллипса*.

Используя параметрические уравнения, постройте несколько эллипсов с двумя фиксированными фокусами $F_1(c; 0), F_2(-c; 0)$.

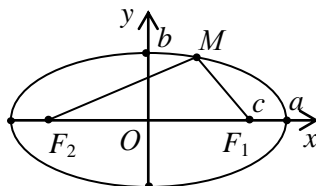


Рис. 5

1.15. Поясните метод образования следующих фигур (рис. 6—13) и напишите компьютерные программы для их построения.

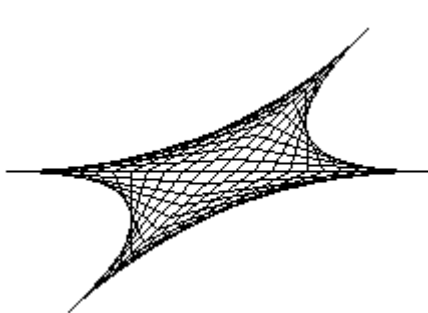


Рис. 6

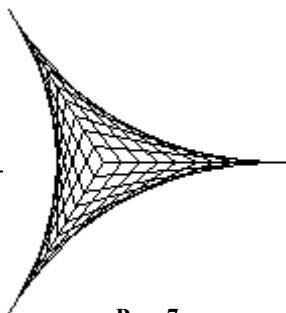


Рис. 7

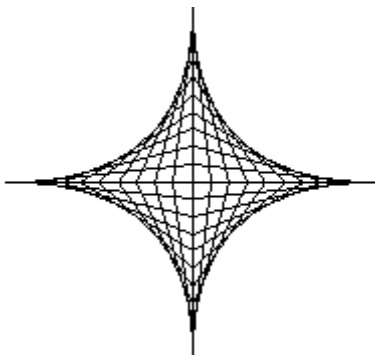


Рис. 8

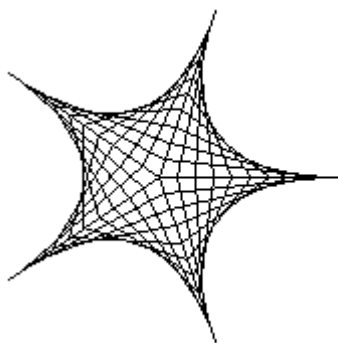


Рис. 9

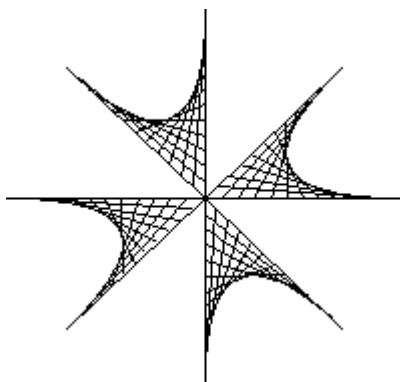


Рис. 10

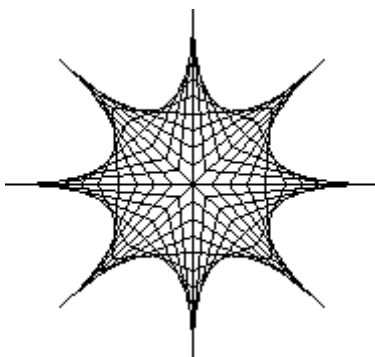


Рис. 11

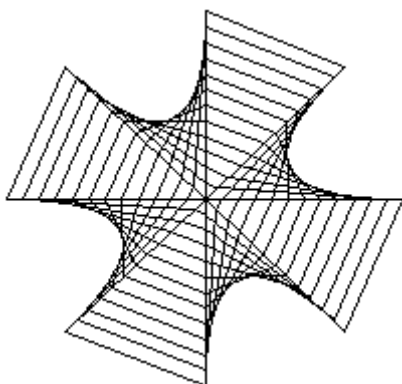


Рис. 12

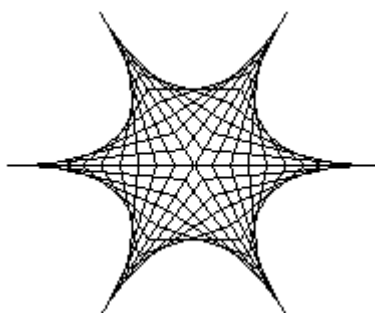


Рис. 13

По двум данным линиям можно построить семейство линий, соединяющих данные линии с помощью непрерывной деформации. Рассмотрим один из методов.

Пусть в первом стакане налита чистая вода, а во втором стакане — ярко окрашенная вода.

В третьем стакане будем образовывать раствор, смешивая некоторое количество воды из первого и второго стаканов. В зависимости от пропорции смешивания в третьем стакане можно получить жидкость, окраска которой может меняться от цвета воды в первом стакане до цвета воды во втором стакане.

Аналогичный процесс можно совершить, смешивая две линии γ_1 и γ_2 (рис. 14).

Пусть линии γ_1 и γ_2 заданы, соответственно, векторными функциями $\bar{r} = \bar{r}_1(t)$ и $\bar{r} = \bar{r}_2(t)$. При фиксированном значении параметра t получим точки $M_1(t)$ и $M_2(t)$ на этих линиях. Соединим отрезком эти точки и рассмотрим на отрезке произвольную точку $M(t)$, для которой

$$\overline{OM} = (1-u)\overline{OM_1} + u\overline{OM_2}, \quad (4)$$

где параметр $u \in [0;1]$.

Если параметр t фиксирован, а параметр u меняется, то получим точки отрезка M_1M_2 . Если параметр u фиксирован, а параметр t меняется, то получим точки одной из линий.

Семейство линий, построенных по формуле (4), называется *пропорциональными линиями*, порожденными данными линиями γ_1 и γ_2 .

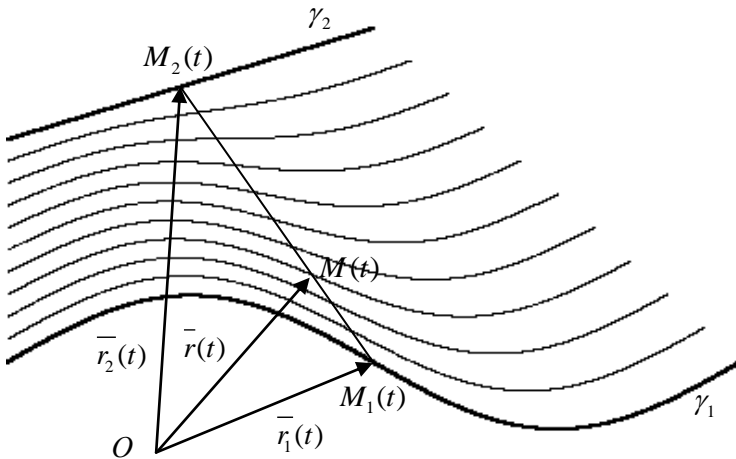


Рис. 14

Программа «Пропорциональные линии» имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1); xc = 100; yc = 1000
For u = 0 To 1 Step 0.1
For t = 0 To 6000
PSet ((1 - u) * t / 2 + u * t + xc, (1 - u) * (2000 + t / 6) + u * 600 * Sin(2 * pi * t / 6000) + yc)
Next t, u
DrawWidth = 2
```

```

For u = 0 To 1
For t = 0 To 6000
PSet ((1 - u) * t / 2 + u * t + xc, (1 - u) * (2000 + t / 6) + u * 600 * Sin(2 * pi * t / 6000) + yc)
Next t, u
End Sub

```

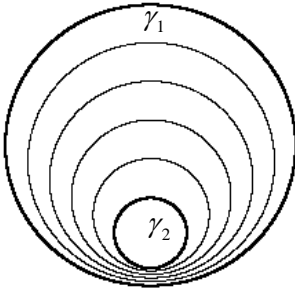


Рис. 15

Для двух окружностей $\gamma_1 : x = acost$, $y = asint$ и $\gamma_2 : x = bcost$, $y = bsint - c$ пропорциональные линии являются снова окружностями (рис. 15). Действительно, из уравнений

$$x = (1 - u)acost + ubcost,$$

$$y = (1 - u)asint + u(bsint - c),$$

или

$$x = [(1 - u)a + ub]cost,$$

$$y = [(1 - u)a + ub]sint - uc,$$

следует, что пропорциональная линия при фиксированном значении параметра u является окружностью с радиусом $R = (1 - u)a + ub$ и центром в точке $(0; -uc)$.

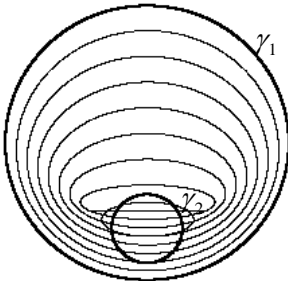


Рис. 16

Замечание. Пропорциональные линии определяются на основе данных векторных функций двух линий. Но одно и то же множество точек на плоскости можно задать различными векторными функциями. Например, данную выше окружность γ_2 можно задать параметрическими уравнениями $\gamma_2 : x = bcost$, $y = -bsint - c$ или равносильными уравнениями $\gamma_2 : x = b\cos(-t)$, $y = b\sin(-t) - c$.

Это означает, что обход точек окружности совершается в обратном направлении, при условии, что $t \in [0; 2\pi]$. Смешивая две векторные функции $x = (1 - u)acost + ubcost$ и $\gamma_2 : x = bcost$, $y = -bsint - c$ при $t \in [0; 2\pi]$, получаем новые пропорциональные линии (рис. 16).

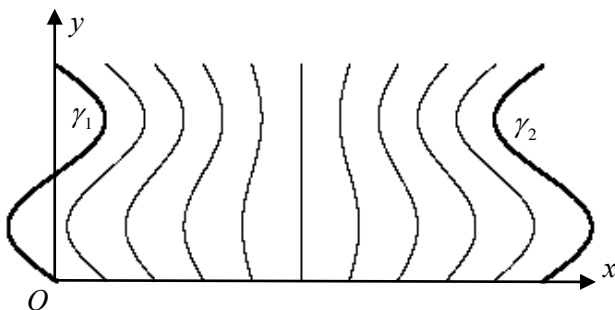


Рис. 17

1.16. На рис. 17 линии γ_1 и γ_2 являются синусоидами. Задайте аналитически эти линии и напишите программу построения пропорциональных линий.

1.17. Докажите, что при смешивании двух векторных функций для отрезков $\overline{r_1((1-t)x_1 + tx_2; (1-t)y_1 + ty_2)}$, $\overline{r_2((1-t)x_1 + tx_3; (1-t)y_1 + ty_3)}$ с общей конечной точкой получаются отрезки с этой же конечной точкой. Напишите программу построения такого семейства отрезков.

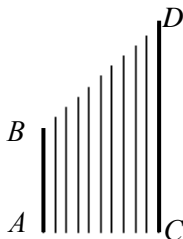


Рис. 18

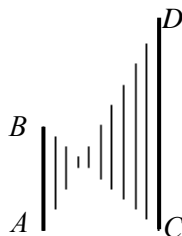


Рис. 19

1.18. При смешивании двух параллельных отрезков могут получиться два различных семейства пропорциональных линий (рис. 18, 19). Поясните эти ситуации.

§ 4. Эквидистантные линии и огибающая линия

В этом параграфе будет показано применение производной функции к изучению семейства линий.

Если клетка для тигра в зоопарке имеет в основании форму прямоугольника $ABCD$, то ограждение вокруг клетки естественно изготовить в виде линии γ_1 , точки которой отстоят на одном

и том же расстоянии h от линии γ (рис. 1). Линия γ_1 в этом примере состоит из четырех отрезков и четырех четвертей

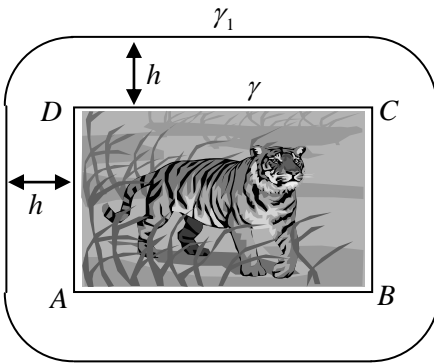


Рис. 1

окружности. Линия γ_1 называется параллельной (или эквидистантной линией) к данной линии γ . Название параллельной частично можно понять, т.к. точки на одной параллельной прямой также находятся на одном и том же расстоянии от другой параллельной прямой. К сожалению, аналогия между параллельными прямыми и параллельными линиями на этом

заканчивается. Далее будет показано, что параллельные линии иногда будут пересекаться, поэтому более естественно употреблять название *эквидистантные линии* или *эквидистанты*.

Рассмотрим определение *эквидистанты*. Пусть плоская линия γ задана векторной функцией $\bar{r} = \bar{r}(t)$ и в каждой точке линии существует касательная. Прямая, проходящая через точку линии

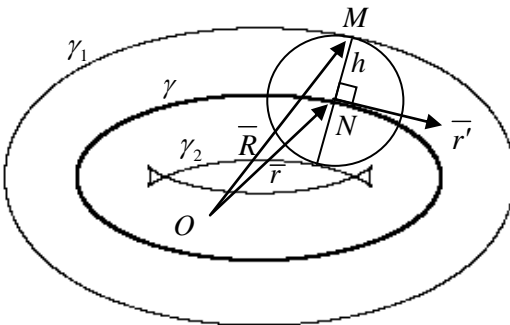


Рис. 2

перпендикулярно касательной, называется *нормалью* к линии.

На нормали, проходящей через точку N , можно рассмотреть два противоположно направленных единичных вектора $\bar{n}_1(t)$ и $\bar{n}_2(t)$. Для произ-

вольного положительного числа h рассмотрим векторы $\overline{NM} = h\bar{n}_1(t)$ и $\overline{NM}' = h\bar{n}_2(t)$ (рис. 2). При изменении параметра t точка M пробегает линию γ_1 , заданную векторной функцией $\bar{R}_1(t) = \bar{r}(t) + h\bar{n}_1(t)$, а точка

вольного положитель-

M' пробегает линию γ_2 , заданную векторной функцией $\overline{R}_2(t) = \overline{r}(t) + h\overline{n}_2(t)$.

Обе линии можно задать одной формулой $\overline{R}(t) = \overline{r}(t) \pm h\overline{n}_1(t)$.

Расстояние между точками линии γ_1 и линии γ по нормали равно h . Расстояние между точками линии γ_2 и линии γ по нормали также равно h . Линия γ называется *базой для эквидистант*, линия γ_1 называется *эквидистантой с высотой h* , а линия γ_2 — *эквидистантой с высотой $(-h)$* при фиксированном векторе на нормали.

Рассмотрим метод определения координат вектора на нормали. В произвольной точке $N(x(t); y(t))$ найдем вектор $\overline{r}'(t)$ с координатами $x'(t), y'(t)$ и отложим его от точки N . Он будет направлен по касательной. Вектор $(y'(t), -x'(t))$ является перпендикулярным к вектору $\overline{r}'(t)$, т.к. их скалярное произведение равно нулю.

Если разделить вектор $(y'(t), -x'(t))$ на его длину, то получим единичный вектор $\overline{n}_1(t) = \frac{(y'(t); -x'(t))}{\sqrt{x'^2(t) + y'^2(t)}}$. Умножая координаты

этого вектора на число h , получим координаты вектора \overline{NM} .

Параметрические уравнения эквидистант:

$$X(t) = x(t) \pm \frac{hy'(t)}{\sqrt{x'^2(t) + y'^2(t)}}, Y(t) = y(t) \mp \frac{hx'(t)}{\sqrt{x'^2(t) + y'^2(t)}}. \quad (1)$$

Пример. В качестве линии γ рассмотрим эллипс, заданный параметрическими уравнениями $x = acost, y = b \sin t$, тогда $x' = -a \sin t, y' = b \cos t, \sqrt{x'^2(t) + y'^2(t)} = \sqrt{a^2 \sin^2(t) + b^2 \cos^2(t)}$.

Параметрические уравнения эквидистант

$$X(t) = acost \pm \frac{hb \cos t}{\sqrt{a^2 \sin^2 t + b^2 \cos^2 t}},$$

$$Y(t) = b \sin t \pm \frac{ha \sin t}{\sqrt{a^2 \sin^2 t + b^2 \cos^2 t}}.$$

Для данного эллипса (он на рис. 3, 4 изображен жирной линией) в зависимости от величин полуосей a , b эллипса и диапазона изменения высот эквидистант получаем различные по форме эквидистанты, которые уже не являются эллипсами.

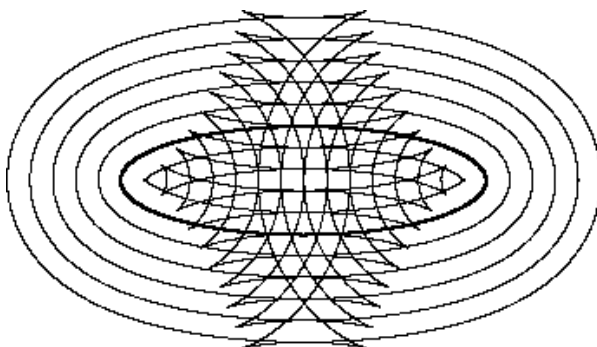


Рис. 3

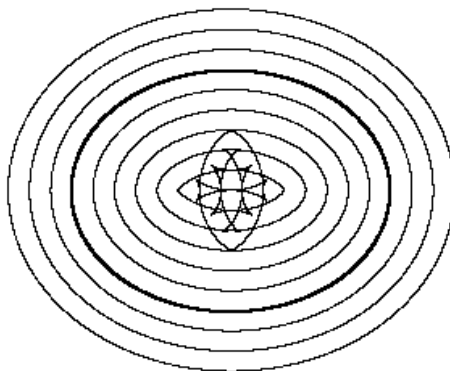


Рис. 4

Следующая программа «Параллельные линии» строит семейство эквидистант.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): xc = 4000: yc = 5000: a = 1500: b = 1200
For h = -1800 To 600 Step 200
For t = 0 To 2 * pi Step 0.001
x = a * Cos(t) + h * b * Cos(t) / Sqr(a ^ 2 * (Sin(t)) ^ 2 + b ^ 2 * (Cos(t)) ^ 2)
```



```

y = b * Sin(t) + h * a * Sin(t) / Sqr(a ^ 2 * (Sin(t)) ^ 2 + b ^ 2 * (Cos(t)) ^ 2)
PSet (x + xc, y + yc): Next t, h
DrawWidth = 2
For t = 0 To 2 * pi Step 0.001
x = a * Cos(t): y = b * Sin(t): PSet (x + xc, y + yc): Next t
End Sub

```

1.19. Вставьте в программу цикл задержки, чтобы выделить процесс построения каждой линии.

Эквидистантную линию с высотой p к данной линии γ можно получить следующим образом. Нужно построить семейство окружностей с центрами на линии γ фиксированного радиуса h . По контуру полученного семейства можно увидеть эквидистантную линию, которую мы в этом случае не строили (рис. 5). Большое количество точек (по одной с каждой окружности и удаленной от γ на расстоянии h) создает видимость линии, хотя на ней построено ограниченное количество точек.

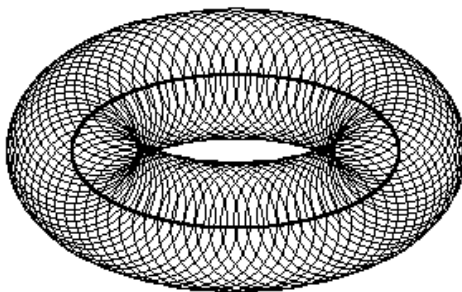


Рис. 5

Огибающей однопараметрического семейства линий на плоскости называется линия, которая в каждой своей точке касается одной из линий этого семейства. Огибающая семейства линий является более общим понятием, чем эквидистантная линия.

Пример 1. На рис. 6 построено несколько линий из семейства окружностей

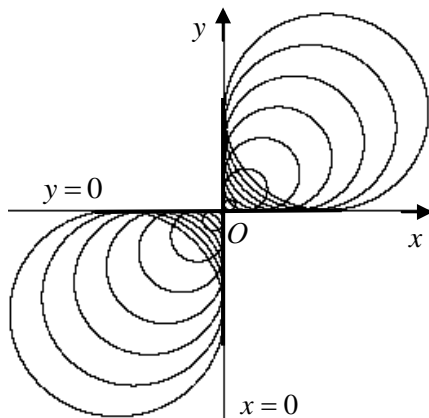


Рис. 6

$$(x-a)^2 + (y-a)^2 = a^2, \quad a \in R \quad (2)$$

Центры окружностей находятся на биссектрисе первого и третьего координатных углов, а радиус каждой окружности равен расстоянию от центра окружности до оси абсцисс.

Если семейство окружностей задано уравнением $f(x, y, \alpha) = 0$, где α — параметр семейства, то координаты точек огибающей удовлетворяют системе уравнений

$$\begin{cases} f(x, y, \alpha) = 0 \\ f'_\alpha(x, y, \alpha) = 0 \end{cases} \quad (3)$$

где $f'_\alpha(x, y, \alpha)$ — производная функции $f(x, y, \alpha)$ по параметру α [3, с. 55; 19, с. 172].

Для семейства линий (2) получаем:

$$f(x, y, \alpha) = (x - \alpha)^2 + (y - \alpha)^2 - \alpha^2 = x^2 - 2x\alpha + y^2 - 2y\alpha + \alpha^2,$$

$$f'_\alpha(x, y, \alpha) = -2x - 2y + 2\alpha.$$

Из равенства $f'_\alpha(x, y, \alpha) = 0$ следует $\alpha = x + y$.

Подставляя значение α в уравнение $(x - \alpha)^2 + (y - \alpha)^2 = \alpha^2$, получаем, что точки огибающей удовлетворяют уравнению $xy = 0$, которое задает пару прямых, т.е. оси координат.

Если считать, что семейство линий (2) при $\alpha = 0$ задает окружность нулевого радиуса, то пара прямых в каждой точке касается одной из данных линий этого семейства.

Программа построения семейства окружностей имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): s0 = 4000: w0 = 5000
Line (s0 - 3000, w0)-(s0 + 3000, w0)
Line (s0, w0 - 3000)-(s0, w0 + 3000)
For i = -1000 To 1000 Step 150
For t = 0 To 2 * pi Step 0.001
x = i * Cos(t) + i: y = i * Sin(t) + i
PSet (x + s0, y + w0)
Next t, i
End Sub
```

Пример 2. В учебном пособии [22, с. 117] рассматривается построение информационной модели движения тела, брошенного под углом к горизонту (рис. 7). Этот пример имеет важное

применение в военном деле, т.к. иллюстрирует зону достижимости при стрельбе из орудия.

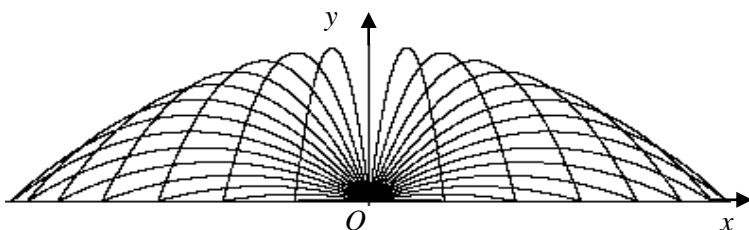


Рис. 7

Пусть тело брошено с начальной скоростью v_0 под углом α к горизонту, тогда дальность полета x и высота y выражаются по формулам:

$$x = v_0 \cos \alpha t, \quad (4)$$

$$y = v_0 \sin \alpha t - gt^2 / 2. \quad (5)$$

Будем считать начальную скорость постоянной величиной для данной задачи и изучим семейство траекторий в зависимости от параметра α .

При фиксированном значении параметра α уравнения (4), (5) являются параметрическими уравнениями от параметра t . Исключим этот параметр, чтобы получить уравнение вида $f(x, y, \alpha) = 0$.

Выражая параметр t из уравнения (4) и подставляя в уравнение (5), получаем

$$y = tg \alpha x - \frac{g}{2v_0^2 \cos^2 \alpha} x^2. \quad (6)$$

При фиксированных значениях α, v_0 это уравнение определяет на плоскости параболу. Изменяя параметр α , т.е. угол наклона к горизонту, под которым бросается тело, получаем семейство линий.

Дифференцируя уравнение (6) по параметру α , получим

$$0 = \frac{1}{\cos^2 \alpha} x - \frac{gx^2 \sin \alpha}{v_0^2 \cos^3 \alpha}. \quad (7)$$

Из уравнения (7) следует $x = 0$ или

$$\operatorname{tg} \alpha = \frac{v_0^2}{gx}. \quad (8)$$

Линия $x = 0$ не является огибающей линией, что очевидно из рис. 7.

Подставим значение $\operatorname{tg} \alpha = \frac{v_0^2}{gx}$, $\frac{1}{\cos^2 \alpha} = 1 + \operatorname{tg}^2 \alpha = 1 + \frac{v_0^4}{g^2 x^2}$ в уравнение (6):

$$y = \frac{v_0^2}{2g} - \frac{gx^2}{2v_0^2}. \quad (9)$$

Уравнение (9) определяет параболу, симметричную относительно оси ординат. До начала ее исследования изучим подробнее данное семейство парабол, заданное параметрическими уравнениями (4), (5).

Определим наибольшую высоту подъема на каждой параболе, используя производную

$$y_t' = v_0 \sin \alpha - gt.$$

Производная обращается в ноль при $t = \frac{v_0 \sin \alpha}{g}$. При этом значении высота достигает максимума, если угол α зафиксирован, т.е.

$$y_{\max}(\alpha) = \frac{v_0^2 \sin^2 \alpha}{2g}.$$

Изменяя параметр α , получаем, что абсолютный максимум достигается при $\alpha = \pi/2$, т.е. при вертикальном бросании вверх и $y_{\max}(\pi/2) = \frac{v_0^2}{2g}$ (рис. 8).

Определим дальность полета.

Момент времени, когда тело упадет на ось Ox , характеризуется условием $y = 0$. Полагая $v_0 \sin \alpha t - gt^2/2 = 0$, получим два значения. Значению $t = 0$ соответствует вылет тела, поэтому выбираем второе значение $t = 2v_0 \sin \alpha / g$.

$$x_{\max}(\alpha) = \frac{v_0^2 \sin 2\alpha}{g}.$$

Абсолютный максимум дальности достигается при $\alpha = \pi/4$ (рис. 8) и равен

$$x_{\max}(\pi/4) = \frac{v_0^2}{g}.$$

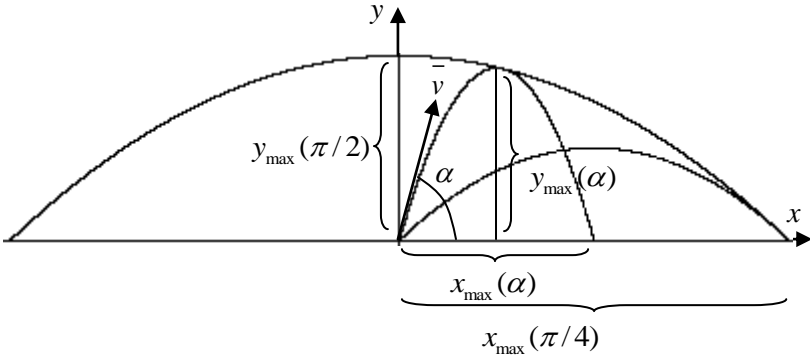


Рис. 8

Легко проверяется, что парабола (9) проходит через точку $(0; v_0^2/(2g))$, т.е. наивысшую точку при стрельбе вертикально вверх, и через точку $(v_0^2/g; 0)$, т.е. наиболее удаленную точку при стрельбе под углом 45° к горизонту.

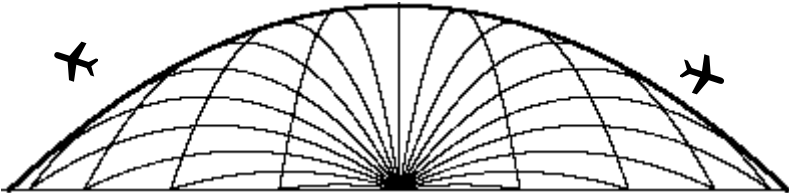


Рис. 9

Парабола (9) является огибающей семейства (6) и называется *параболой безопасности*, т.к. она ограничивает зону обстрела. Выше этой линии полет самолета является безопасным (рис. 9).

Программа «Огибающая парабол» имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): s0 = 4000: w0 = 5000: v0 = 170: g = 9.8
Line (s0 - 4000, w0)-(s0 + 4000, w0)
Line (s0, w0)-(s0, w0 + 4000)
For al = 0 To pi Step pi / 20
t0 = Abs(v0 * Sin(al) / g)
For t = 0 To 2 * t0 Step 0.01
x = v0 * Cos(al) * t
y = v0 * Sin(al) * t - g * t ^ 2 / 2
PSet (x + s0, y + w0)
Next t, al
DrawWidth = 2
For x = -(v0) ^ 2 / g To (v0) ^ 2 / g Step 0.01
y = (v0) ^ 2 / g / 2 - g * x ^ 2 / 2 / v0 ^ 2
PSet (x + s0, y + w0)
Next x
End Sub
```

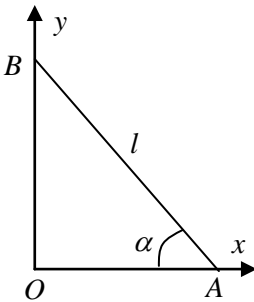


Рис. 10

Пример 3. Отрезок постоянной длины l скользит своими концами по двум взаимно перпендикулярным прямым. Найти огибающую семейства отрезков.

Рассмотрим систему координат, указанную на рис. 10, а в качестве параметра для семейства отрезков выберем угол α , образованный отрезком AB с отрицательным направлением оси абсцисс, тогда $OB = l \sin \alpha$. Для прямой AB получаем уравнение:

$$y = -\operatorname{tg} \alpha x + l \sin \alpha. \quad (10)$$

Дифференцируя по параметру α , получим

$$0 = -\frac{x}{\cos^2 \alpha} + l \cos \alpha \quad \text{или} \quad x = l \cos^3 \alpha.$$

Подставляя значение x в уравнение (10), получим $y = l \sin^3 \alpha$.

Уравнения $x = l \cos^3 \alpha$, $y = l \sin^3 \alpha$ являются параметрическими уравнениями огибающей, которая называется *астроидой*.

Для получения общего уравнения огибающей найдем

$$\cos^2 \alpha = \left(\frac{x}{l}\right)^{\frac{2}{3}}, \sin^2 \alpha = \left(\frac{y}{l}\right)^{\frac{2}{3}}, \cos^2 \alpha + \sin^2 \alpha = \left(\frac{x}{l}\right)^{\frac{2}{3}} + \left(\frac{y}{l}\right)^{\frac{2}{3}} = 1$$

или

$$x^{\frac{2}{3}} + y^{\frac{2}{3}} = l^{\frac{2}{3}}.$$

На рис. 11 построена астроида, если параметр $\alpha \in [0; 2\pi]$.

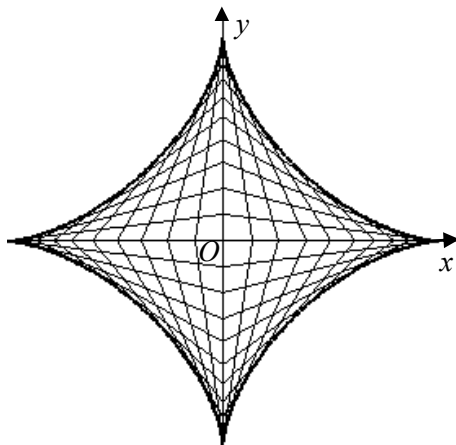


Рис. 11

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): xc = 4000: yc = 5000: l = 2000
Line (xc - 4000, yc)-(xc + 4000, yc)
Line (xc, yc - 4000)-(xc, yc + 4000)
For al = 0 To 2 * pi Step pi / 24
Line (l * Cos(al) + xc, yc)-(xc, l * Sin(al) + yc)
Next al
DrawWidth = 2
For al = 0 To 2 * pi Step 0.01
PSet (l * (Cos(al)) ^ 3 + xc, l * (Sin(al)) ^ 3 + yc)
Next al
End Sub
```

Замечание. Иногда однопараметрическое семейство линий аналитически задается с помощью двухпараметрического уравнения $f(x, y, \alpha, \beta) = 0$, причем параметры α, β удовлетворяют

уравнению $g(\alpha, \beta) = 0$. Познакомьтесь с методом поиска огибающей в этом случае в пособии [3].

1.20. Составьте параметрические уравнения эквидистант и определите тип этих линий для следующих линий:

а) для окружности $x = R \cos t, y = R \sin t, t \in [0; 2\pi]$;

б) для прямой $x = mt + x_0, y = nt + y_0, t \in R$;

в) для параболы $y = ax^2 + bx + c$.

1.21. Постройте эквидистанты для прямоугольного треугольника с катетами a, b .

1.22. Для параболы $y = ax^2$ напишите параметрические уравнения. Найдите параметрические уравнения эквидистантных линий (рис. 12) и напишите программу построения семейства этих линий.

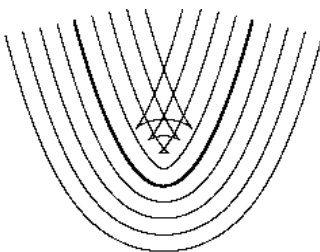


Рис. 12

1.23. Для кубической параболы $y = ax^3$ определите параметр a , чтобы парабола наглядно изображалась на экране компьютера (рис. 13). Найдите параметрические уравнения кубической параболы. Постройте семейство окружностей с центрами на кубической параболе и фиксированного радиуса h . Найдите параметрические уравнения эквидистантных линий с высотой h и постройте две эквидистантные линии с этой высотой.



Рис. 13

1.24. Напишите уравнение семейства окружностей данного радиуса R с центрами на прямой (рис. 14). Составьте уравнение огибающей линии. Напишите программу построения семейства линий и огибающей.

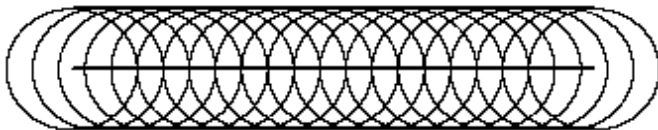


Рис. 14

1.25. Найдите уравнение семейства окружностей данного радиуса R_1 с центрами на окружности радиуса R , где $R > R_1$ (рис. 15). Составьте уравнение огибающей линии. Напишите программу построения семейства линий и огибающей.

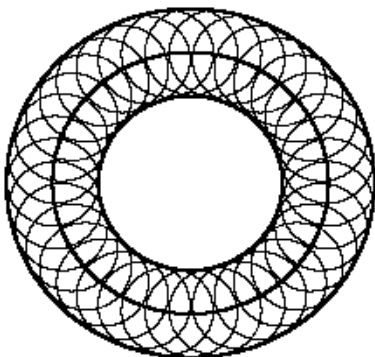


Рис. 15

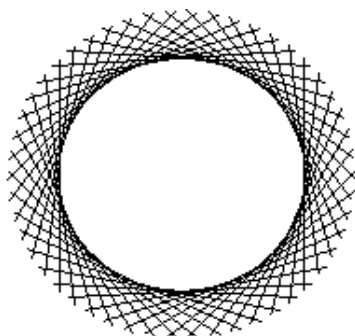


Рис. 16

1.26. На рис. 16 задано однопараметрическое семейство прямых $\cos \alpha x + \sin \alpha y - R = 0$, где $R = \text{const}$, $\alpha \in [0; 2\pi]$. Найдите огибающую этого семейства линий. Проверьте, что точки $A(R \cos \alpha - R \sin \alpha; R \sin \alpha + R \cos \alpha)$, $B(R \cos \alpha + R \sin \alpha; R \sin \alpha - R \cos \alpha)$ принадлежат прямой $\cos \alpha x + \sin \alpha y - R = 0$. Постройте однопараметрическое семейство отрезков с концами в этих точках.

1.27. Дан прямой угол (рис. 17). Проводятся различные прямые, которые отсекают на сторонах угла треугольники, имеющие фиксированную площадь S . Составьте уравнение однопараметрического семейства прямых и найдите огибающую линию этого семейства. Напишите программу построения семейства линий.

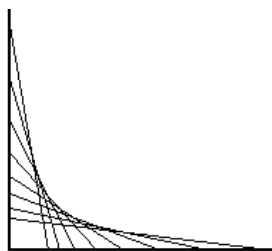


Рис. 17

§ 5. Построение аналогичных фигур

Построенная фигура на экране компьютера в большинстве случаев зависит от нескольких параметров. Иногда требуется расположить на экране несколько аналогичных фигур и показать зависимость от параметров. В одних случаях это легко программируется с помощью цикла. В других случаях зависимость между параметрами сложно отразить в программе и проще перечислить несколько наборов значений параметров. Чтобы не повторять программу для нескольких случаев, удобно использовать обращение к процедуре построения этой фигуры.

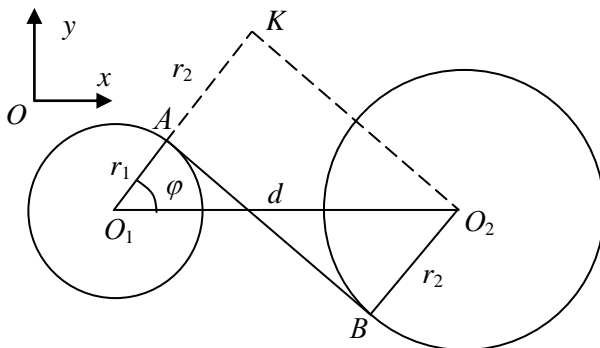


Рис. 1

Задача. Требуется построить несколько аналогичных фигур, каждая из которых состоит из двух окружностей; отрезка, соединяющего центры окружностей, и одной общей внутренней касательной. Для упрощения вычислений можно считать, что линия центров окружностей является горизонтальной прямой.

Разработаем математическую модель построения фигуры.

Пусть фигура определяется геометрическими параметрами:

r_1, r_2 — радиусы окружностей,

d — расстояние между центрами окружностей.

Для изображения нескольких фигур нужно начинать строить аналогичную фигуру с новой точки. Выберем в качестве вспомогательных параметров координаты x_1, y_1 центра O_1 первой окружности (расположенной слева), тогда центр O_2 второй окружности имеет координаты $x_1 + d, y_1$.

Пусть AB — внутренняя касательная окружностей, K — основание перпендикуляра, опущенного из точки O_2 на прямую O_1A , тогда прямые AB и O_2K параллельны и, следовательно, $AK = r_2, O_1K = r_1 + r_2, O_2K = \sqrt{d^2 - (r_1 + r_2)^2}$.

Пусть $\angle O_2O_1A = \varphi$, тогда $\overline{O_1A}(r_1 \cos \varphi; r_1 \sin \varphi)$,
 $\overline{O_2B}(r_2 \cos(\varphi + \pi); r_2 \sin(\varphi + \pi), \overline{O_2B}(-r_2 \cos \varphi; -r_2 \sin \varphi)$,
 $\cos \varphi = \frac{O_1K}{O_1O_2} = \frac{r_1 + r_2}{d}$, $\overline{O_1A}\left(r_1 \frac{r_1 + r_2}{d}; r_1 \frac{\sqrt{d^2 - (r_1 + r_2)^2}}{d}\right)$.

Найдем координаты точек касания:

$$\overline{OA} = \overline{OO_1} + \overline{O_1A}, A\left(x_1 + r_1 \frac{r_1 + r_2}{d}; y_1 + r_1 \frac{\sqrt{d^2 - (r_1 + r_2)^2}}{d}\right).$$

$$B\left(x_2 - r_2 \frac{r_1 + r_2}{d}; y_2 - r_2 \frac{\sqrt{d^2 - (r_1 + r_2)^2}}{d}\right), \text{ где } x_2 = x_1 + d, y_2 = y_1.$$

Начинаем компьютерное моделирование рисунков.

Вначале объявляем процедуру построения фигуры, присвоив ей определенное имя и указав в ней входные параметры. Например, **Окружности(x1, y1, r1, r2, d)**, где x_1, y_1 — координаты центра первой окружности, r_1 — радиус первой окружности, r_2 — радиус второй окружности, d — расстояние между центрами окружностей.

```
Private Sub Окружности(x1, y1, r1, r2, d)
Circle (x1, y1), r1: Circle (x1 + d, y1), r2
Line (x1, y1)-(x1 + d, y1)
xa = x1 + r1 * (r1 + r2) / d: xb = x1 + d - r2 * (r1 + r2) / d
ya = y1 + r1 * Sqr(d ^ 2 - (r1 + r2) ^ 2) / d
yb = y1 - r2 * Sqr(d ^ 2 - (r1 + r2) ^ 2) / d
Line (xa, ya)-(xb, yb): End Sub
```

Определяем операторы построения каждой окружности, отрезка, соединяющего центры окружностей; вычисляем координаты точек касания и строим касательную.

Следующий блок в программе является процедурой-событием **Command1_Click** вызова командной кнопки для запуска программы и предписывает компьютеру обратиться три раза

к написанной выше программе для заданных наборов входных параметров (рис. 2).

```
Private Sub Command1_Click()
Окружности 1000, 1000, 300, 600, 1200
Окружности 4000, 1000, 500, 500, 1200
Окружности 1500, 3000, 1200, 800, 4000
End Sub
```

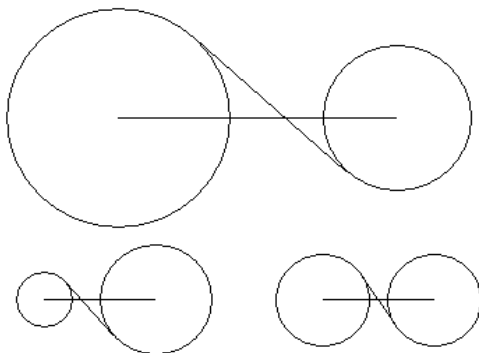


Рис. 2

Протестируйте программу «Касательная к окружностям» для других значений наборов параметров.

Дополните программу так, чтобы она строила вторую внутреннюю касательную к двум окружностям.

Измените программу так, чтобы она строила две внешние касательные к двум окружностям.

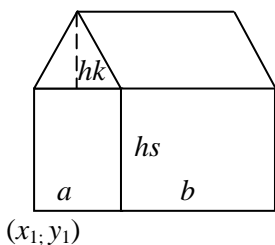


Рис. 3

1.28. Разработайте математическое моделирование плоской модели домика (рис. 3) с параметрами:

x_1, y_1 — координаты одного из углов (закладки первого кирпича);

a — ширина домика;

b — длина домика;

hs — высота стены;

hk — высота крыши.

1.29. Напишите программу построения нескольких домиков (рис. 4), используя обращение к процедуре рисования домика, и сравните ее с программой «Домики SM».

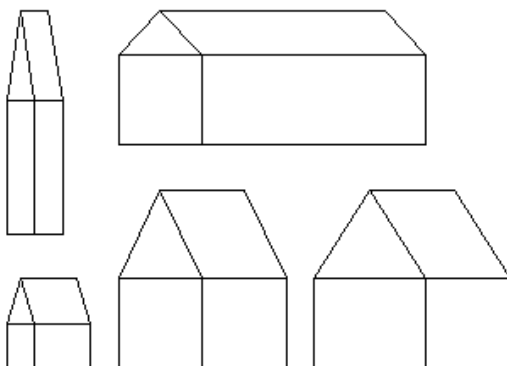


Рис. 4

§ 6. Приемы изображения движущегося объекта

а) Прием поэлементного рисования и стирания движущегося объекта.

Разверткой окружности называется траектория конца туго натянутой нити, сматываемой с неподвижной круглой плоской катушки (рис. 1).

Пусть радиус катушки равен R , начало системы координат находится в центре катушки, N — произвольная точка развертки окружности, туго натянутая нить MN касается катушки в точке M окружности. Пусть точке M на окружности соответствует параметр t , т.е. $\angle AOM = t$, тогда: $\overline{OM}(R \cos t; R \sin t)$, $x_M = R \cos t$, $y_M = R \sin t$. Длина отрезка MN равна длине развернутой дуги, т.е. $MN = \cup MA = Rt$.

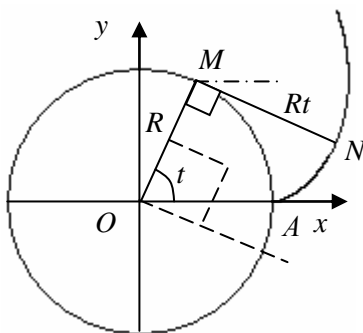


Рис. 1

Вектор \overline{MN} образует с осью Ox угол $(t - 90^\circ)$,
 $\overline{MN} \left(Rt \cos \left(t - \frac{\pi}{2} \right); Rt \sin \left(t - \frac{\pi}{2} \right) \right)$,
 $\overline{MN} (Rt \sin t; -Rt \cos t)$,
 $\overline{ON} = \overline{OM} + \overline{MN}$,
 $x_N = R \cos t + Rt \sin t$,
 $y_N = R \sin t - Rt \cos t$.

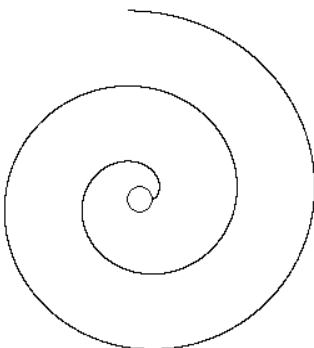


Рис. 2

Математическое моделирование траектории выполнено.

Программа построения развертки окружности — на рис. 2.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): x0 = 5000: y0 = 3500: r = 200
Circle (x0, y0), r
For t = 0 To 4 * pi Step 0.001
xn = r * Cos(t) + r * t * Sin(t): yn = r * Sin(t) - r * t * Cos(t)
PSet (xn + x0, yn + y0)
For k = 1 To 100000
Next k, t
End Sub
```

Начинаем компьютерное моделирование линии (рис. 3) с постепенным усложнением проекта.

Представляет интерес смоделировать процесс разматывания нити с катушки, т.е. для каждого значения параметра t нужно изобразить движущийся отрезок MN . Если изобразить все отрезки, то они накроют саму развертку окружности.

Построим несколько отрезков (рис. 3), дополнив вышеприведенную программу циклом с большим шагом:

```
For t = 0 To 5 * pi + 0.01 Step pi / 8
xm = r * Cos(t): ym = r * Sin(t)
xn = r * Cos(t) + r * t * Sin(t): yn = r * Sin(t) - r * t * Cos(t)
Line (xm + x0, ym + y0)-(xn + x0, yn + y0)
Next t
```

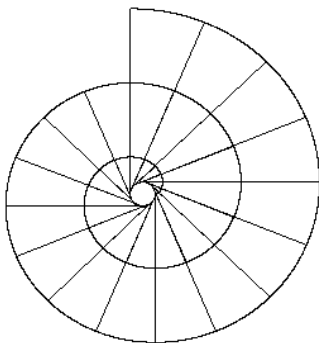


Рис. 3

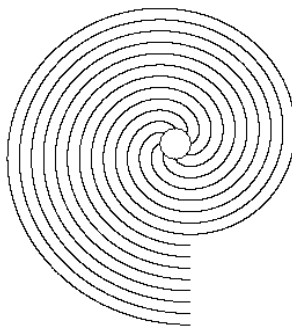


Рис. 4

1.30. Постройте семейство разверток одной и той же окружности, но повернутых вокруг начала координат (рис. 4).

1.31. Дополните рис. 4 другим семейством разверток окружностей, но «закрученных» в обратном направлении (рис. 5).

Интересно рассмотреть процесс непрерывного сматывания нити. Термин «непрерывное сматывание» в этом месте будем понимать как обычно в информатике, т.е. построение с довольно малым естественным шагом изменения угла, например, 0,01.

К сожалению, нарисованные отрезки MN нужно стирать с экрана компьютера при переходе к рисованию нового отрезка, иначе значительная часть экрана будет

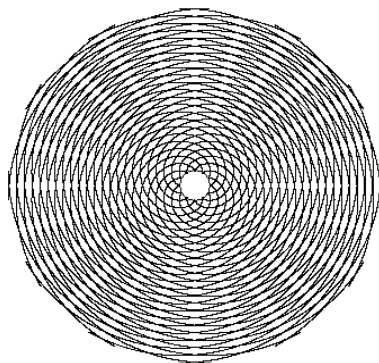


Рис. 5

покрыта отрезками и точки развертки невозможно найти на экране. Стирание отрезка приводит и к стиранию точек самой линии, а также к стиранию данной окружности. Особенно наглядно это видно, если развертка окружности содержит более одного оборота вокруг начала координат. Поэтому в программу нужно вставить цикл по восстановлению витков, нарисованных на предыдущих этапах. Если повторить рисование всех точек развертки при каждом значении параметра t , то это приведет к значительному замедлению работы программы. Можно восстановить только

несколько точек, стертых отрезком на развертке. Для этого нужно определить количество точек пересечения отрезка с разверткой и дорисовать их снова.

Программа построения процесса развертки окружности с изображением натянутой нити имеет вид:

```
Private Sub Command1_Click()  
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)  
pi = 4 * Atn(1): x0 = 5000: y0 = 3500: r = 200: Circle (x0, y0), r  
For t = 0 To 6 * pi Step 0.01  
xm = r * Cos(t): ym = r * Sin(t)  
xn = r * Cos(t) + r * t * Sin(t): yn = r * Sin(t) - r * t * Cos(t)  
Circle (xm + x0, ym + y0), 20, QBColor(0): Line (xm + x0, ym + y0)-(xn + x0, yn + y0)  
For s = 0 To 200000  
Next s  
Circle (xm + x0, ym + y0), 20, QBColor(15)  
Line (xm + x0, ym + y0)-(xn + x0, yn + y0), QBColor(15)  
Circle (x0, y0), r  
i = Int(t / (2 * pi))  
For q = 0 To i  
xn = r * Cos(t) + r * (t - q * 2 * pi) * Sin(t): yn = r * Sin(t) - r * (t - q * 2 * pi) * Cos(t)  
DrawWidth = 2  
PSet (xn + x0, yn + y0)  
DrawWidth = 1  
Next q, t: End Sub
```

Эта программа лишь частично решает эту проблему. А может читатель сможет решить эту проблему лучше? На компакт-диске в программе **«Развертка окружности»** предложено еще одно решение этой проблемы.

Выделим этапы изображения движущегося объекта:

1. Разработать математическое моделирование объекта в зависимости от параметра, например, от t .
2. Определить границы изменения параметра t для лучшего изображения динамики объекта на экранной форме.
3. Сформировать цикл для переменной t и заданными границами изменения параметра. Изобразить этот объект некоторым цветом, отличным от цвета экранной формы.
4. Внутри цикла поставить новый цикл задержки.
5. После цикла задержки, но внутри внешнего цикла по переменной t стереть этот объект, т.е. изобразить повторно объект цветом экранной формы.
6. Если предыдущая операция привела к стиранию некоторых элементов других объектов, то нарисовать их снова.

Замечание. После завершения работы программы все изображение исчезнет, т.к. при последнем значении переменной на пятом этапе изображение будет стерто с экрана.

Как следует изменить этот алгоритм, чтобы на экране оставалось изображение объекта после завершения работы программы?

1.32. На рис. 6 изображено несколько радиусов окружности.

Разработайте моделирование и напишите программу изображения на экране компьютера процесса вращения радиусов окружности.

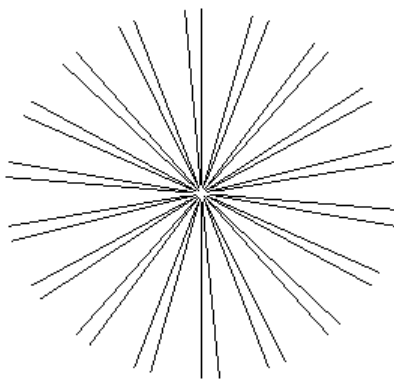


Рис. 6

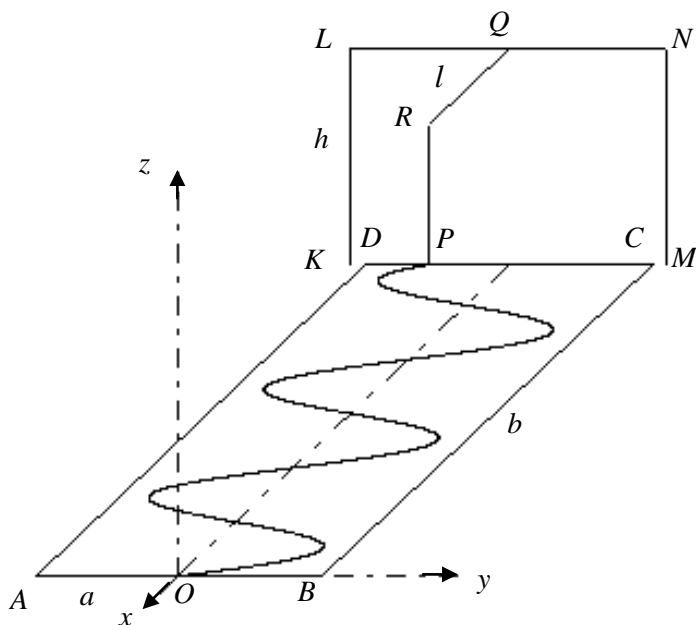


Рис. 7

В некоторых методических пособиях по математике иллюстрацию образования синусоиды предлагается представить с помощью следующей конструкции (рис. 7).

На перекладине LN в точке Q подвешено ведро R с песком, которое совершает гармонические колебания с постоянной угловой скоростью в вертикальной плоскости $KLNM$. В горизонтальной плоскости под перекладиной расположен лист бумаги $ABCD$, $AB = 2a$, $AD = b$, который движется с постоянной скоростью в направлении, перпендикулярном плоскости $KLNM$. Из ведра с небольшим отверстием на движущийся лист бумаги высыпается песок. Предположим, что песчинки после падения не отскакивают от плоскости листа, а также предположим, что сила тяжести, направленная вниз, во много раз превосходит силу отклонения частицы под воздействием гармонических колебаний, т.е. песчинка после прохождения отверстия в ведре падает вертикально вниз, тогда песчинки на бумаге будут расположены на синусоиде.

Изобразить этот процесс на компьютере сложно, т.к. нужно изобразить движущийся лист с сохранением точек на синусоиде. Для компьютерного изображения проще считать лист бумаги неподвижным, а движущейся — конструкцию перекладины $KLNM$ вместе с ведром.

Рассмотрим систему координат, представленную на рис. 7, и произвольное положение точки $R(x; y; z)$ в момент времени t . Будем считать, что в момент времени $t = 0$ вся конструкция находилась в плоскости Oyz .

Точка R участвует в двух движениях. С одной стороны, она движется вдоль оси Ox в отрицательном направлении этой оси, поэтому $x = -vt$, где v — линейная скорость. С другой стороны, она совершает движения по окружности с центром в точке Q и радиусом $QR = l$, поэтому $y = l \sin \varphi$, $z = h - l \cos \varphi$, где угол отклонения φ должен изменяться пропорционально времени. Последняя фраза требует уточнения, т.к. ведро не может совершать круговые движения. Оно должно подняться на определенный угол α в одну сторону, затем на такой же угол в другом направлении.

1.33. Разработайте программу этой модели и сравните ее с программой «Песок на синусоиду».

Б) Прием стирания участка экранной формы.

Если объект состоит из множества точек, которые были построены в цикле, и этот процесс занимал определенное время, то его стирание может занять также определенное количество времени. Чтобы процесс стирания был незаметен на экране, необходимо предусмотреть цикл задержки по времени, превосходящий по длительности процесс стирания объекта. Работа программы становится медленной.

В этом случае иногда может помочь другой прием стирания участка экранной формы, т.е. определенная часть экранной формы, содержащая нарисованный объект, должна переместиться в новое положение на экране. Другими словами, эту часть экрана нужно скопировать в память, как целый объект, прежнее расположение вырезать и вставить в новое положение на экранной форме.

Этот прием имеет свой недостаток, т.к. выделенная часть экрана с движущимся объектом не должна содержать элементов других неподвижных объектов.

Запустите программу «**Копирование участка формы**» и проследите действие операторов программы.

```
Private Sub Command1_Click()  
Circle (250, 250), 200, QBColor(10) : Line (50, 50)-(450, 450), QBColor(9)  
Line (50, 450)-(450, 50), QBColor(9)  
'рисунок на форме в квадрате(50, 50)-(450, 450)  
For k = 0 To 1000000  
Next k  
Dim M(0 To 400, 0 To 400)  
'заполнение массива точками рисунка  
For x = 0 To 400  
For y = 0 To 400  
M(x, y) = Point(x + 50, y + 50)  
Next y, x  
Line (50, 50)-(450, 450), QBColor(15), BF  
'стирание первоначального рисунка на форме  
For x = 0 To 400  
For y = 0 To 400  
PSet (500 + x, 500 + y), M(x, y)  
'восстановление рисунка из массива в новое место  
Next y, x: End Sub
```

1.34. Напишите программу, которая показывает движение окружности методом стирания прежнего положения участка формы и восстановления рисунка в новом месте.

В) Прием дискретного изображения промежуточных положений.

Если изобразить все положения движущегося объекта, то изображенные точки объекта на предыдущих этапах мешают зрителю воспринимать положение объекта в настоящее время. В данном случае эти положения нужно разделить так, чтобы их изображения не пересекались для различных временных значений. Следовательно, нужно дискретно изменять параметр с некоторым шагом.

Рассмотрим движение рисунка на экранной форме.

Пусть в некотором файле находится рисунок. Скопируем этот рисунок. Начинаем создавать проект обычным способом. Нажимая клавишей мышки на Microsoft Visual Basic 6.0, открываем **Новый проект**, выбираем **Standard EXE**, затем выбираем **Picture Box**, дважды щелкая по этой иконке, получаем на форме место, куда вставим рисунок. Потянув за угол этого рисунка, можно изменить его размеры.

Известным ранее методом размещаем на форме командную кнопку **Command Button**. Нажимая на эту кнопку, переходим в окно для написания программы.

Запустите программу «**Движение картинки**» и проследите действие операторов программы.

```
Private Sub Command1_Click()  
For i = 1 To 4  
Picture1.Left = Picture1.Left + 1500  
For K = 0 To 10000000  
Next K, i  
End Sub
```

В предложенной программе на экранной форме вставлен второй рисунок, к которому программа не обращается. Вставьте в эту программу только один оператор, чтобы двигался и второй рисунок.

Добавьте на форму третий рисунок и измените программу так, чтобы три рисунка двигались с различными шагами переноса.

§ 7. Геометрия шарнирных механизмов

1. Постановка задачи.

Для передачи энергии от источника энергии к потребителю в технике и в быту применяют механизмы. Среди разнообразных механизмов широкий класс образуют шарнирные механизмы, в которых некоторые стержни могут вращаться вокруг осей. В двигателях автомобилей, самолетов и кораблей энергия, полученная при сгорании топлива, передается на вращающийся вал и затем — на средства, приводящие этот объект в движение. В этом случае прямолинейное движение поршня преобразуется во вращательное движение коленчатого вала. В швейной машинке с электрическим приводом вращательное движение ротора электромотора преобразуется через систему механизмов в прямолинейное движение иглы.

Механизмы, в которых при движении одной точки по окружности или по дуге окружности некоторая другая точка механизма движется по прямой линии, называются *направляющими механизмами*. Задача об отыскании направляющего механизма была решена Уаттом (Ваттом), который, совершенствуя паровую машину, встретился с необходимостью передачи прямолинейного движения головки поршневого штока к концу коромысла, движущегося по дуге окружности. В механике механизм называют *механизмом Уатта*, а в математике линию, начертанную одной из точек этого механизма, близкую к отрезку, называют *кривой Уатта*.

Преобразование движения одной точки механизма по окружности в прямолинейное движение некоторой другой точки этого механизма и наоборот является важной конструкторской задачей. Наряду с математической формулировкой задачи при решении практических задач приходится учитывать много факторов из механики: возникающие силы и моменты, зависящие от скорости и ускорения при движении точки по окружности.

В большинстве случаев для решения конкретной практической задачи ограничиваются приближенным решением этой задачи, т.е. преобразованием точек окружности в точки линии, достаточно близко воспроизводящий прямолинейный отрезок.

Направляющие механизмы подразделяются на точные, в которых движение теоретически должно происходить точно по прямой линии и по дуге окружности, и на приближенные.

Большой вклад в конструирование шарнирных механизмов, превращающих вращательное движение в прямолинейное, внес русский ученый Пафнутий Львович Чебышев (1821—1894), известный математик и механик.

Шарнирные направляющие механизмы находят широкое применение в двигателях, системах подвески ходовой части автомобилей, в робототехнике.

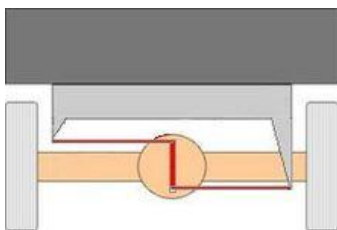


Рис. 1

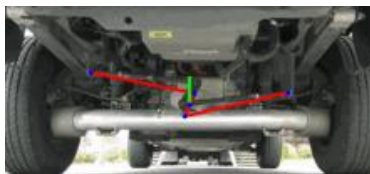


Рис. 2

Механизм Уатта предназначен для предотвращения смещения моста автомобиля вбок относительно кузова автомобиля, позволяя мосту в то же время двигаться в вертикальной плоскости. Механизм состоит из двух горизонтальных звеньев одинаковой длины, прикрепленных с каждой из сторон шасси автомобиля (рис. 1, 2). Звенья соединены коротким звеном, центр которого присоединен к центру оси. Центр вертикального звена вынужден двигаться вертикально, и не позволяет смещаться по сторонам мосту автомобиля при качке на неровностях дороги.

В настоящее время большой интерес представляет компьютерное моделирование движения направляющего механизма.

Вначале изучим моделирование точного направляющего механизма — инверсора.

2. Конструкция инверсора.

В 1864 г. французский ученый Посселье изобрел инверсор, преобразующий движение точки по окружности в движение другой точки механизма по отрезку, но это открытие осталось незамеченным для техники. Независимо от Посселье, петербургский

студент Липкин под влиянием П.Л.Чебышева изобрел инверсор спустя семь лет.

Инверсор Липкина-Посселье состоит из шести стержней, соединенных шарнирно. Четыре стержня образуют ромб $MPNQ$ (рис. 3). Остальные два стержня OP и OQ равны между собой, причем каждый из них длиннее стороны ромба.

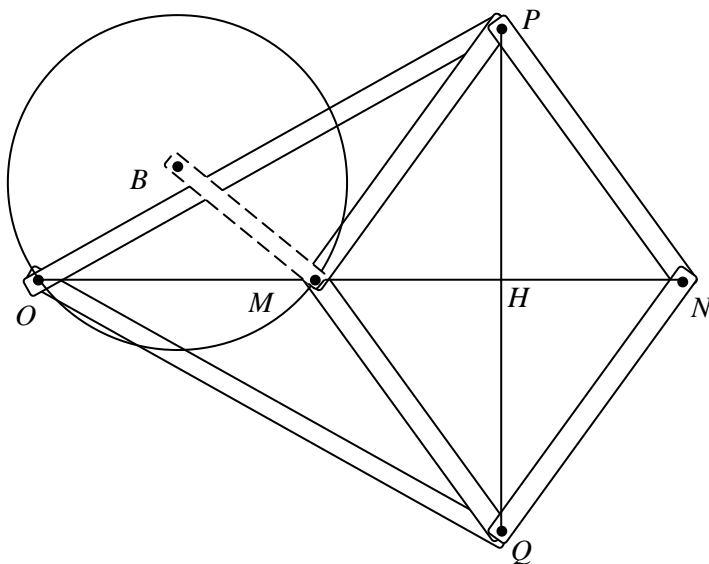


Рис. 3

Обозначим $OP = l$, $MP = a$, $MN \cap PQ = H$, $l^2 - a^2 = R^2$, тогда точки O , M и N расположены на одной прямой в силу симметричности механизма относительно прямой ON .

$$\begin{aligned} OM \cdot ON &= (OH - MH)(OH + HM) = OH^2 - MH^2 = \\ &= (OP^2 - PH^2) - (MP^2 - PH^2) = l^2 - a^2 = R^2. \end{aligned}$$

Таким образом, точки M и N являются инверсными относительно окружности с центром O и радиусом R .

Если точка M будет двигаться по окружности, проходящей через центр O , то ее образ должен двигаться по прямой. Для задания движения точки M механизма по окружности и движения точки N механизма по прямой на механизм накладывают дополнительное условие.

В точке O механизм закрепляют так, чтобы стержни OP и OQ могли вращаться вокруг точки O . К инверсору присоединяют шарнирно в точке M стержень, второй конец которого закрепляют в некоторой точке B , чтобы $OB = BM$.

3. Постановка проблемы проекта.

В Википедии (свободно распространяемой энциклопедии) представлена модель инверсора в движении. К сожалению, в этой модели не приводятся параметры механизма, поэтому модель нельзя изменить. Видимое изображение работы механизма создает впечатление, что в ней реализован частный случай механизма Липкина-Посселье. Модель имеет ограниченные углы поворота механизма, поэтому на ней трудно реализовать прохождение отрезка заданной длины. Параллелограмм механизма не полностью складывается, т.к. угол для предельного положения не исследован. На одном чертеже невозможно изобразить несколько положений механизма.

Математического и компьютерного моделирования этой модели в Интернете нет, и не указаны никакие ссылки на моделирование.

Целью предлагаемого проекта является:

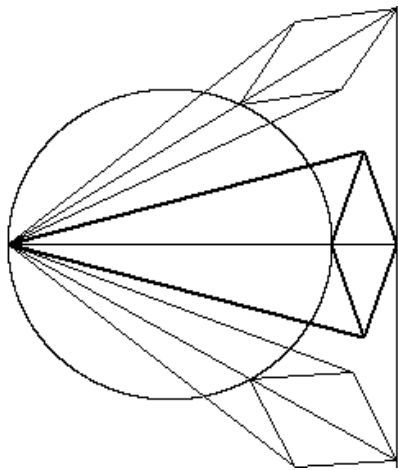


Рис. 4

- увеличение числа параметров механизма (изменяя параметры в компьютерной программе, можно видоизменять как сам механизм, так и диапазон его движения);

- математическое моделирование работы механизма Липкина-Посселье (рис. 4);

- разработка различных алгоритмов прохождения дуги окружности в данном и противоположном направлениях;

- написание компьютерной программы движения модели механизма на экране компьютера.

В Википедии представлена также модель движения механизма Ватта, в которой не меняются параметры.

Кривая Уатта определяется общим уравнением шестой степени [15]. Далее в работе будет представлен вывод этого уравнения. Для моделирования линии нужно одну из переменных выразить через другую, что в явном виде невозможно. Если использовать подпрограмму для решения этого уравнения, то встает проблема об определении диапазона изменения одной из переменных. Этот метод моделирования не является перспективным.

Кривую Уатта можно задавать уравнением в полярных координатах. Кривая имеет две оси симметрии и центр симметрии в точке O (рис. 5). Вся кривая находится в паре вертикальных углов. Центр симметрии кривой является особой точкой. Если точки приближается по кривой Уатта к особой точке, то касательная к кривой стремится к предельному положению, т.е. к одной из прямых l_1 или l_2 . Параметризация движения механизма с помощью полярного угла в этом случае является неравномерной. Для наглядного изображения движения механизма в этом случае нужно изобретать особый способ изменения параметра, т.е. полярного угла.

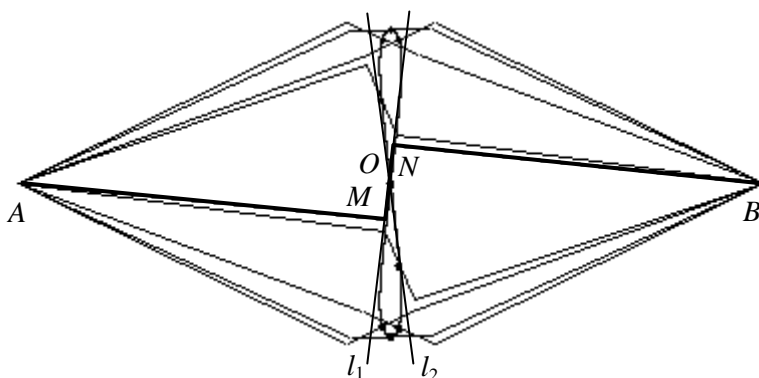


Рис. 5

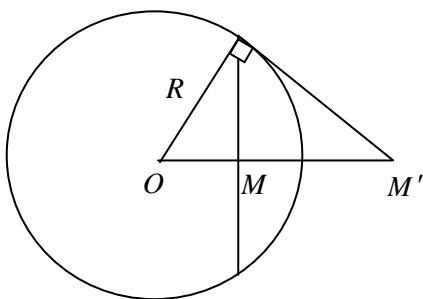
Во второй части проекта представлено новое моделирование движения механизма Ватта, в основе которого лежит принцип равномерно вращения ведущего звена механизма.

4. Математическая основа инверсора.

Приведем известные сведения о математическом преобразовании плоскости, т.е. инверсии, на которой основано действие механизма.

Подчеркнем, что видимое движение точки на экране компьютера по окружности или по отрезку еще не означает, что эти линии являются таковыми.

Инверсией $I_{M_0}^R$ с центром в данной точке M_0 и радиусом R называется отображение плоскости в себя, при котором произвольная точка M , отличная от точки M_0 , отображается



в точку M' на луче M_0M , такую, что $M_0M \cdot M_0M' = R^2$. На рис. 6 представлен один из алгоритмов построения инверсной точки.

Если центр инверсии находится в начале декартовой системы координат и точка

Рис. 6

$M(x; y)$ отображается в точку $M'(x'; y')$, то

$$x' = \frac{R^2 x}{x^2 + y^2}, \quad y' = \frac{R^2 y}{x^2 + y^2}.$$

При инверсии (рис. 7) относительно окружности ω :

— прямая a , проходящая через центр инверсии, отображается в себя;

— прямая b , не проходящая через центр инверсии, отображается в окружность b' , проходящую через центр инверсии;

— окружность b' , проходящая через центр инверсии, отображается в прямую b , не проходящую через центр инверсии, и наоборот;

— окружность γ , не проходящая через центр инверсии, отображается в окружность γ' , не проходящую через центр инверсии.

При инверсии $I_{M_0}^R$ для расстояний AB и $A'B'$, где $A' = I_{M_0}^R(A)$, $B' = I_{M_0}^R(B)$, выполняется равенство $A'B' = \frac{AB \cdot R^2}{M_0A \cdot M_0B}$.

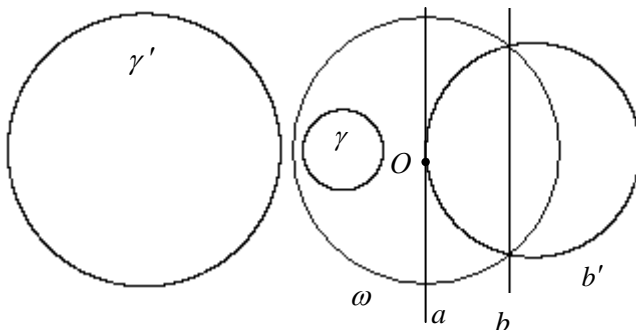


Рис. 7

5. Математическое моделирование движения механизма Липкина-Посселье.

Пусть окружность γ с центром B , проходящая через точку O , отображается в прямую γ' (рис. 8). Обозначим основание перпендикуляра, опущенного из точки O на прямую γ' , через N_0 . Пусть в начальный момент механизм находится в положении $OM_0Q_0N_0P_0$. Введем параметры механизма в начальный момент:

$2m$ — длина диагонали M_0N_0 ;

$2h$ — длина диагонали P_0Q_0 ;

$2b$ — длина диаметра OM_0 .

Тогда $a = M_0P_0 = \sqrt{m^2 + h^2}$, $l = OP_0 = \sqrt{(2b + m)^2 + h^2}$.

Замечание. Для механизма Липкина-Посселье с конструктивной точки зрения являются наиболее естественными параметры: l , a , OM_0 . Но при тестировании компьютерной программы удобнее визуально наблюдать и изменять параметры $2b, 2m, 2h$.

Пусть механизм перешел в положение $OMQNP$ и $\angle M_0OM = t$, тогда $\angle M_0BM = 2t$, $OM = 2b \cos t$.


$$MP^2 = OM^2 + OP^2 - 2OM \cdot OP \cdot \cos v$$

ИЛИ

$$\cos v = \frac{l^2 + 4b^2 \cos^2 t - a^2}{4lb \cos t}.$$

Для каждого значения t можно определить значение угла v , поэтому в программе можно использовать значения углов

$$\angle MOP = t + v, \angle MOQ = t - v.$$

Для точек механизма получаем координаты

$$M(b + b \cos 2t; b \sin 2t), P(l \cos(t + v); l \sin(t + v)),$$

$$Q(l \cos(t-v); l \sin(t-v), N(2(b+m); 2(b+m) \operatorname{tgt})).$$

Определим диапазон изменения угла t для механизма.

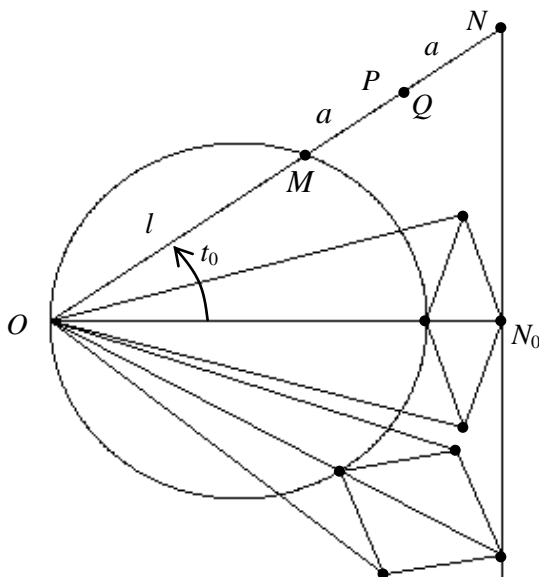


Рис. 9

Механизм Липкина-Посселье достигает предельного положения, когда ромб вырождается в дважды проходимый отрезок (рис. 9). По определению инверсии получаем

$$OM \cdot ON = R^2 \text{ или } OM \cdot (OM + 2a) = l^2 - a^2.$$

Решая квадратное уравнение относительно OM , получаем

$$OM = l - a, \quad 2b \cos t_0 = l - a,$$

$$\cos t_0 = \frac{l - a}{2b} \text{ или}$$

$$\operatorname{tg} t_0 = \frac{\sqrt{4b^2 - (l - a)^2}}{l - a}. \quad (1)$$

Заметим, что в треугольнике OM_0P_0 (рис. 4) всегда выполняется неравенство $l - a < 2b$, поэтому для любых значений выбранных параметров можно определить угол t_0 из формулы (1).

При моделировании механизма параметр t должен удовлетворять условию $t \in [-t_0; t_0]$.

6. Разработка алгоритма для компьютерной программы.

Для создания видимости движения механизма на экране компьютера необходимо изобразить много различных положений этого механизма. Если изобразить все положения механизма, то на экране будет построено нагромождение отрезков, поэтому при переходе к следующему изображению предыдущее изображение нужно стереть.

Очистку экрана можно выполнить оператором

Line (0, Form1.Height)-(Form1.Width, 0), QBColor(15), BF,
который закрашивает весь экран в белый цвет.

Очистку экрана также можно выполнить оператором Cls.

Компьютер очень быстро проводит вычисления. Чтобы весь процесс движения не промелькнул с большой скоростью, в программу нужно вставить цикл задержки:

```
For s = 0 To 1000000  
Next s
```

Для создания видимости того, что одна из точек механизма движется по окружности, а ее образ — по прямой, нужно построить эти линии.

Если окружность построить одним оператором, а точка механизма потом будет двигаться по окружности, то эти траектории могут различаться в зависимости от параметров экрана компьютера. Поэтому окружность построим по точкам с использованием цикла:

```
For u = 0 To pi Step 0.008  
PSet (b + b * Cos(2 * u) + x0, b * Sin(2 * u) + y0), QBColor(12)  
Next u
```

Построение прямой, т.е. образа окружности:

```
Line (2 * b + 2 * m + x0, 100)-(2 * b + 2 * m + x0, 8000), QBColor(9)
```

Движение точки M по дуге окружности должно осуществляться в прямом и в противоположном направлениях. Для наглядного изображения того, что в предельных положениях механизма при значениях угла $t = \pm t_0$ изменяются скорости и ускорения частей механизма, нужно полный цикл прохода повторить несколько раз. Такое движение можно осуществить следующим образом с помощью вложенных циклов:

```

For j = 1 To 3 ' повторение полного прохода механизма
  For t = -t0 To t0 Step pi / 400 'прохождение по дуге вперед
  ..... 'блок операторов цикла прохождения дуги
Next t
For t = -t0 To t0 Step pi / 400 'прохождение по дуге назад
  ..... 'блок операторов цикла прохождения дуги
Next t
Next j

```

Повторение цикла для прохождения дуги окружности в обратном направлении приводит к увеличению числа операторов вдвое.

Повторить полное прохождение механизма можно использованием периодической функции:

```

For w = 0 To 4 * pi Step pi / 100
  t = t0 * Sin(w)
  ..... 'блок операторов цикла прохождения дуги
Next w.

```

В этом случае число операторов сокращается почти вдвое.

При завершении демонстрации программы движения механизма последний оператор очищает экран. Для обсуждения параметров механизма желательно, чтобы начальное положение механизма осталось на экране компьютера. Поэтому в программу нужно внести уточнение и на последнем шаге не стирать изображение:

```

If w <= 4 * pi - pi / 400 Then Cls.

```

Напишите программу движения механизма и сравните с программами «Прямой и обратный ход механизма» и «Механизм Липкина-Посселе», «Механизм Липкина.pas».

7. Расширение темы проекта.

Изучим пример приближенного направляющего механизма.

Механизм Уатта (рис. 10) состоит из двух звеньев AM и BN одинаковой длины и звена MN , длина которого меньше длины звена AM . Звенья AM и BN закреплены, соответственно, в точках A и B , но могут вращаться вокруг этих точек. В точках M и N звенья могут вращаться вокруг осей, тогда точка U , являющаяся

серединой звена MN , движется по кривой Уатта, близкой к отрезку в окрестности точки U .

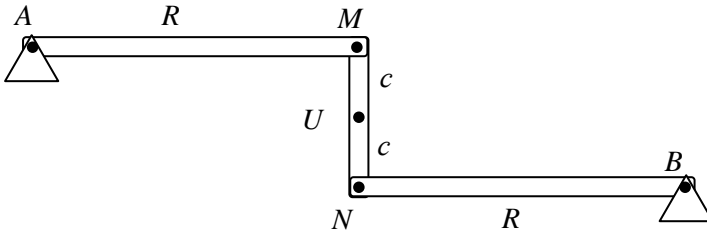


Рис. 10

Рассмотрим моделирование движения точек M , N и U , выбрав систему координат, указанную на рис. 11.

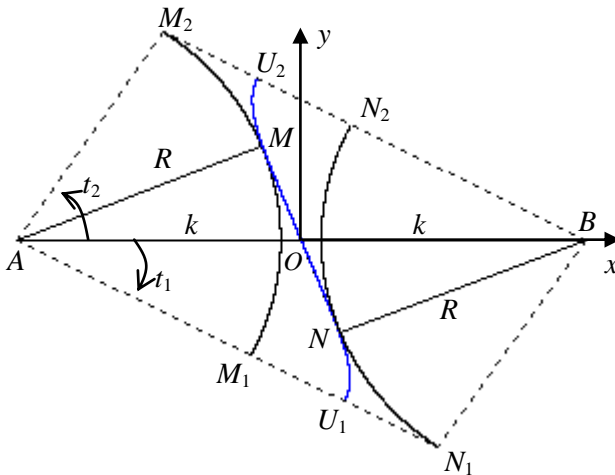


Рис. 11

Пусть O — середина отрезка AB , $R = AB = BN$, $MN = 2c$,
 $k = AO = OB = \sqrt{R^2 + c^2}$.

Движение точки M по окружности вызывает движение точки N также по окружности. Ломаная BNM может распрямиться и принять положение отрезка BM_2 , где $M_2N_2 = MN = 2c$. Трехзвенный механизм $AMNB$ примет предельное положение AM_2B . По теореме косинусов для треугольника AM_2B получаем

$$BM_2^2 = AB^2 + AM_2^2 - 2AB \cdot AM_2 \cos \angle BAM_2$$

или

$$(R + 2c)^2 = 4k^2 + R^2 - 4kR \cos t_2.$$

Вернее предельное значение угла t_2 определяется из равенства:

$$\cos t_2 = \frac{k^2 - cR - c^2}{Rk}.$$

Ломаная BNM может распрямиться и принять положение отрезка AN_1 , где $M_1N_1 = MN = 2c$. Трехзвенный механизм $AMNB$ примет предельное положение AN_1 . По теореме косинусов для треугольника AN_1 получаем:

$$BN_1^2 = AB^2 + AN_1^2 - 2AB \cdot AN_1 \cos \angle BAN_1 \text{ или}$$

$$R^2 = 4k^2 + (R + 2c)^2 - 4k(R + 2c) \cos t_1.$$

Нижнее предельное значение угла t_1 определяется из равенства:

$$\cos t_1 = \frac{k^2 + cR + c^2}{(R + 2c)k}.$$

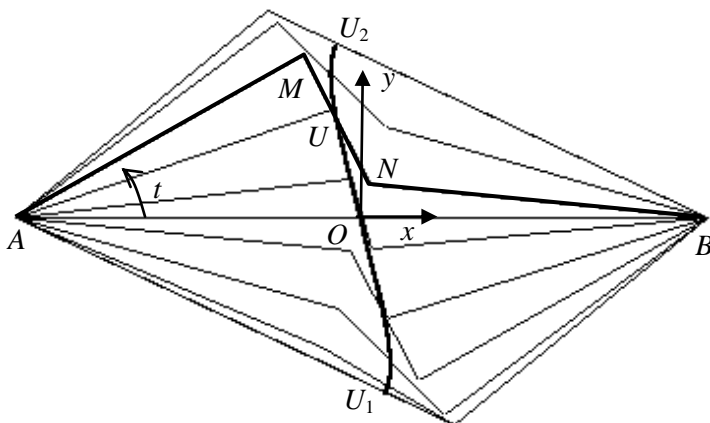


Рис. 12

Середина отрезка MN перемещается по линии U_1U_2 , которая в окрестности точки O близка к отрезку (рис. 11, 12).

Для моделирования перемещения точки M по окружности введем основной параметр $t = \angle BAM$ и вспомогательный параметр u ,

образованный лучом $[BN)$ с положительным направлением оси Ox , тогда

$$M(-k + R \cos t; R \sin t), N(k + R \cos u; R \sin u).$$

Для длины отрезка MN выполняется равенство:

$$MN^2 = [2k + R(\cos u - \cos t)]^2 + [R(\sin u - \sin t)]^2 = 4c^2$$

или

$$R \sin t \sin u - (2k - R \cos t) \cos u = 3R - 2k \cos t.$$

Обозначим $m = \sqrt{R^2 + 4k^2 - 4Rk \cos t}$ и запишем последнее равенство в виде:

$$m \left(\frac{R \sin t}{m} \sin u - \frac{2k - R \cos t}{m} \right) \cos u = 3R - 2k \cos t. \quad (2)$$

Из равенства $\left(\frac{R \sin t}{m} \right)^2 + \left(\frac{2k - R \cos t}{m} \right)^2 = 1$ следует, что можно

ввести обозначения

$$\frac{R \sin t}{m} = \sin v, \frac{2k - R \cos t}{m} = \cos v. \quad (3)$$

Для каждого значения параметра t можно определить значение v .

Равенство (2) примет вид

$$m(\sin v \sin u - \cos v \cos u) = 3R - 2k \cos t$$

или

$$\cos(u + v) = \frac{2k \cos t - 3R}{m}.$$

Из последнего равенства для каждого значения параметра t компьютер определит значение угла $w = u + v$, тогда для угла $u = w - v$ можно построить точку $N(k + R \cos u, R \sin u)$, а следовательно, можно построить положение всех звеньев механизма.

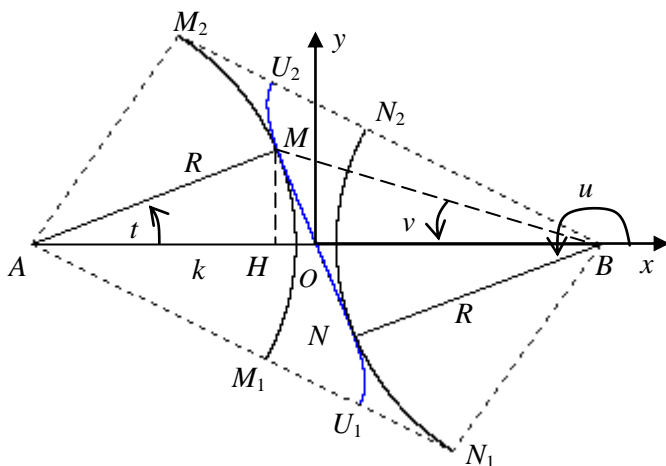


Рис. 13

Угол v , введенный по формулам (3), имеет простой геометрический смысл (рис. 13) $v = \angle MBA$, т.к. $HM = R \sin t$, $HB = 2k - R \cos t$, $BM = \sqrt{R^2 + 4k^2 - 4Rk \cos t} = m$.

В пособии [19, с. 272—282] приведен другой способ моделирования аналогичного шарнирного механизма.

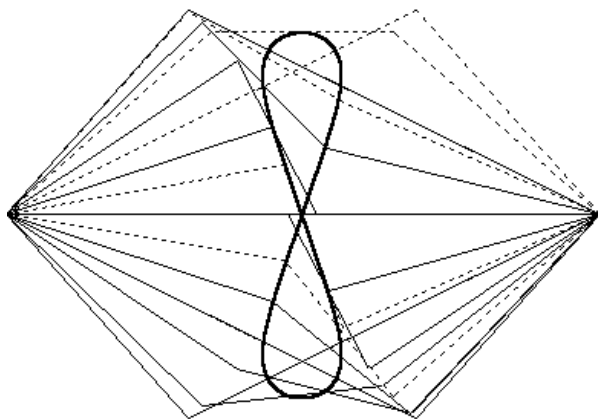


Рис. 14

Если ведущим является звено AM , то траектории трехзвенного механизма и траектория точки Уатта представлены на рис. 14

тонкими линиями, а если ведущим является звено BN , то траектории механизма представлены пунктирными линиями, а точки Уатта — жирной линией.

Напишите компьютерную программу движения механизма и сравните ее с программой «Механизм Уатта».

Приведем текст другой программы, написанной на языке PascalABC:

```

program Watt3;
uses
  Crt, GraphABC;
var
  r, c, x0, y0, j, s, n, e: integer;
  k, ct2, st2, ct1, st1, m, t, t1, t2, sv, cv, cw, sw, u, v, w: real;
begin
  SetWindowSize(1000, 2000);
  n:=2512; { переменная для цикла }
  r:=200; c:=80; k:=sqrt(r*c*r+c*c);
  x0:=300; y0:=300; { сдвиг рисунка в точку (x0,y0) }
  ct2:= (k*k-c*c*r-c*c)/r/k; { cos t2 }
  st2:= Sqrt(1-ct2*ct2); { sin t2 }
  t2:=arctan(st2/ct2); { arctg t2 }
  ct1:=(k*k+c*c*r+c*c)/k/(r+2*c); { cos t1 }
  st1:=sqrt(1-ct1*ct1); { sin t1 }
  t1:=arctan(st1/ct1); { -arctg t1 }
  for j:=0 to n do { цикл для построения движущегося механизма }
  Begin
    t:=(t1+t2)/2+(t2-t1)/2*sin(j/100); { изменение угла t от -t2 до t1 }
    m:=sqrt(4*k*k+r*r-4*k*r*cos(t));
    sv:=R*sin(t)/m; cv:=(2*k-R*cos(t))/m; v:=arctan(sv/cv); { sin v, cos v, v }
    cw:=(2*k*cos(t)-3*R)/m; sw:=Sqrt(1-cw*cw); w:=pi-Arctan(sw/cw); { sin w, cos w, w }
    u:=w-v;
    Line (-trunc(k)+x0,y0,trunc(k)+x0,y0); { AB }
    Line(-trunc(k)+x0,y0,trunc(-k+r*cos(t))+x0,Trunc(R*sin(t))+y0); { AM }
    Line(trunc(-
    k+R*cos(t))+x0,trunc(R*sin(t))+y0,trunc(k+r*cos(u))+x0,Trunc(R*sin(u))+y0);{ MN }
    Line(trunc(k)+x0,y0,trunc(k+R*cos(u))+x0,trunc(R*sin(u))+y0); { BN }
    For s:=0 To 400000 do { цикл задержки }
      s:=s+1;
    If j<n-2 Then ClearWindow; { очистка экрана }
    j:=j+40;
  End;
  For e:=trunc(t1*10000) to trunc(t2*10000) do { цикл для построения кривой Уатта }
  Begin
    m:=sqrt(4*k*k+R*R-4*k*R*cos(e/10000));

```

```

sv:= R*sin(e/10000)/m; cv:=(2*k-R*cos(e/10000))/m; v:=Arctan(sv/cv); {sin v, cos
v, v}
cw:=(2*k*cos(e/10000)-3*R)/m; sw:=Sqrt(1-cw*cw); w:=pi-Arctan(sw/cw); {sin w,
cos w, w}
u:=w-v;
SetPixel(trunc(-k+R*cos(e/10000))+x0,trunc(R*sin(e/10000))+y0,clRed);    {дуга
окружности M1M2}
SetPixel(trunc(k+R*cos(u))+x0,trunc(R*sin(u))+y0,clRed);    {дуга  окружности
N1N2}

SetPixel(trunc((R*cos(e/10000)+R*cos(u))/2)+x0,trunc((R*sin(e/10000)+R*sin(u))/2)+
y0,_ clBlue);{точка кривой Уатта}
For s:=0 to 1000 do {цикл задержки}
s:=s+1;
End;
m:=sqrt(4*k*k+r*r-4*k*r*cos(t1/3));{построение одного положения механизма
Ватта}
sv:=R*sin(t1/3)/m; cv:=(2*k-R*cos(t1/3))/m; v:=arctan(sv/cv);{sin v, cos v, v}
cw:=(2*k*cos(t1/3)-3*R)/m; sw:=Sqrt(1-cw*cw);w:=pi-Arctan(sw/cw);{sin w, cos w, w}
u:=w-v;
Line(-trunc(k)+x0,y0,trunc(-k+r*cos(t1/3))+x0,Trunc(R*sin(t1/3))+y0); {AM}
Line(trunc(-
k+R*cos(t1/3))+x0,trunc(R*sin(t1/3))+y0,trunc(k+r*cos(u))+x0,Trunc(R*sin(u))+y0)
;{MN}
Line(trunc(k)+x0,y0,trunc(k+R*cos(u))+x0,trunc(R*sin(u))+y0); {BN}
SetPenStyle(psDot);Line (-trunc(k)+x0,y0, trunc(k)+x0,y0); {отрезок AB}
Line(-trunc(k)+x0,y0,trunc(-
k+(r+2*c)*cos(t1))+x0,Trunc((R+2*c)*sin(t1))+y0);{AN1}
Line(trunc(k)+x0,y0,trunc(-
k+(r+2*c)*cos(t1))+x0,Trunc((R+2*c)*sin(t1))+y0);{BN1}
Line(-trunc(k)+x0,y0,trunc(-k+R*cos(t2))+x0,trunc(R*sin(t2))+y0); {AM2}
Line(trunc(k)+x0,y0,trunc(-k+R*cos(t2))+x0,trunc(R*sin(t2))+y0); {BM2}
End.

```

Найдите в Википедии определение механизма Хойкена и разработайте для него математическое и компьютерное моделирование.

Рассмотрим вывод общего уравнения и полярного уравнения для кривой Уатта.

Пусть вектор \overline{MN} образует с положительным направлением оси Ox угол φ и пусть $M(x_1; y_1), U(x; y), N(x_2; y_2)$, тогда

$$x - x_1 = c \cos \varphi, y - y_1 = c \sin \varphi, \quad x_2 - x = c \cos \varphi, y_2 - y = c \sin \varphi.$$

соответствующие положения механизма: AM_1N_1B и AM_2N_2B . Обозначим $OU_1 = h$ и найдем величину h из прямоугольного треугольника AHM_1

$$h = \sqrt{R^2 - (k - c)^2}.$$

Формул для нахождения корней уравнения шестой степени не существует, но некоторые уравнения шестой степени можно приближенно решить с любой степенью точности, например, методом деления пополам диапазона изменения переменной.

Рассмотрим идею приближенного построения кривой Уатта.

Для переменной y получена оценка $-h \leq y \leq h$, причем переменная y принимает все значения из отрезка $[-h; h]$. Для переменной x можно получить простую оценку $-k \leq x \leq k$. Если переменную y зафиксировать, то переменную x можно приближенно найти из уравнения (4) на отрезке $[0; k]$. После нахождения точки $(x; y)$ достаточно построить симметричные точки $(x; -y)$, $(-x; y)$, $(-x; -y)$.

Реализацию этой идеи можно осуществить с помощью двух вложенных циклов. Во внешнем цикле переменная y изменяется от значения $-h$ до значения h . Во внутреннем цикле приближенно решается уравнение (4) с заданной степенью точностью.

Напишите программу построения рис. 15 и сравните ее с программой «**Приближенное построение кривой Уатта**».

Найдем полярный угол u_0 , соответствующий точке с координатами $x = 0$, $y = 0$. Полагая $\rho = 0$, получим $tg u_0 = \pm R/c$. Угол u_0 имеет простой геометрический смысл. Если точка U кривой Уатта совпадает с серединой отрезка AB (рис. 11), то $tg u_0 = tg \angle AOM = AM / MO = R/c$.

При моделировании кривой Уатта полярный угол должен изменяться в пределах $u \in [u_0; \pi - u_0] \cup [\pi + u_0; 2\pi - u_0]$.

Кривая имеет касательные в начале координат

$$y = \pm \frac{R}{c} x.$$

При моделировании кривой Уатта полярный угол должен изменяться в пределах $u \in [u_0, \pi - u_0] \cup [\pi + u_0, 2\pi - u_0]$.

После построения кривой Уатта можно построить механизм в движении.

Для точки M получаем координаты

$$x_1 = x - \frac{x^2 + y^2 + k^2 + c^2 - R^2}{2k}, \quad y_1 = y - x \frac{k^2 + R^2 - c^2 - x^2 - y^2}{2ky}.$$

Для точки N — координаты

$$x_2 = x + \frac{x^2 + y^2 + k^2 + c^2 - R^2}{2k}, \quad y_2 = y + x \frac{k^2 + R^2 - c^2 - x^2 - y^2}{2ky}.$$

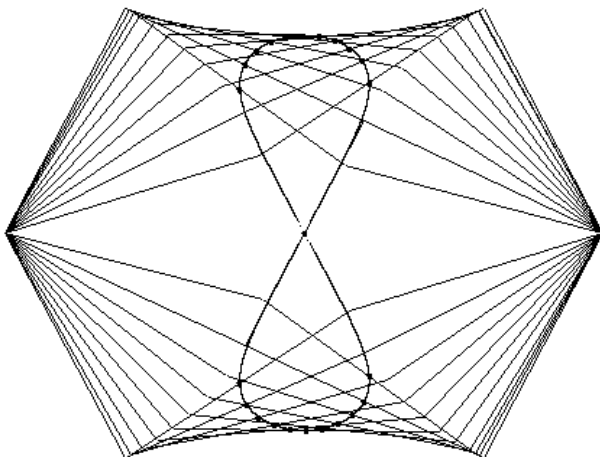


Рис. 16

Напишите компьютерную программу построения кривой Уатта и движения механизма Уатта, используя полярные координаты (рис. 16), и сравните ее с программой «Механизм Уатта в полярных координатах».

Обращаем внимание читателя на тот факт, что при приближении значения полярного угла к предельному значению механизм быстро изменяет скорость движения (рис. 16). Для моделирования значений угла, близких к предельному значению, необходимо повторять блок цикла с малым шагом изменения параметра.

§ 8. Изображение пространственной фигуры на плоскости

Уточним понятие «изображение фигуры».

Пусть заданы плоскость α и прямая l , которая не лежит в этой плоскости и не параллельна плоскости (рис. 1).

Для произвольной (плоской или пространственной) фигуры F рассмотрим проекцию каждой точки этой фигуры на плоскость α вдоль направления прямой l , т.е. через каждую точку M фигуры F проведем прямую, параллельную данной прямой l и найдем точку пересечения \tilde{M} с плоскостью α .

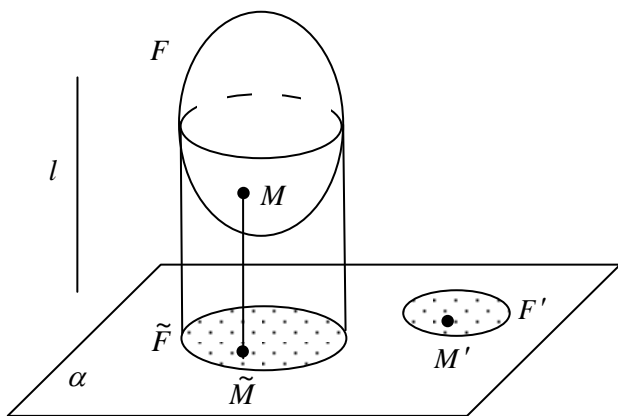


Рис. 1

Пусть \tilde{F} — проекция фигуры F на плоскость α вдоль направления l и F' — фигура, подобная фигуре \tilde{F} , тогда плоская фигура F' называется изображением фигуры F на плоскость α вдоль направления l при параллельном проектировании. Фигура F называется оригиналом для изображения F' .

Если прямая l перпендикулярна плоскости α , то проектирование называется ортогональным.

Каждая фигура подобна самой себе с коэффициентом, равным единице, поэтому в качестве изображения фигуры F можно рассматривать фигуру \tilde{F} , т.е. проекцию фигуры. А зачем усложнять определение, если его можно упростить, т.е. всегда ли можно называть изображением фигуры ее проекцию?

Если здание спроектировать на плоскость, то его проекция займет иногда несколько сот квадратных метров. Перенести такое изображение в другое место не очень удобно, да и зачем тратить столько бумаги, если проекцию можно сжать.

Если устройство часового механизма или платы компьютера спроектировать на плоскость, то проекцию часто приходится рассматривать с помощью лупы или микроскопа. В этом случае проекцию лучше увеличить. Таким образом, каждую проекцию можно назвать изображением фигуры, но лучше использовать фигуру, подобную проекции.

Если заданы плоскость α , прямая l и фигура F , то изображение всегда можно построить, но не однозначно, т.к. если построено одно изображение, то все фигуры, подобные этому изображению, также являются изображениями фигуры F .

Рассмотрим в некотором смысле обратную задачу. Даны две фигуры F и F' . Можно ли одну из них, например F' , считать изображением другой фигуры? Рассмотрим конкретный пример.

В плоскости α задан треугольник ABC , а в плоскости α' задан треугольник $A'B'C'$ (рис. 2). Можно ли считать треугольник $A'B'C'$ изображением треугольника ABC , т.е. можно ли указать направление проектирования треугольника ABC в некоторый треугольник, а затем преобразование подобия этого вспомогательного треугольника в треугольник $A'B'C'$?

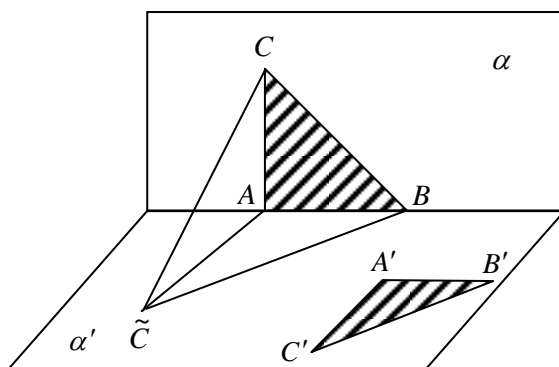


Рис. 2

Рисунок подтверждает, что такую последовательность действий можно указать.

На отрезке AB построим треугольник $ABC\tilde{}$, подобный треугольнику $A'B'C'$. Проектирование плоскости α на плоскость α' в направлении прямой $C\tilde{C}$ переводит треугольник ABC в треугольник $ABC\tilde{}$, а затем подобием переведем треугольник $ABC\tilde{}$ в треугольник $A'B'C'$.

Итак, доказано важное свойство:

Любой треугольник $A'B'C'$ можно считать изображением данного треугольника ABC .

Параллельное проектирование и подобие отображает точки, принадлежащие одной прямой, в точки, принадлежащие одной прямой; отображает параллельные прямые в параллельные прямые.

Параллельное проектирование и подобие сохраняет отношение длин отрезков на одной или на параллельных прямых. Сохраняется простое отношение трех точек на прямой, т.е. если три точки A, B, C оригинала лежали на одной прямой, то соответствующие точки A', B', C' изображения также лежат на одной прямой и выполняется равенство:

$$\frac{AB}{BC} = \frac{A'B'}{B'C'}.$$

Построение изображений многоугольников выполняется на основе этих свойств.

Изображение пространственных фигур выполняется по тем же правилам, что и для плоских фигур, но с использованием основного закона изображения тетраэдра:

Вершины любого четырехугольника $A'B'C'D'$ вместе с диагоналями могут служить изображением данного тетраэдра $ABCD$ (рис. 3).

Пусть точка $B(0;1;0)$ проектируется в точку $B'(1;0)$. Подставляя в формулы (1), получим $b_1 = 1, b_2 = 0$.

Пусть точка $C(0;0;1)$ проектируется в точку $C'(0;1)$. Подставляя в формулы (1), получим $c_1 = 0, c_2 = 1$.

Формулы (1) для параллельного проектирования примут вид

$$s = -px + y, \quad w = -qx + z. \quad (2)$$

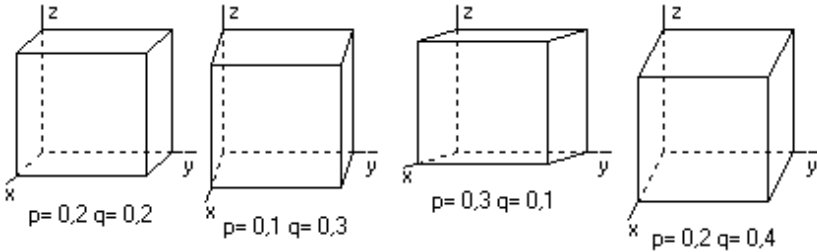


Рис. 5

Программа «Изображение куба — 1» построения изображения куба с системой координат (рис. 5) имеет вид:

```
Private Sub Куб(s0, w0, a, p, q)
sa = 0: wa = 0: sb = -p * a: wb = -q * a
sc = -p * a + a: wc = -q * a: sd = a: wd = 0
DrawStyle = 0
Line (sb + s0, wb + w0)-(sc + s0, wc + w0)
Line (sc + s0, wc + w0)-(sd + s0, wd + w0)
sa1 = 0: wa1 = a: sb1 = -p * a: wb1 = -q * a + a
sc1 = -p * a + a: wc1 = -q * a + a: sd1 = a: wd1 = a
Line (sa1 + s0, wa1 + w0)-(sb1 + s0, wb1 + w0)
Line (sb1 + s0, wb1 + w0)-(sc1 + s0, wc1 + w0)
Line (sc1 + s0, wc1 + w0)-(sd1 + s0, wd1 + w0)
Line (sd1 + s0, wd1 + w0)-(sa1 + s0, wa1 + w0)
Line (sb + s0, wb + w0)-(sb1 + s0, wb1 + w0)
Line (sc + s0, wc + w0)-(sc1 + s0, wc1 + w0)
Line (sd + s0, wd + w0)-(sd1 + s0, wd1 + w0)
sa2 = 0: wa2 = 1.2 * a: sb2 = -p * 1.4 * a: wb2 = -q * 1.4 * a
sd2 = 1.2 * a: wd2 = 0
Line (sd + s0, wd + w0)-(sd2 + s0, wd2 + w0)
Line (sb + s0, wb + w0)-(sb2 + s0, wb2 + w0)
Line (sa1 + s0, wa1 + w0)-(sa2 + s0, wa2 + w0)
DrawStyle = 2
Line (sa + s0, wa + w0)-(sa1 + s0, wa1 + w0)
Line (sa + s0, wa + w0)-(sb + s0, wb + w0)
```

```

Line (sd + s0, wd + w0)-(sa + s0, wa + w0)
PSet (sb2 + s0, wb2 + w0 + 50), vbWhite: Print "x"
PSet (sd2 + s0 - 100, wd2 + w0), vbWhite: Print "y"
PSet (sa2 + s0 + 50, wa2 + w0 + 50), vbWhite: Print "z"
PSet (sb + s0 + 100, wb + w0 - 200), vbWhite
Print "p="; p; "q="; q
End Sub
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
Куб 1000, 1000, 1000, 0.2, 0.2: Куб 2400, 1000, 1000, 0.1, 0.3
Куб 4200, 1000, 1000, 0.3, 0.1: Куб 5800, 1000, 1000, 0.2, 0.4
End Sub

```

При моделировании конусов, пирамид, призм и цилиндров наиболее сложным этапом является моделирование основания. Вначале разрабатывается плоская модель основания, а затем, добавляя пространственные координаты, завершается моделирование пространственной фигуры.

Например, требуется построить с помощью программы конус, основание которого состоит из полуокружности и прямоугольника.

Начинаем моделировать основание (рис. 6), затем переходим к пространственной фигуре. Используя формулы проектирования (1), получаем фигуру, изображенную на рис. 7.

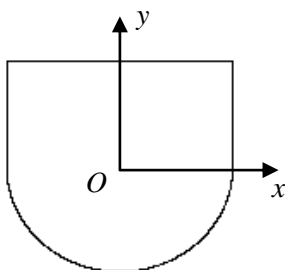


Рис. 6

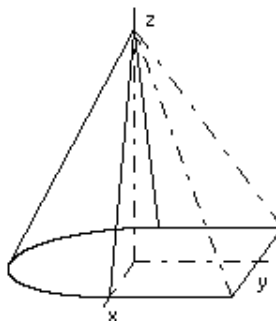


Рис. 7

Программа построения основания имеет вид:

```

Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 3000: y0 = 3000: r = 1000: pi = 4 * Atn(1)
xa = -r: ya = 0: xc = r: yc = 0: xd = r: yd = r: xe = -r: ye = r
For t = pi To 2 * pi Step 0.001
x = r * Cos(t): y = r * Sin(t)

```

End Sub

Конус изображен в привычной для нас системе координат, но восприятие его затруднено, т.к. сопоставляя его с изображением основания на рис. 6, получаем, что один из этих рисунков нужно поворачивать. Первый совет в возникшей ситуации — перед моделированием на плоскости выбрать систему координат, согласованную с системой координат в пространстве.

Второй совет — если моделирование на плоскости уже совершено, что часто случается, то нужно уйти от стереотипа изображения системы координат в привычном виде. Рассмотрим систему координат, изображенную на рис. 8.

С точки зрения геометрии в этих системах координат принципиальных различий нет, т.к. одна из них получается из другой поворотом вокруг оси Oz .

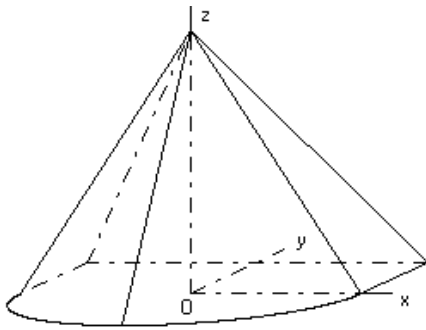


Рис. 8

Но построение такой системы координат и объектов в ней на экране компьютера требует другого способа проектирования.

Легко заметить, что оси Ox и Oy переставляются, причем направление оси Oy меняется на противоположное, а следовательно, формулы (2) преобразуются в формулы проектирования

$$s = x + py, \quad w = qy + z. \quad (3)$$

Задача. Рассмотрите вывод формул (3) без ссылки на формулы (2), используя проектирование точек (рис. 9):

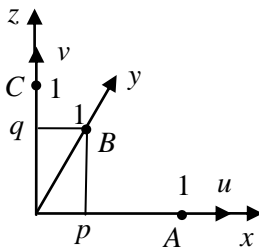


Рис. 9

- точка $A(1;0;0)$ проектируется в точку $A'(1;0)$;
- точка $B(0;1;0)$ проектируется в точку $B'(p;q)$;
- точка $C(0;0;1)$ проектируется в точку $C'(0;1)$.

Программа «**Конус над полукругом**» имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
s0 = 3000: w0 = 3000: p = 0.4: q = 0.2: r = 1500: h = 2500: pi = 4 * Atn(1)
so = 0: wo = 0
sa = -r: wa = 0: sb = -p * r: wb = -q * r
sc = r: wc = 0: sd = r + p * r: wd = q * r
se = -r + p * r: we = q * r
For t = pi To 2 * pi Step 0.001
x = r * Cos(t): y = r * Sin(t)
sm = x + p * y: wm = q * y
PSet (sm + s0, wm + w0)
Next t
ss = 0: ws = h
Line (sc + s0, wc + w0)-(sd + s0, wd + w0)
Line (sa + s0, wa + w0)-(ss + s0, ws + w0)
Line (sb + s0, wb + w0)-(ss + s0, ws + w0)
Line (sc + s0, wc + w0)-(ss + s0, ws + w0)
Line (sd + s0, wd + w0)-(ss + s0, ws + w0)
sz = 0: wz = 1.1 * h: sy = p * 1.4 * r: wy = q * 1.4 * r
sx = 1.2 * r: wx = 0
sf = p * r: wf = q * r
Line (ss + s0, ws + w0)-(sz + s0, wz + w0)
Line (sc + s0, wc + w0)-(sx + s0, wx + w0)
DrawStyle = 3
Line (sc + s0, wc + w0)-(sd + s0, wd + w0)
Line (sd + s0, wd + w0)-(se + s0, we + w0)
Line (se + s0, we + w0)-(sa + s0, wa + w0)
Line (sd + s0, wd + w0)-(se + s0, we + w0)
Line (se + s0, we + w0)-(sa + s0, wa + w0)
Line (so + s0, wo + w0)-(sc + s0, wc + w0)
Line (so + s0, wo + w0)-(sy + s0, wy + w0)
Line (so + s0, wo + w0)-(ss + s0, ws + w0)
Line (se + s0, we + w0)-(ss + s0, ws + w0)
PSet (sz + s0 + 100, wz + w0), vbWhite: Print "z"
PSet (sy + s0 + 100, wy + w0 + 200), vbWhite: Print "y"
PSet (sx + s0 + 50, wx + w0 + 50), vbWhite: Print "x"
PSet (s0 - 80, w0 - 20), vbWhite: Print "O"
End Sub
```

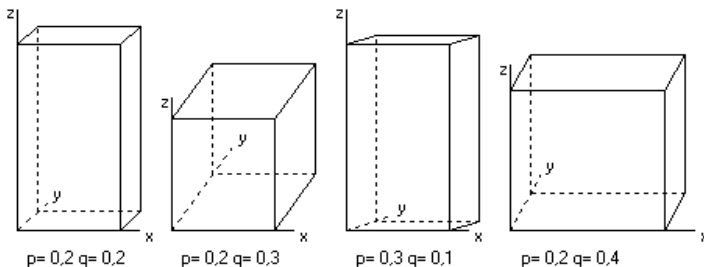



Рис. 10

1.35. Разработайте программу для построения семейства прямоугольных параллелепипедов (рис. 10) с длинами сторон a, b, c и сравните ее с программой «Параллелепипед».

Для некоторых проекций удобно изображать систему координат согласно рис. 11.

Точки $A(1;0;0)$, $B(0;1;0)$, $C(0;0;1)$, в пространственной системе координат изображаются точками

$$A'(p_1; q_1), B'(p_2; q_2), C'(0; 1).$$

Формулы параллельного проектирования принимают вид

$$s = p_1x + p_2y, \quad w = q_1x + q_2y + z. \quad (4)$$

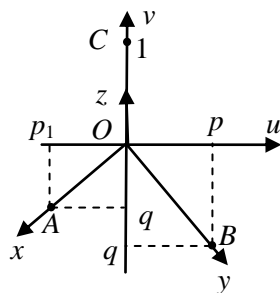


Рис. 11

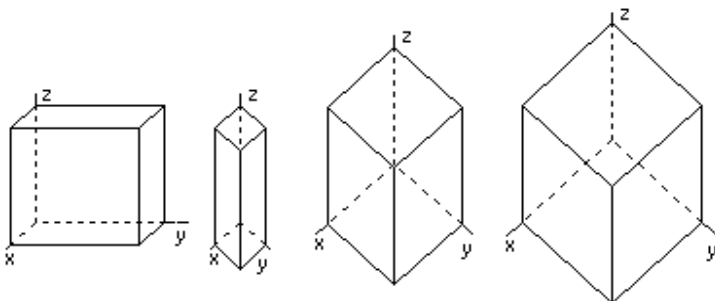


Рис. 12

1.36. Напишите программу построения семейства кубов, изображенных на рис. 12, и сравните ее с программой «Параллелепипед повернутый».

Заметим, что при таком изображении две противоположные вершины куба могут совпасть. Такое изображение не является наглядным, т.к. диагональ, соединяющая эти вершины, изображается парой совпавших точек.

§ 9. Закрашивание областей в разные цвета

Большие возможности закрашки различных областей можно реализовать в исполнителе **Чертежник** [4].

Познакомимся с операторами компьютерной графики в этом исполнителе, чтобы понимать, в каком виде далее определять простейшие линии.

цикл ($k=0 ; k \leq 11 ; k=k+0.1$) { }

for (; ;) { } — объявление цикла с начальным значением, конечным значением параметра и законом изменения параметра

{ } — в скобках перечисляются операторы цикла

если ($p > 0$)

if ($p == 2$) — условные операторы, проверяющие условие

Точка (,)

Point (,); — построение точки

Прямоугольник (, , ,)

Rectangle (, , ,) — построение прямоугольника, для которого указываются координаты вершин левого верхнего угла и правого нижнего угла.

Эллипс (, , ,)

Ellipse (, , ,) — построение эллипса, касающегося прямоугольника. Для прямоугольника указываются координаты вершин левого верхнего угла и правого нижнего угла. Если прямоугольник является квадратом, то эллипс является окружностью.

Дуга (, , , , ,)

Arc (, , , ,) — построение дуги, для которой указываются координаты двух углов прямоугольника, внутри которого расположена дуга, а также начальное и конечное значения углов концевых точек. Для формирования цвета используется функция

RGB (red, green, blue)

Команда для установки свойств кисти имеет английский и русский варианты

Brush (Color, Style);

Кисть (Color, Style);

Эта процедура принимает два параметра:

Color	Числовое значение цвета, полученного с помощью функции RGB , например, RGB(10, 20, 40) .			
Style	Стиль кисти, который выбирается из следующих значений:			
	0	сплошная заливка	4,5	диагональная штриховка
	1	нет заливки	6	сетчатая штриховка
	2	горизонтальная штриховка	7	диагональная сетка
	3	вертикальная штриховка		

Проанализируем простую программу построения раскраски прямоугольной доски в два цвета (рис. 1).

```
{ целые n,m,a,i,j,x0,y0,p,q;
n=6;m=5;a=24;x0=5;y0=5
//построение сетки из квадратов
цикл ( i=0;i<=n-1 ;i=i+1 )
цикл ( j=0;j<=m-1 ;j=j+1 )
{Прямоугольник (i*a+x0,j*a+y0,(i+1)*a+x0,(j+1)*a+y0);}
//закраска области
цикл ( i=0;i<=n-1 ;i=i+1 )
цикл ( j=0;j<=m-1 ;j=j+1 )
{p=i+j; q=p-p/2*2;
если (q= =0 )
{Кисть (RGB ( 0,0,0),0);
Заливка (i*a+a/2+x0,j*a+a/2+y0);}}
```

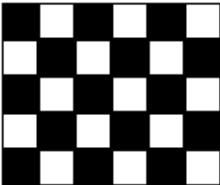


Рис. 1

Вначале с помощью вложенных циклов выполняется построение сети из квадратов со стороной, равной *a*. В горизонтальном

ряду расположено n квадратов, а в вертикальном ряду m квадратов. Переменные p , q будут использоваться для определения четности номеров клеток.

0+0	1+0	2+0	3+0	4+0	5+0
0+1	1+1	2+1	3+1	4+1	5+1
0+2	1+2	2+2	3+2	4+2	5+2
0+3	1+3	2+3	3+3	4+3	5+3
0+4	1+4	2+4	3+4	4+4	5+4

Рис. 2

Вначале вычисляется сумма параметров $p = i + j$ (рис. 2). При переходе от одной клетке к смежной клетке четность числа p меняется.

Для определения четности числа применяется следующий прием. Вначале применяется целочисленное деление $p/2$, затем умножение полученного частного на число 2. Если число p четное, то $p/2*2 = p$. Если число p нечетное, то $p/2*2 = p-1$. Для четного p получаем $p-p/2*2=0$, а для нечетного p получаем $p-p/2*2=1$.

Число переменных в программе для определения четности можно уменьшить, например, полагая $q=(i+j)-(i+j)/2*2$.

1.37. Напишите программу закраски фигуры на рис. 3. Сравните ее с программой «Закраска квадрата разными цветами». (Путь к программе: Robotwin, Robot.exe, Файл, Открыть).

Будем предполагать, что в каждой задаче рисунка с заливкой требуется осуществлять моделирование с учетом дальнейшего изменения параметров различных линий на рисунке. Это позволит в дальнейшем изменять расположение линий семейства, сохраняя его структуру, и получать различные рисунки.

Чтобы стороны прямоугольника не оказались на границе экрана компьютера, построение фигуры начинается от точки (x_0, y_0) , координаты которой заданы в начале программы.

Второй блок операторов состоит из двух вложенных циклов и предназначен для указания центров квадратов, которые заливается черным цветом.

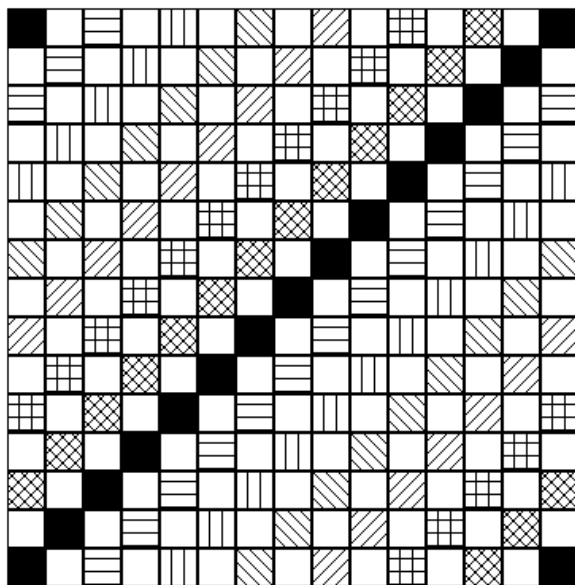


Рис. 3

Рассмотрим реализацию этапов моделирования построения более сложной заливки на примере рис. 4.

При решении задач с заливкой областей можно выделить два основных этапа.

I. Построение сети под будущую заливку.

Выделение одного или нескольких семейств линий сети на рисунке, которые образуют замкнутые области для заливки. Построение каждого семейства линий должно осуществляться в зависимости от соответствующего параметра. В некоторых задачах расположение одного семейства линий относительно другого семейства также можно задать в виде зависимости от третьего параметра.

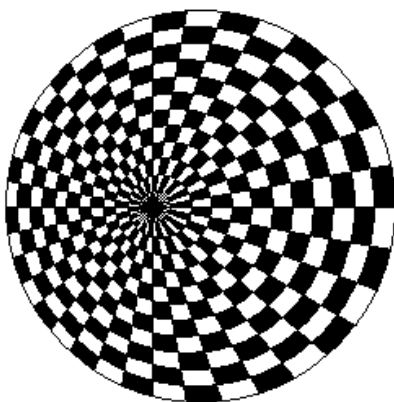


Рис. 4

Визуальное прочтение метрических размеров для каждого семейства на рисунке может быть различным, поэтому и моделирование сети может быть различным. При увеличении конечных значений параметров циклов в каждой модели можно обнаружить различия в разработанных моделях.

II. Моделирование схемы закраски областей.

Выделение закономерности чередования цветов для раскраски.

Поиск последовательности точек внутри закрашенных областей или всех областей, удобных для автоматизации повторения в цикле. Точки лучше выбирать по одной в каждой области и зависящие от одного или нескольких параметров.

Определение координат последовательности точек для каждой закрашиваемой области.

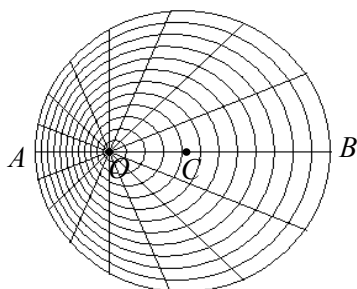


Рис. 5

Выбор метода закраски определенных областей указанным цветом.

I. Сеть на рис. 4 состоит из семейства окружностей и семейства хорд окружностей, проходящих через одну точку.

Анализируя рисунок, будем предполагать, что центры всех окружностей расположены на одной прямой.

Обозначим:

O — общая точка всех хорд окружностей;

AB — диаметр окружности наибольшего радиуса, на котором расположены центры всех окружностей (рис. 5);

R — радиус наибольшей окружности;

C — центр наибольшей окружности;

c — длина отрезка OC ;

n — число секторов с вершиной в точке O ;

k — число окружностей семейства.

Почти все геометрические параметры фигуры заданы. Осталось определить зависимость смещения центров окружностей. Анализируя рис. 4, будем предполагать, что отрезки AO и OB делятся на равное число частей и затем на полученных отрезках, как на диаметрах, строятся окружности.

Рассмотрим моделирование элементов семейства окружностей (рис. 6).

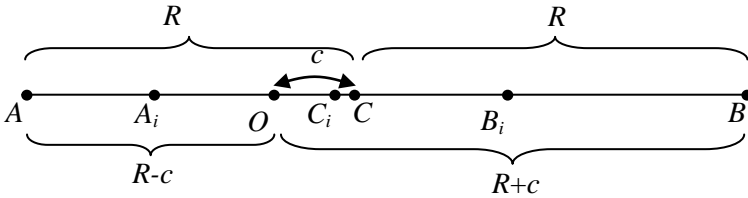


Рис. 6

Длины следующих отрезков равны: $AO = R - c$, $OB = R + c$.

Систему координат на плоскости выберем с началом в точке O , а ось Ox направим вдоль луча $[OB)$.

Для окружности семейства с номером i , диаметром A_iB_i , центром O_i и радиусом R_i получаем координаты точек и величину радиуса:

$$A_i\left(-\frac{i(R-c)}{k}; 0\right), B_i\left(\frac{i(R+c)}{k}; 0\right), O_i\left(\frac{ic}{k}; 0\right), R_i = \frac{iR}{k}.$$

Для построения окружности достаточно указать координаты угловых точек описанного квадрата:

$$E_i\left(\frac{ic}{k} - \frac{iR}{k}; -\frac{iR}{k}\right), F_i\left(\frac{ic}{k} + \frac{iR}{k}; \frac{iR}{k}\right).$$

Рассмотрим моделирование семейства хорд окружности. Анализируя рис. 4, будем предполагать, что все хорды делят полный угол в точке O на n равных частей, тогда угол между двумя соседними хордами равен $2\pi/n$. Пусть вектор ON_j образует с осью Ox угол, равный $u = 2\pi j/n$ (рис. 7), тогда прямая ON_j имеет параметрические уравнения

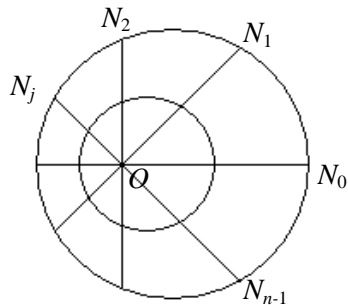


Рис. 7

$$x = t \cos u, y = t \sin u. \quad (1)$$

Подставляя значения (1) в уравнение внешней окружности $(x-c)^2 + y^2 = R^2$, найдем значения параметра t точек пересечения прямой ON_j с окружностью:

$$t_1 = c \cos u + \sqrt{R^2 - c^2 \sin^2 u}, t_2 = c \cos u - \sqrt{R^2 - c^2 \sin^2 u}.$$

Выбирая любое из этих значений и подставляя в равенства (1), получим координаты точки N_j на окружности.

Отрезок ON_j можно построить по координатам концевых точек.

II. Рассмотрим реализацию второго этапа.

Цвета смежных областей чередуются в кольце между двумя соседними окружностями.

Цвета смежных областей чередуются в секторе между двумя соседними лучами.

Пусть нумерация областей определяется с помощью текущих параметров двух циклов.

При переходе из одной области в смежную область меняется только один из параметров. Используя четность или нечетность суммы двух параметров, можно сформулировать критерий, когда нужно выполнять закраску областей.

Если бы все закрашиваемые области были равны между собой, то, используя сдвиг на некоторый постоянный вектор, можно было

быстро подобрать последовательность точек на основе моделирования областей на предыдущем этапе.

В нашем случае почти все области различные, за исключением некоторых из них, которые симметричны относительно оси Ox .

Мысленно представим более густую сеть на плоскости, которая получается, если между двумя построенными смежными окружностями дополнительно построить по одной окружности и каждый сектор разделить на два сектора биссектрисой угла. На рис. 8 к построенной сети добавлены линии первого семейства (окружности) и линии

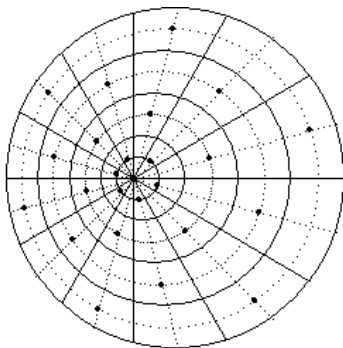


Рис. 8

к построенной сети добавлены линии первого семейства (окружности) и линии

второго семейства (лучи) пунктирными линиями. Для моделирования линий каждого семейства можно использовать циклы, аналогичные предыдущим циклам при решении проблемы на первом этапе, но изменить диапазоны параметров $0 \leq i \leq 2k-1$, $0 \leq j \leq 2n-1$.

На пересечении линий новой сети выберем по точке. Половину из них нужно использовать для закрашки областей. На рисунке они выделены жирным шрифтом. При переходе от одной точки к другой в кольце между двумя смежными окружностями величина параметра будет изменяться на 2. Аналогично, при переходе от одной точки к другой в секторе между двумя смежными радиусами величина параметра будет изменяться на 2.

Критерием построения точки на пересечении линий новой сети является признак: делится ли сумма параметров $i + j$ на 4.

Изучите подробности компьютерной реализации этой идеи в следующей программе.

Программа «Смещенные окружности» в Robotwin

```
{ дробные s,Ri,xi,u0,u,t2,x2,y2,m;
целые c,R, i,k,j,n,p,q,x0,y0;
R=120;c=30;k=12;n=36,x0=200;y0=300;s=3.1416;u0=2*s/n
// построение окружностей семейства
цикл (i=k ;i>=1 ;i=i-1)
{ xi=i*c/k; Ri=i*R/k;
  Эллипс (xi-Ri+x0,-Ri+y0,xi+Ri+x0,y0+Ri); }
//Построение хорд семейства
цикл (j=1 ;j<=n ;j=j+1)
{ u=j*u0;
  m= (R^2-c^2*(sin(u))^2)^0.5 ;
  t2=c*cos(u)+m;x2=cos(u)*t2; y2=sin(u)*t2;
  Линия ( x0,y0 ,x2+x0 ,y2+y0); }
// закрашка областей
Кисть (RGB ( 0,0,0),0);
цикл (i=2*k-1 ;i>=1 ;i=i-2)
{ xi=i*c/k/2; Ri=i*R/k/2;
  цикл (j=1 ;j<=2*n-1 ;j=j+2)
  { u=j*u0/2;
    m= (Ri^2-xi^2*(sin(u))^2)^0.5 ;
    t2=xi*cos(u)+m;x2=cos(u)*t2; y2=sin(u)*t2;
    q=(i+j)/4;p=(i+j)-q*4
    if ( p>0 )
      Заливка (x2+x0 ,y2+y0); } }
```

1.38. Разработайте математическое и компьютерное моделирование закрашки следующих областей (рис. 9—13):



Рис. 9

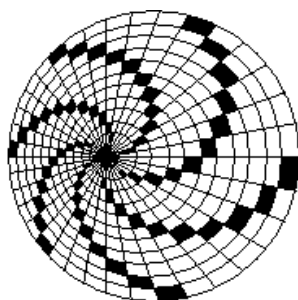


Рис. 10

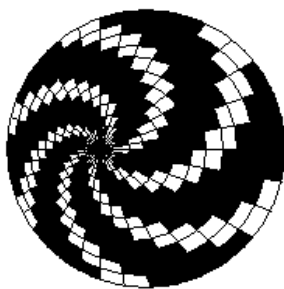


Рис. 11

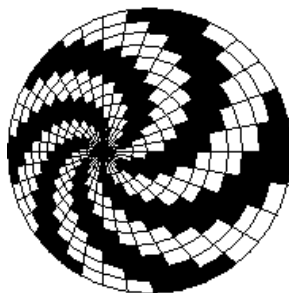


Рис. 12

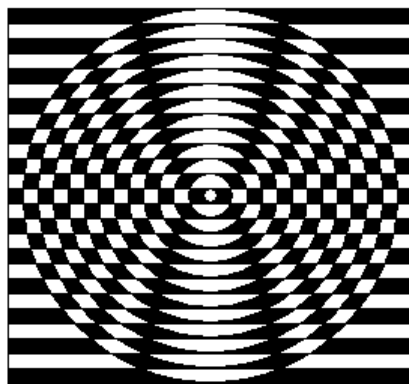


Рис. 13

Иногда сетку для заливки можно быстро рассчитать, а для координат точек последовательности заливки приходится рассматривать много различных случаев. В этой ситуации легче построить сетку для заливки, скопировать сетку, вставить в Paint, а затем раскрасить в Paint, используя простую заливку.

На рис. 14 построен треугольник Релло, который является фигурой постоянной ширины. Он состоит из трех дуг, центры которых расположены в вершинах равностороннего треугольника. Каждая дуга разделена на n равных частей и из каждой вершины равностороннего треугольника построены секторы.

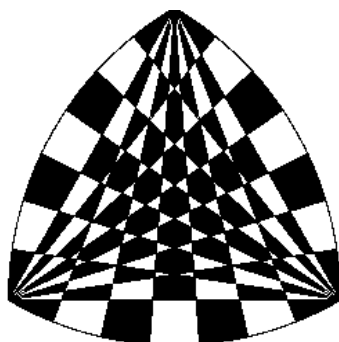


Рис. 14

1.39. Напишите программу построения сетки для треугольника Релло, имеющего ширину R и $3n$ секторов. Сравните ее с программой «Сетка в треугольнике Релло».

1.40. Верхняя сторона и боковые стороны прямоугольника (рис. 15) разделены на равные части. Проведите дальнейший анализ построения рис. 15 и постройте сетку для заливки фигуры.

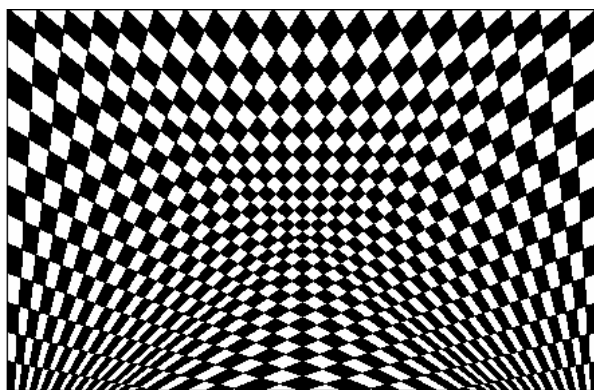


Рис. 15

Рассмотрим моделирование одной из замечательных теорем элементарной математики — теоремы Морлея для произвольного треугольника [11].

Пусть в треугольнике ABC проведены трисектрисы углов — $AA_1, AA_2, BB_1, BB_2, CC_1, CC_2$, т.е. отрезки, делящие углы на три равные части (рис. 16), тогда треугольник MNK является равно-
сторонним.

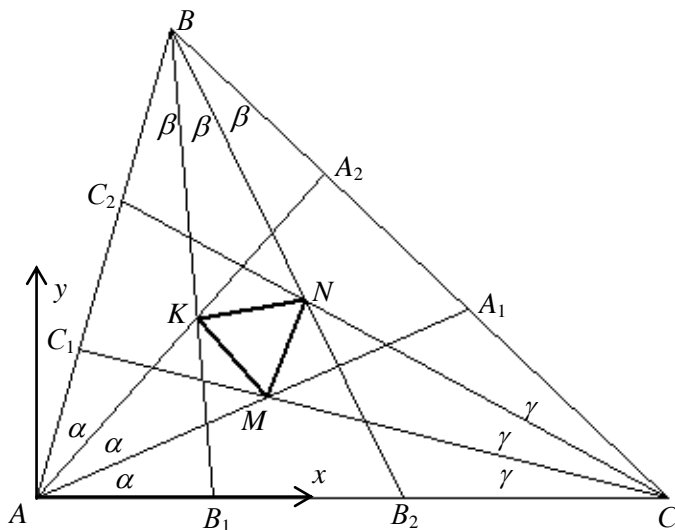


Рис. 16

Пусть данный треугольник имеет элементы $AC = b$, $\angle A = 3\alpha$, $\angle C = 3\gamma$. Обозначим $\angle B = 3\beta$, тогда $\beta = 60^\circ - \alpha - \gamma$. Используя теорему синусов, найдем остальные элементы треугольника:

$$c = AB = \frac{b \sin 3\gamma}{\sin(3\alpha + 3\gamma)}, a = DC = \frac{b \sin 3\alpha}{\sin(3\alpha + 3\gamma)}.$$

Для построения трисектрис найдем длины отрезков и координаты точек для системы координат, представленной на рис. 16.

$$AA_1 = \frac{b \sin 3\gamma}{\sin(\alpha + 3\gamma)}, AA_2 = \frac{b \sin 3\gamma}{\sin(2\alpha + 3\gamma)},$$

$$A_1 \left(\frac{b \sin 3\gamma \cos \alpha}{\sin(\alpha + 3\gamma)}, \frac{b \sin 3\gamma \sin \alpha}{\sin(\alpha + 3\gamma)} \right), A_2 \left(\frac{b \sin 3\gamma \cos 2\alpha}{\sin(2\alpha + 3\gamma)}, \frac{b \sin 3\gamma \sin 2\alpha}{\sin(2\alpha + 3\gamma)} \right),$$

$$AC_1 = \frac{b \sin 3\alpha}{\sin(\gamma + 3\alpha)}, AC_2 = \frac{b \sin 3\alpha}{\sin(2\gamma + 3\alpha)},$$

$$C_1 \left(\frac{b \sin 3\alpha \cos 3\alpha}{\sin(\gamma + 3\alpha)}, \frac{b \sin 3\alpha \sin 3\alpha}{\sin(\gamma + 3\alpha)} \right), C_2 \left(\frac{b \sin 3\alpha \cos 3\alpha}{\sin(2\gamma + 3\alpha)}, \frac{b \sin 3\alpha \sin 3\alpha}{\sin(2\gamma + 3\alpha)} \right)$$

$$AB_1 = \frac{c \sin \beta}{\sin(\beta + 3\alpha)}, AB_2 = \frac{c \sin 2\beta}{\sin(2\beta + 3\alpha)},$$

$$B_1 \left(\frac{c \sin \beta}{\sin(\beta + 3\alpha)}, 0 \right), B_2 \left(\frac{c \sin 2\beta}{\sin(2\beta + 3\alpha)}, 0 \right).$$

Для построения равностороннего треугольника MNK найдем длины отрезков и координаты точек

$$AM = \frac{b \sin \gamma}{\sin(\alpha + \gamma)}, M \left(\frac{b \sin \gamma \cos \alpha}{\sin(\alpha + \gamma)}, \frac{b \sin \gamma \sin \alpha}{\sin(\alpha + \gamma)} \right),$$

$$AK = \frac{c \sin \beta}{\sin(\alpha + \beta)}, K \left(\frac{c \sin \beta \cos 2\alpha}{\sin(\alpha + \beta)}, \frac{c \sin \beta \sin 2\alpha}{\sin(\alpha + \beta)} \right),$$

$$CN = \frac{a \sin \beta}{\sin(\gamma + \beta)}, K \left(b - \frac{a \sin \beta \cos 2\gamma}{\sin(\gamma + \beta)}, \frac{a \sin \beta \sin 2\gamma}{\sin(\gamma + \beta)} \right).$$

Разработайте компьютерную программу построения рис. 16 и сравните ее с программой «**Теорема Морлея**».

Многократное применение теоремы синусов для моделирования деления углов на три равные части пора применить для построения рису. 17—20, в которых все углы делятся на заданное число n равных частей.

Если в геометрии точное деление угла на заданное число частей является иногда неразрешимой задачей, то в информатике эта проблема решается проще после проведенного математического моделирования. Но мы должны помнить, что всякое компьютерное построение является приближенным построением, т.к. используются только целочисленные значения координат точек экрана. Большая точность вычислений обеспечивает преимущество компьютерного построения над геометрическими построениями. Возможность вставлять рисунок в текст и возможность пересылки этого рисунка в сети дополняют преимущества компьютерного построения.

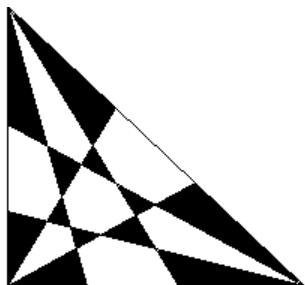


Рис. 17

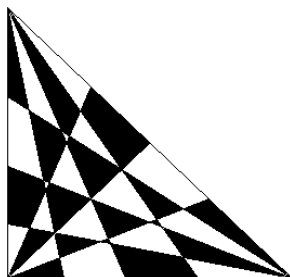


Рис. 18

Но самое большое преимущество компьютерного построения рисунков в этой задаче заключается в следующем. Компьютерная программа является очень компактной, причем она быстро делит углы на произвольное число частей.

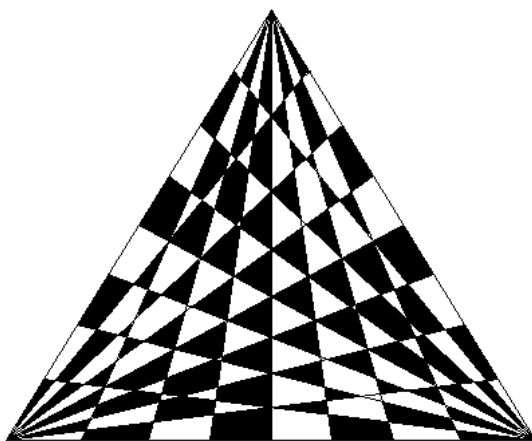


Рис. 19

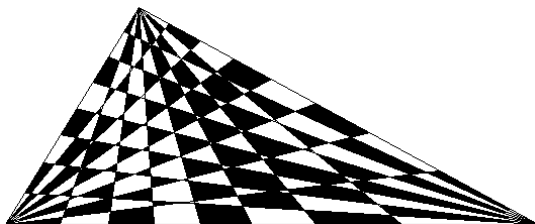


Рис. 20

Напишите программу построения этих рисунков и сравните ее с программой «Углы треугольника на равные части».

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 1000: y0 = 2500: pi = 4 * Atn(1):
b = 8000: n = 8: al = 60 * pi / 180: gm = 30 * pi / 180
al = al / n: gm = gm / n: bt = pi / n - al - gm
c = b * Sin(n * gm) / Sin(n * al + n * gm)
a = b * Sin(n * al) / Sin(n * al + n * gm)
Line (x0, y0)-(b + x0, y0)
Line (x0, y0)-(c * Cos(n * al) + x0, c * Sin(n * al) + y0)
Line (b + x0, y0)-(c * Cos(n * al) + x0, c * Sin(n * al) + y0)
For i = 1 To n - 1
xai = b * Sin(n * gm) * Cos(i * al) / Sin(i * al + n * gm)
yai = b * Sin(n * gm) * Sin(i * al) / Sin(i * al + n * gm)
xci = b - b * Sin(n * al) * Cos(i * gm) / Sin(i * gm + n * al)
yci = b * Sin(n * al) * Sin(i * gm) / Sin(i * gm + n * al)
xbi = c * Sin(i * bt) / Sin(i * bt + n * al)
Line (x0, y0)-(xai + x0, yai + y0)
Line (b + x0, y0)-(xci + x0, yci + y0)
Line (c * Cos(n * al) + x0, c * Sin(n * al) + y0)-(xbi + x0, y0)
Next i
End Sub
```

Все фигуры на рис. 17—20 удалось раскрасить в два цвета. А можно ли раскрасить в два цвета сетку для любого натурального числа n ?

§ 10. Пунктирная линия и радианная мера дуги

Рассмотрим построение отрезка на экране компьютера с помощью параметрических уравнений $x = x_0 + mt, y = y_0 + nt$, где $m^2 + n^2 = 1$, а параметр t изменяется с шагом, равным 1, на отрезке $[0; T]$.

Требуется изобразить этот отрезок пунктирной линией, на которой длина черты равна T_1 и длина пропуска равна T_2 (рис. 1), где T_1, T_2 — натуральные числа.

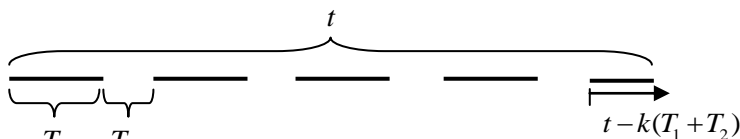


Рис. 1

Для произвольного значения параметра t необходимо отделить целое число периодов $T_1 + T_2$ и на остатке $t - k(T_1 + T_2)$ осуществлять построение точки, если этот остаток меньше или равен T_1 , и не выполнять построение точки, если остаток больше T_1 .

Пример программы построения пунктирного горизонтального отрезка.

```
Private Sub Command1_Click()
    T = 4100: T1 = 500: T2 = 300
    For i = 0 To T
        v = i Mod (T1 + T2): If v <= T1 Then PSet (300 + i, 3000)
    Next i: End Sub
```

Недостаток этого моделирования состоит в том, что последний штрих может оказаться короче, чем предыдущие.

1.41. В начале программы задайте целое число штрихов, которые должны появиться на отрезке, задайте коэффициент, показывающий, во сколько раз длина штриха должна превосходить длину пропуска и далее постройте отрезок пунктирной линией. Напишите соответствующую программу.

В новой программе есть свой недостаток, т.к. конец отрезка на экране не будет виден. Точки отрезка на последнем этапе не изображены, т.к. для них подразумевается пропуск.

1.42. Измените программу так, чтобы отрезок начинался со штриха и заканчивался штрихом, т.е. число штрихов на отрезке должно быть на один больше, чем число пропусков. Для упрощения программы можно длину штриха и длину пропуска считать равными.

Следующая программа строит отрезок пунктирной линией от точки $(x_1; y_1)$ до точки $(x_2; y_2)$ с помощью параметрических уравнений $x = (1-t)x_1 + tx_2$, $y = (1-t)y_1 + ty_2$, где $t \in [0; 1]$ с заданным числом штрихов n .


```

Private Sub Command1_Click()
X1 = 200: Y1 = 300: X2 = 5000: Y2 = 2000: n = 7: t0 = 1 / (2 * n - 1)
For t = 0 To 1 Step 0.001
k = Int(t / (2 * t0)): r = t - k * 2 * t0
x = (1 - t) * X1 + t * X2: y = (1 - t) * Y1 + t * Y2: If r <= t0 Then PSet (x, y)
Next t: End Sub

```

Построим пунктирной линией окружность с заданным числом n штрихов.

Вначале определяем период (рис. 2), для которого нужно нарисовать одну дугу и совершить один пропуск, т.е. $t_0 = 2\pi / n$.

Для произвольного значения угла t определяем максимальное количество целых значений k периодов, которые содержатся в этом значении угла. Если разность $t - 2k\pi / n$ меньше, чем $t_0 / 2$, то рисуем точку окружности для этого значения t , в противном случае точка не рисуем.

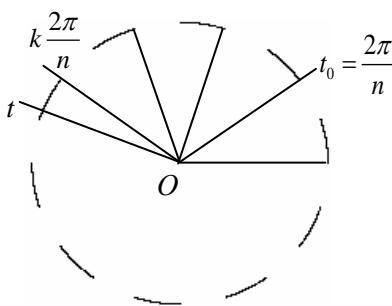


Рис. 2

```

Private Sub Command1_Click()
pi = 4 * Atn(1): R = 1400: n = 10: t0 = 2 * pi / n: x0 = 4000: y0 = 3000
For t = 0 To 2 * pi Step 0.001
k = Int(t / t0): x = R * Cos(t): y = R * Sin(t)
If (t - k * t0 < pi / n) Then PSet (x + x0, y + y0)
Next t: End Sub

```

Длина окружности вычисляется по формуле $C = 2\pi R$. Этой формулой часто пользуются при вычислении длины забора вдоль здания, имеющего цилиндрическую форму, при вычислении длины опалубки для будущего фундамента круглого здания и т.д.

Существует класс проблем и задач, для которых в качестве меры длины дуги окружности или сектора принимается величина центрального угла, опирающегося на дугу, т.е. радианная мера дуги окружности.

В радиолокации угол обзора определяется как разность величин углов тех радиусов, между которыми находится объект на экране монитора.

В компьютерной графике дуга окружности (или эллипса) задается с помощью углов, определяющих начало и конец дуги.

CIRCLE (x, y), ... , ... , ... , ... , ... — оператор рисования окружности (дуги эллипса при указании начала и конца дуги).

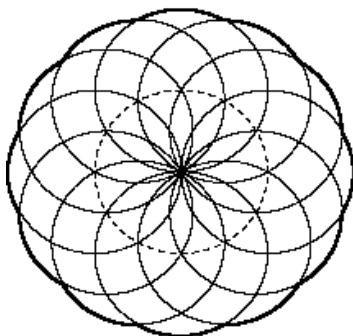


Рис. 3

В некоторых задачах начальное и конечное значения дуги приходится точно рассчитывать.

Пусть построено n окружностей $\omega_0, \omega_1, \dots, \omega_{n-1}$ радиуса R , центры которых находятся на окружности γ также радиуса R и пусть центры семейства окружностей делят окружность γ на равные части.

Все окружности проходят через точку O , являющуюся центром окружности γ (рис. 3).

Требуется выделить жирной линией контур фигуры, образованной семейством окружностей $\omega_0, \omega_1, \dots, \omega_{n-1}$.

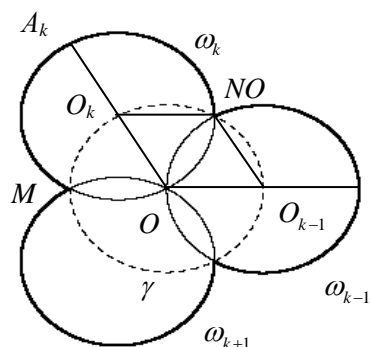


Рис. 4

Очевидно в этом случае, что необходимо определить значения параметров для точек пересечения двух смежных окружностей семейства и указать начальное и конечное значения параметра при построении точек окружности.

Пусть окружность ω_k и ω_{k-1} пересекаются со смежными окружностями ω_{k-1} и ω_{k+1} , соответственно, в точках N и M , тогда

четырехугольник $OO_{k-1}NO_k$ является ромбом (рис. 4).

Векторы $\overline{OO_k}$ и $\overline{OO_{k-1}}$ образуют с осью Ox , соответственно, углы $\frac{k2\pi}{n}$ и $\frac{(k-1)2\pi}{n}$, $\angle O_{k-1}OO_k = \frac{2\pi}{n}$, $\angle NO_kA_k = \frac{2\pi}{n}$.

Точки N и M имеют, соответственно, параметры $\frac{(k-1)2\pi}{n}$ и $\frac{(k+1)2\pi}{n}$.

Постройте фигуры, представленные на рис. 5—8.

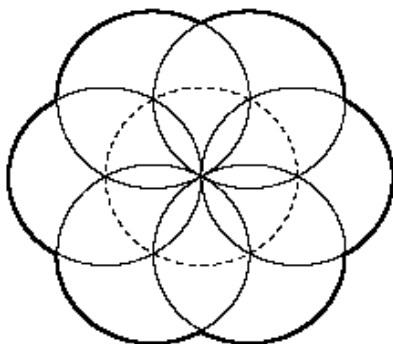


Рис. 5

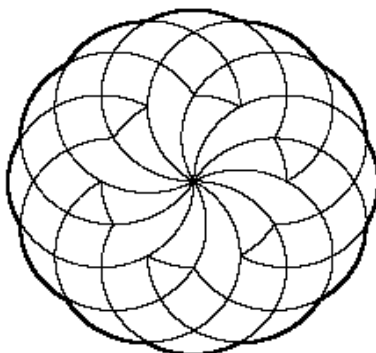


Рис. 6

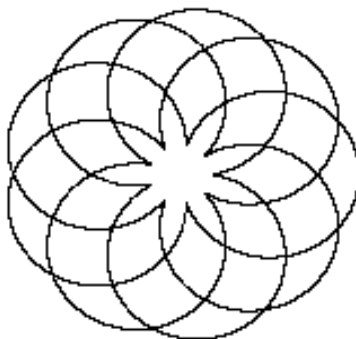


Рис. 7

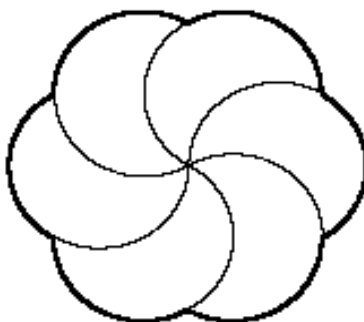


Рис. 8

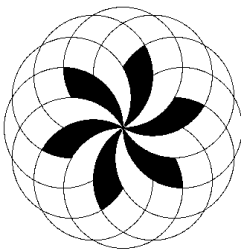


Рис. 9

1.43. На рис. 9 проведена штриховка некоторых областей. Построение такого рисунка и штриховки выполнено в исполнителе Чертежник, изучаемом в курсе информатики. Познакомьтесь с программой **«Заливка областей между окружностями.prg»** построения этого рисунка и продолжите его раскраску. (Путь к программе: Robotwin, Robot.exe, Файл, Открыть).

§ 11. Седловые поверхности в строительстве

Пример 1. Возьмем два листа плотного картона (два куса фанеры или оргстекла), наложим один из них на другой, построим на одном листе окружность и просверлим на окружности отверстия через два листа. Разъединим листы картона и закрепим на них нити одинаковой длины или резинки, т.е. изготовим простейшую модель цилиндра (рис. 1).

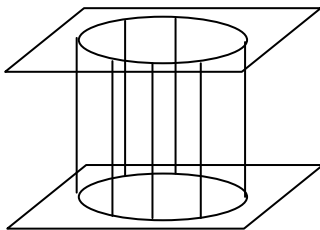


Рис. 1

Удерживая верхнее основание цилиндра одной рукой, повернем нижнее основание другой рукой так, чтобы оно вращалось вокруг оси цилиндра. На рис. 2 изображено несколько положений образующих линий для случая, когда расстояние между верхним и нижним основаниями остается неизменным, т.е. когда между основаниями натянуты резинки.

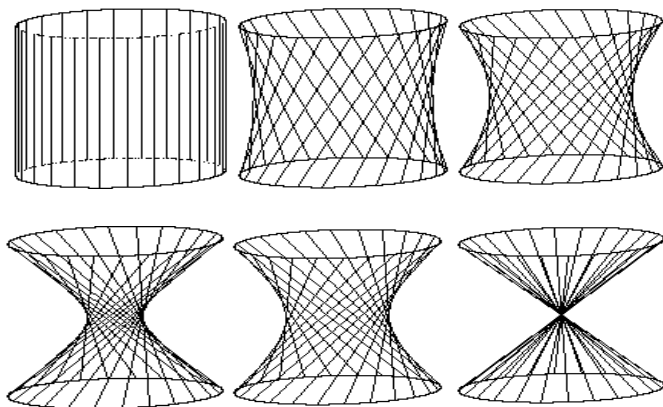


Рис. 2

Если между основаниями натянуты нити, то нижнее основание при таких операциях будет приподниматься вверх. Если на поверхность цилиндра нанести более густую сеть прямых линий, то при поворотах получим поверхности, которые изменяются от цилиндра, затем проходят различные виды однополостных гиперболоидов, и, наконец, образуют конус.

1.44. Напишите программу закручивания образующих линий на гиперболоиде и сравните ее с программой «Закручивание образующих на гиперболоиде».

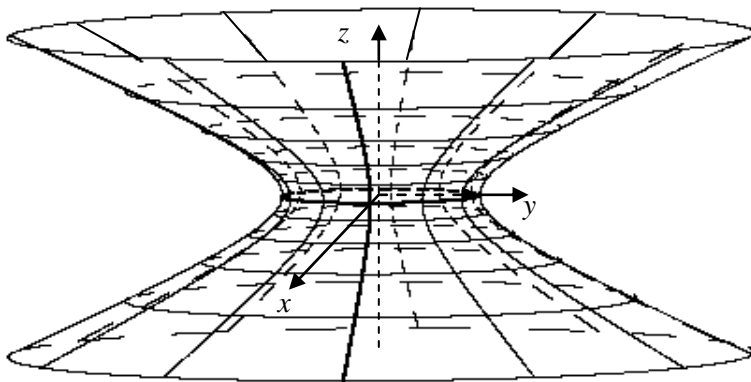


Рис. 3

На рис. 3 изображен однополостный гиперboloид вращения. Самый короткий замкнутый цикл на поверхности вокруг оси вращения называется *горловым эллипсом*. Если система координат выбрана в центре гиперboloида, ось Oz направлена вдоль оси гиперboloида, а плоскость Oxy содержит горловой эллипс, то гиперboloид вращения имеет каноническое уравнение:

$$\frac{x^2}{a^2} + \frac{y^2}{a^2} - \frac{z^2}{c^2} = 1.$$

Для компьютерного построения лучше использовать параметрические уравнения:

$$x = a \operatorname{ch} v \cos u, \quad y = a \operatorname{ch} v \sin u, \quad z = c \operatorname{sh} v$$

или

$$x = a \frac{e^v + e^{-v}}{2} \cos u, \quad y = a \frac{e^v + e^{-v}}{2} \sin u, \quad z = c \frac{e^v - e^{-v}}{2}.$$

1.45. Напишите программу построения однополостного гиперboloида и сравните ее с программой «Однополостный гиперboloид вращения».

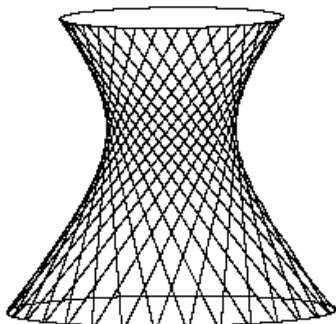


Рис. 4

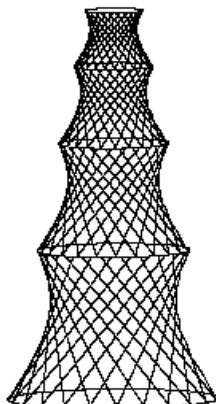


Рис. 5

Однополостный гиперboloид интересен тем, что на его поверхности расположено два семейства прямых, которые называются *прямолинейными образующими*. На рис. 4 изображены прямые двух семейств, расположенные на передней части однополостного гиперboloида.

Форма однополостного гиперboloида используется в строительной технике. Использование прямолинейных образующих в различных башнях сочетает прочность конструкции и простоту ее исполнения. Идея использования формы однополостного гиперboloида с прямолинейными образующими принадлежит русскому инженеру В.Г.Шухову (1853—1939). Телевизионная башня на Шаболовке в Москве сооружена по системе Шухова и кажется состоящей из однополостных гиперboloидов вращения с общей осью. В действительности она состоит не из гиперboloидов, а из прямолинейных конструкций, которые установлены по окружности под определенным углом (рис. 5). При рассмотрении этой конструкции со стороны человеческий глаз дополняет ее до огибающей и идентифицирует с однополостными гиперboloидами, если знаком с такими поверхностями. Красота, изящество и самое главное — практическая польза от использования прямых линий поражают оригинальностью.

Первая в мире конструкция в виде гиперboloида разработана В.Г.Шуховым в 1896 г. в Полибине (рис. 6).



Рис. 6



Рис. 7

Первый проект башни на Шаболовке был разработан В.Г.Шуховым в 1919 г. с расчетной высотой 350 метров. Из-за дефицита

Однополостный гиперболоид вращения можно получить при повороте одной из двух скрещивающихся прямых вокруг другой прямой.

$$z \uparrow \quad l \downarrow$$


136

Рассмотрим произвольную точку $M(x; y; z)$ на поверхности вращения. Через точку M проведем плоскость, перпендикулярную оси вращения. Пусть эта плоскость пересекает ось вращения в точке H , прямую l — в точке $M_1(x_1; a; z)$, а поверхность вращения — по окружности ω с центром в точке H .

Из равенства $HM = HM_1$ следует $x^2 + y^2 = x_1^2 + a^2$.

Точка M_1 расположена на прямой l , поэтому координаты этой точки удовлетворяют уравнению $z = kx$, т.е. $z = kx_1$.

Следовательно,

$$x^2 + y^2 = \left(\frac{z}{k}\right)^2 + a^2, \quad \frac{x^2}{a^2} + \frac{y^2}{a^2} = \frac{z^2}{k^2 a^2} + 1.$$

Обозначим $ka = c$, тогда получаем уравнение однополостного гиперболоида вращения:

$$\frac{x^2}{a^2} + \frac{y^2}{a^2} - \frac{z^2}{c^2} = 1.$$

Чтобы построить одно семейство прямолинейных образующих на однополостном гиперболоиде, можно выбрать пару точек $M_1(b_1; a; kb_1)$, $M_2(b_2; a; kb_2)$ на прямой l , провести отрезок M_1M_2 и затем применить поворот вокруг оси Oz :

$$x' = x \cos u - y \sin u,$$

$$y' = x \sin u + y \cos u,$$

$$z' = z.$$

Для наглядного изображения гиперболоида точки M_1 и M_2 нужно выбирать по разные стороны от плоскости Oxy , т.е. с различными по знаку аппликатами.

Для построения второго семейства прямолинейных образующих на однополостном гиперболоиде на прямой l можно выбрать пару точек $M_3(-b_1; a; kb_1)$, $M_4(-b_2; a; kb_2)$.

1.46. Напишите программу построения рис. 5 и сравните ее с программой «Модель телебашни на Шаболовке».

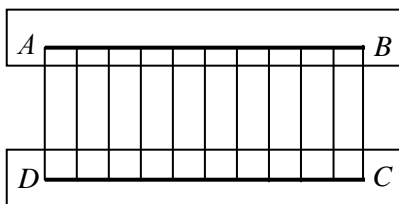


Рис. 9

Пример 2. Возьмем две пластины в виде прямоугольников, наложим одну на другую. Проведем на одной из них прямую, отметим на ней точки с равными расстояниями между ними. Просверлим отверстия в двух пластинах,

отодвинем пластины и натянем между ними резинки (рис. 9). Повернем один прямоугольник относительно второго так, чтобы просверленные отверстия оказались на скрещивающихся прямых AB и DC (рис. 10). Если резинок между пластинами натянуто много, то создается впечатление, что получена поверхность, состоящая из прямых. Такая поверхность называется *гиперболическим параболоидом*.

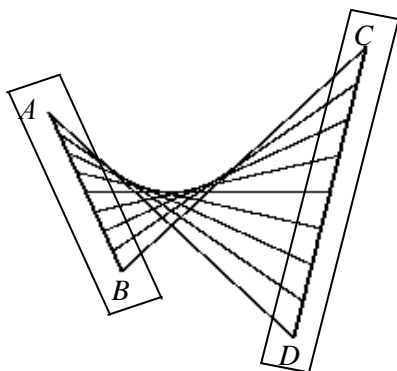


Рис. 10

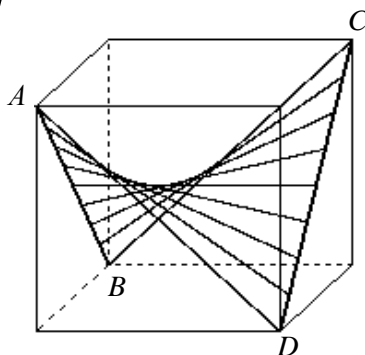


Рис. 11

На рис. 11 эта конструкция изображена снова в кубе. Отрезки AB и DC являются скрещивающимися диагоналями двух противоположных граней куба.

На рис. 12 построены две скрещивающиеся прямые AB и DC , где $A(0; -f; h_1)$, $C(0; f; h_1)$, $B(e; 0; h_2)$, $D(-e; 0; h_2)$.

На скрещивающихся прямых AB и DC на равных расстояниях друг от друга взяты точки:

на прямой AB точки $M_0 = A, M_1, M_2, \dots, M_n = B$;

на прямой DC точки $N_0 = D, N_1, N_2, \dots, N_n = C$.

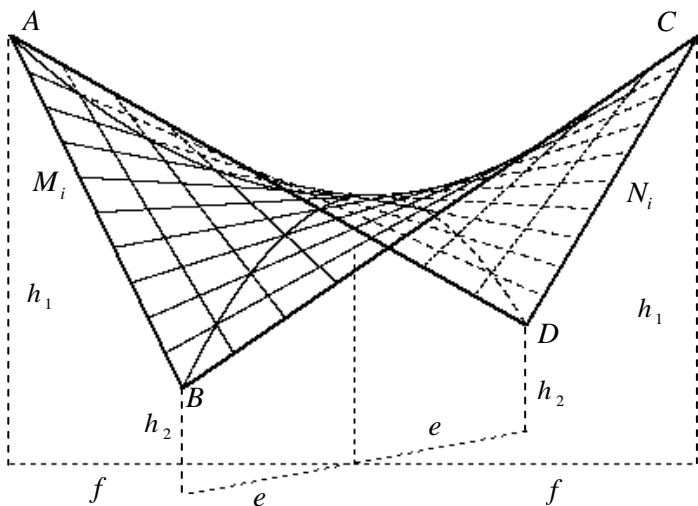


Рис. 12

Прямые $M_0N_0, M_1N_1, \dots, M_nN_n$ лежат на поверхности одного и того же гиперболического параболоида. Это одно семейство прямолинейных образующих на поверхности.

Аналогично соединяя соответствующие точки на двух скрещивающихся прямых BC и AD , получим второе семейство прямолинейных образующих на поверхности.

1.47. Напишите программу построения двух семейств гиперболического параболоида, порожденных двумя парами скрещивающихся прямых, и сравните ее с программой «Крыша гиперболического параболоида».

Найдем аналитическое задание построенного гиперболического параболоида.

Пусть $Q(x; y; z)$ — произвольная точка на прямой MN семейства прямых (рис. 13), где $M \in AB, N \in DC, AM = DN$ и $\overline{MQ} = v \overline{MN}$.

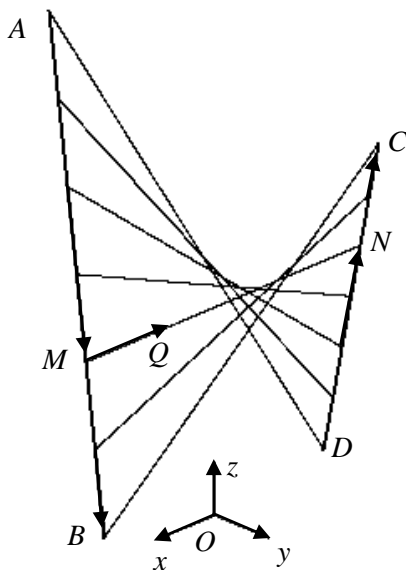


Рис. 13

Пусть $\overline{AM} = u \overline{AB}$, тогда $\overline{DN} = u \overline{DC}$ и

$$\overline{AB}(e; f; h_2 - h_1), \overline{AM}(eu; fu; (h_2 - h_1)u),$$

$$\overline{DC}(e; f; h_1 - h_2), \overline{DN}(eu; fu; (h_1 - h_2)u),$$

$$\overline{OM} = \overline{OA} + \overline{AM}, \overline{OM}(eu; fu - f; (h_2 - h_1)u + h_1),$$

$$\overline{ON} = \overline{OD} + \overline{DN}, \overline{ON}(eu - e; fu; (h_1 - h_2)u + h_2),$$

$$\overline{MN} = \overline{ON} - \overline{OM}, \overline{MN}(-e; f; (2u - 1)(h_1 - h_2)),$$

$$\overline{MQ}(-ev; fv; (2u - 1)(h_1 - h_2)v), \overline{OQ} = \overline{OM} + \overline{MQ},$$

$$\overline{OQ}(eu - ev); fu - f + fvv; (h_2 - h_1)u + h_1 + (2u - 1)(h_1 - h_2)v).$$

Для произвольной точки $Q(x; y; z)$ получаем выражение координат как функции от параметров u и v :

$$\begin{aligned} x &= e(u - v), y = f(u + v) - f, \\ z &= (h_1 - h_2)(2uv - u - v) + h_1. \end{aligned} \quad (1)$$

Выражая из первых двух уравнений параметры u и v и подставляя в третье уравнение, получаем аналитическое задание искомого множества точек:

$$z - \frac{h_1 + h_2}{2} = -\frac{x^2(h_1 - h_2)}{2e^2} + \frac{y^2(h_1 - h_2)}{2f^2}. \quad (2)$$

Это уравнение является частным случаем уравнения гиперболического параболоида

$$2(z - z_0) = -\frac{x^2}{a^2} + \frac{y^2}{b^2}.$$

Уравнения (1) называются *параметрическими уравнениями гиперболического параболоида*. Используя эти уравнения, можно быстрее моделировать два семейства прямолинейных образующих на поверхности. Изучите программу «**Параметризация гиперболического параболоида**» с использованием параметрических уравнений.

Программа «**Семейство прямых на гиперболическом параболоиде**» строит однопараметрическое семейство прямых и соответствующую систему координат, представленные на рис. 13. Варьирование параметров p_1, q_1, p_2, q_2 в формулах параллельного проектирования $s = p_1x + p_2y$, $w = q_1x + q_2y + z$ позволяет поворачивать систему координат, а за ней и семейство прямолинейных образующих.

Однополостный гиперboloид и гиперболический параболоид относятся к одному классу — классу седловых поверхностей.

Пример 3. Найдем множество точек пространства, равноудаленных от двух данных скрещивающихся прямых.

Пусть AB — общий перпендикуляр скрещивающихся прямых a и b , длину которого обозначим через $2c$ (рис. 14). Пусть a_1 — образ прямой a при параллельном переносе в точку B и b_1 — образ прямой b при параллельном переносе в точку A . Две пары прямых a и b_1 , b и a_1 определяют две параллельные плоскости.

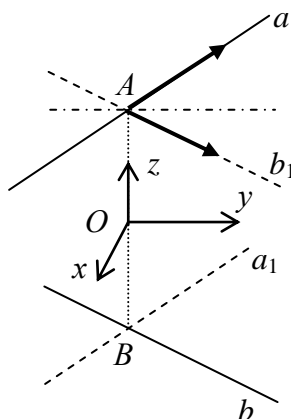


Рис. 14

В точке O , середине отрезка AB , выберем начало прямоугольной декартовой системы координат. Ось Oz направим вдоль луча $[OA)$, а оси Ox, Oy направим параллельно биссектрисам углов между прямыми a и b_1 . Для упрощения предстоящих вычислений система координат выбрана таким образом, что данные прямые a и b оказались параллельными координатной плоскости Oxy .

Рассмотрим направляющие векторы:

$\bar{p} \quad (-m;n;0)$ для прямой a и $\bar{q} \quad (m;n;0)$ для прямой b .

Для произвольной точки $M(x, y, z)$ искомого множества точек получаем $\rho(M, a) = \rho(M, b)$ или

$$\frac{|\overline{AM} \times \bar{p}|}{|\bar{p}|} = \frac{|\overline{BM} \times \bar{q}|}{|\bar{q}|}.$$

Векторы выбраны таким образом, что $|\bar{p}| = |\bar{q}|$, поэтому

$$|\overline{AM} \times \bar{p}| = |\overline{BM} \times \bar{q}|, \quad \overline{AM}(x; y; z - c), \quad \overline{BM}(x; y; z + c).$$

Равенство модулей векторных произведений в координатах примет вид:

$$\begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ x & y & z - c \\ -m & n & 0 \end{vmatrix} = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ x & y & z + c \\ m & n & 0 \end{vmatrix},$$

$$|(-n(z - c); -m(z - c); nx + my)| = |(-n(z + c); m(z + c); nx - my)|,$$

$$z = \frac{mny}{c(m^2 + n^2)}.$$

Это уравнение гиперболического параболоида. Применяя поворот $x = \frac{\sqrt{2}}{2}x' + \frac{\sqrt{2}}{2}y', y = -\frac{\sqrt{2}}{2}x' + \frac{\sqrt{2}}{2}y'$ в плоскости Oxy вокруг начала координат на угол (-45°) , получим уравнение

$z = \frac{mn(-x'^2 + y'^2)}{2c(m^2 + n^2)}$. В новой системе координат $Ox'y'z$

изображение поверхности является более наглядным.

При построении поверхности изобразим также данные прямые и оси координат.

На прямой b выберем две точки $X_1(-ml; -nl; -c)$, $X_2(ml; nl; -c)$, где l — некоторая константа, чтобы отрезок X_1X_2 наглядно изображался на рисунке. Координаты точек заданы в первоначальной системе координат. Аналогично на прямой a выберем две точки $X_3(ml; -nl; c)$, $X_4(-ml; nl; c)$.

В новой системе координат эти точки имеют координаты:

$$X_1\left(-\frac{ml\sqrt{2}}{2} + \frac{nl\sqrt{2}}{2}; -\frac{ml\sqrt{2}}{2} - \frac{nl\sqrt{2}}{2}; -c\right),$$

$$X_2\left(\frac{ml\sqrt{2}}{2} - \frac{nl\sqrt{2}}{2}; \frac{ml\sqrt{2}}{2} + \frac{nl\sqrt{2}}{2}; -c\right),$$

$$X_3\left(\frac{ml\sqrt{2}}{2} + \frac{nl\sqrt{2}}{2}; \frac{ml\sqrt{2}}{2} - \frac{nl\sqrt{2}}{2}; c\right),$$

$$X_4\left(-\frac{ml\sqrt{2}}{2} - \frac{nl\sqrt{2}}{2}; -\frac{ml\sqrt{2}}{2} + \frac{nl\sqrt{2}}{2}; c\right).$$

В области изменения переменных на плоскости $Ox'y'$ рассмотрим первое семейство параллельных прямых $x' = x'_0 = \text{const}$. Вычисляя значение функции для каждой точки этой прямой, получим график функции. Графиками функции будут параболы, ветви которых направлены вверх.

Для второго семейства параллельных прямых $y' = y'_0 = \text{const}$ графиками функции над этими прямыми будут параболы, ветви которых направлены вниз.

Два семейства парабол на поверхности образуют координатную сеть на поверхности. Одно семейство парабол является сечением гиперболического параболоида плоскостями $x' = x'_0$, а другое — сечением параболоида плоскостями $y' = y'_0$.

На рис. 15 с помощью программы «Гиперболический параболоид» построены прямые a и b и гиперболический параболоид.

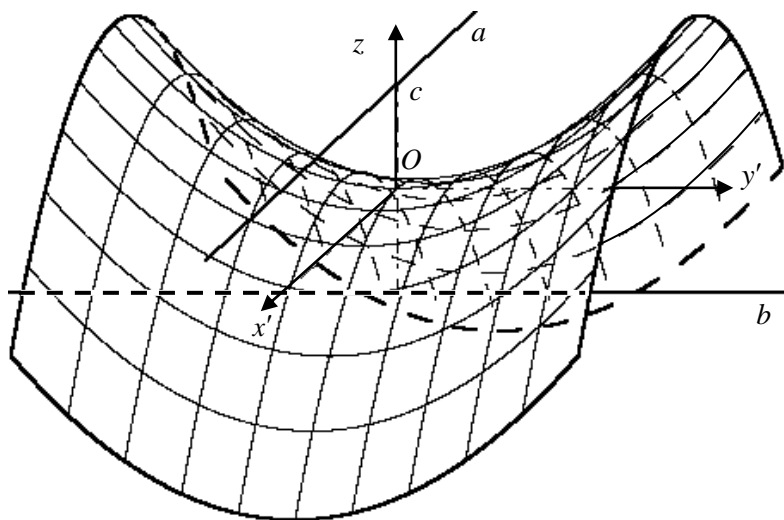


Рис. 15

На плоскости $Ox'y'$ можно рассмотреть два других семейства линий и над этими линиями построить графики функций.

Рассмотрим в плоскости $Ox'y'$ семейство концентрических окружностей $r = r_0 = const$ с центром в точке O . Графики функции над этими окружностями являются пересечениями параболоида с вертикальными цилиндрами (рис. 16). График построен с помощью программы «Концентрически-радиальная сеть на параболоиде».

Рассмотрим в плоскости $Ox'y'$ семейство лучей $t = t_0 = const$ с началом в точке O . Графики функции над этими лучами являются пересечениями параболоида с вертикальными плоскостями, проходящими через лучи. Среди этих линий на поверхности параболоида можно заметить две прямые, проходящие через точку O .

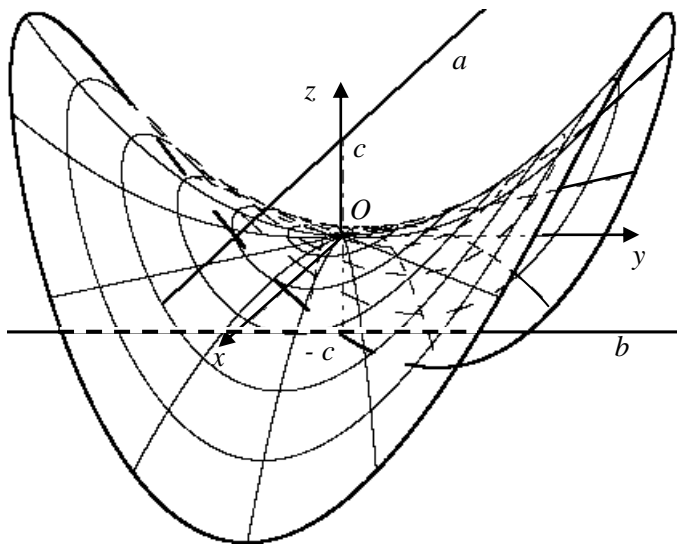


Рис. 16

Это можно обнаружить и аналитическим путем. Если в аналитическом задании параболоида положить $z=0$, то получим $-x'^2 + y'^2 = 0$. Это уравнение определяет пару прямых, заданных уравнениями $x' - y' = 0$ или $x' + y' = 0$. Таким образом, плоскость $z=0$ пересекает поверхность по паре прямых.

Вид гиперболического параболоида на рис. 14 похож на седло.

На рис. 17 изображен один цикл над окружностью.

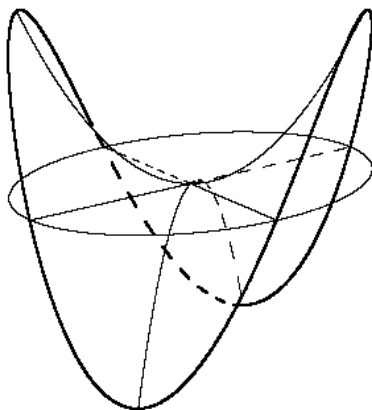


Рис. 17

Горизонтальные плоскости $z = z_0 = \text{const}$, где $z_0 \neq 0$, пересекают гиперболический параболоид по гиперболам. На рис. 18 ветви двух гипербол соединены пунктирными линиями, чтобы обозначить плоскости, в которых расположены гиперболы.

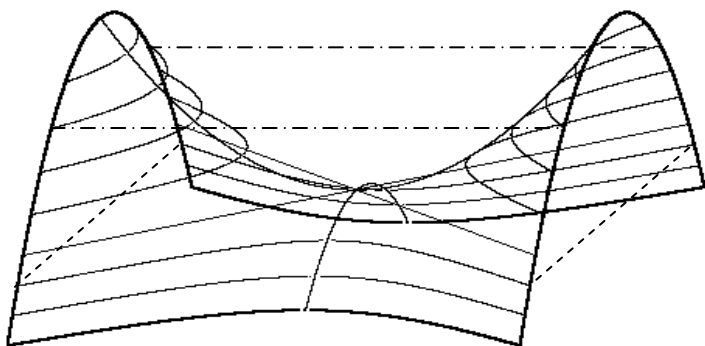


Рис. 18

1.48. Разработайте программу построения рис. 18 и сравните ее с программой «Линии уровня на гиперboloиде».

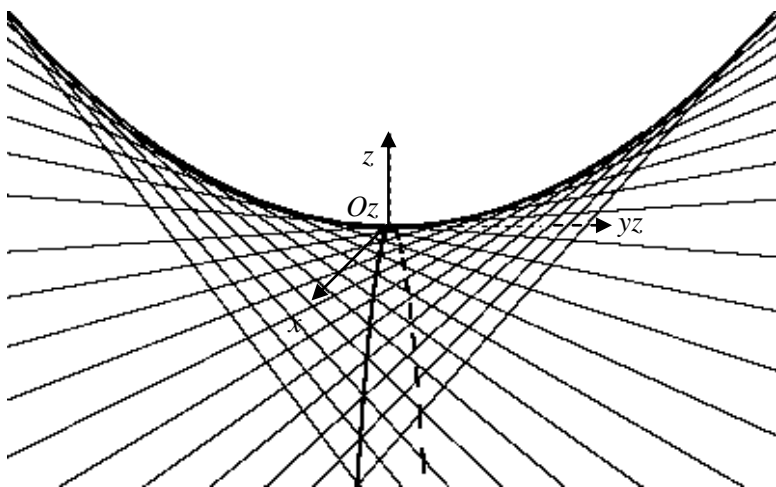


Рис. 19

Докажем утверждение о том, что через каждую точку гиперболического параболоида

$$2z = -\frac{x^2}{a^2} + \frac{y^2}{b^2}, \text{ где } a > 0, b > 0 \quad (3)$$

проходят две и только две образующие (рис. 19).

Действительно, пусть $(x_0; y_0; z_0)$ — произвольная точка гиперболоида (3), а $x = x_0 + lt, y = y_0 + mt, z = z_0 + nt$ — параметрические уравнения прямой, проходящей через точку $(x_0; y_0; z_0)$ и имеющей направляющий вектор $(l; m; n)$.

Для того, чтобы прямая полностью лежала на параболоиде, необходимо и достаточно, чтобы координаты произвольной точки прямой удовлетворяли уравнению параболоида при любом значении параметра t , т.е.

$$2(z_0 + nt) = -\frac{(x_0 + lt)^2}{a^2} + \frac{(y_0 + mt)^2}{b^2}$$

или

$$\left(\frac{m^2}{b^2} - \frac{l^2}{a^2}\right)t^2 + 2\left(\frac{my_0}{b^2} - \frac{lx_0}{a^2} - n\right)t - \frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} - 2z_0 = 0.$$

Точка $(x_0; y_0; z_0)$ принадлежит параболоиду, поэтому

$$-\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} - 2z_0 = 0.$$

Равенство $\left(\frac{m^2}{b^2} - \frac{l^2}{a^2}\right)t^2 + 2\left(\frac{my_0}{b^2} - \frac{lx_0}{a^2} - n\right)t = 0$ выполняется

для всех значений t тогда и только тогда, когда коэффициенты квадратного трехчлена равны нулю, т.е.

$$\frac{m^2}{b^2} - \frac{l^2}{a^2} = 0, \frac{my_0}{b^2} - \frac{lx_0}{a^2} - n = 0 \text{ или } m = \pm \frac{b}{a}l, n = \pm \frac{ly_0}{ab} - \frac{lx_0}{a^2}.$$

Получаем следующие направляющие векторы двух прямых, проходящих через точку $(x_0; y_0; z_0)$:

$$\left(l; \pm \frac{b}{a}l; \pm \frac{ly_0}{ab} - \frac{lx_0}{a^2}\right), \text{ где } l \in R, l \neq 0.$$

Для наглядного изображения прямолинейных образующих на параболоиде можно выбрать точки $(x_0; y_0; z_0)$ на параболе

$$2z = -\frac{x^2}{a^2} + \frac{y^2}{b^2}, y = 0,$$

расположенной в плоскости Oxz , т.е. прямые будем проводить через точки $(x_0, 0, -x_0^2/2a^2)$.

Направляющие векторы выберем в следующем виде:

$$(a; b; -x_0/a), (-a; b; x_0/a).$$

Первое семейство прямолинейных образующих:

$$x = at + x_0, y = bt, z = -\frac{x_0}{a}t - \frac{x_0^2}{a^2}, t \in R.$$

Второе семейство прямолинейных образующих:

$$x = -at + x_0, y = bt, z = \frac{x_0}{a}t - \frac{x_0^2}{a^2}, t \in R.$$

На рис. 19 построены первое и второе семейства прямолинейных образующих для точек $(x_0, 0, -x_0^2/2a^2)$, где $x_0 \geq 0$ на параболке.

1.49. Напишите программу построения аналогичного рисунка и сравните ее с программой «**Образующие на гиперболическом параболоиде**».

Прямолинейные образующие гиперболического параболоида и однополостного гиперboloида используются в архитектуре и строительстве [5]. Иногда крышу павильона изготавливают в виде нескольких сегментов гиперболического параболоида.

Замечание. Уравнения однополостного гиперboloида и гиперболического параболоида во всех учебниках по геометрии предлагается для исследования без пояснений, каким образом появляются эти уравнения. Мы решили геометрические задачи, приводящие к этим уравнениям, и подробно исследовали сеть из образующих таким образом, чтобы удобно было моделировать на компьютере.

При фиксированном значении параметра $u = u_0$ уравнения

$$\begin{aligned} x &= a \sin u t + a \cos u, y = -b \cos u t + b \sin u, z = ct, \\ u &\in [0; 2\pi], t \in R \end{aligned} \quad (2)$$

определяют прямую с направляющим вектором $(a \sin u; -b \cos u; c)$ и проходящую через точку $(a \cos u, b \sin u, 0)$. Эта прямая проходит через точку $(x_0; y_0; 0)$, расположенную

на горловом эллипсе $\frac{x_0^2}{a^2} + \frac{y_0^2}{b^2} = 1$ в горизонтальной плоскости Oxy (рис. 20). Изменяя параметр u , получаем семейство прямых.

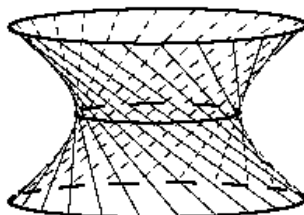


Рис. 20

1.50. Проверьте, что следующие прямые:

$$x = -a \sin u t + a \cos u, y = b \cos u t + b \sin u, z = ct, u \in [0; 2\pi], t \in R$$

образуют второе семейство прямолинейных образующих на однополостном гиперболоиде.

Напишите компьютерную программу построения одного семейства образующих на однополостном гиперболоиде и сравните ее с программой «Образующие на однополостном гиперболоиде».

1.51. На рис. 21 и 22 изображены прямые на гиперболических параболоидах в кубе и пунктиром система координат. Напишите параметрические уравнения гиперболического параболоида и напишите программу построения этих рисунков.

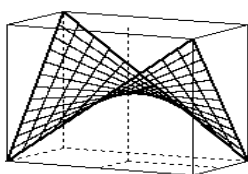


Рис. 21

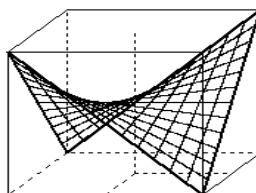


Рис. 22

$$x = a(u - 1/2), y = a(v - 1/2), z = a(2uv - u - v + 1), u \in [0; 1], v \in [0; 1].$$

1.52. На рис. 23 изображен тор, т.е. поверхность, полученная при вращении окружности вокруг прямой, не пересекающей окружность.

В плоскости Oxy рассмотрим окружность $x = r \cos u, y = r \sin u + d, z = 0$, где $d > r$. Применяя поворот $x' = x, y' = y \cos v - z \sin v, z' = y \sin v + z \cos v$ вокруг оси Ox , получим тор.

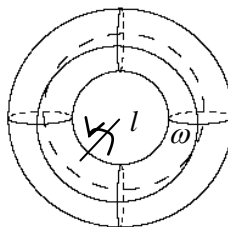


Рис. 23

Напишите программу построения изображения этой поверхности и сравните ее с программой «Тор».

§ 12. Построение диаграммы Вороного

Пусть на плоскости задана система точек $S = \{P_1, P_2, \dots, P_n\}$, причем координаты каждой точки известны, т.е. $P_i(x_i; y_i)$ заданы.

Многоугольником (ячейкой) Вороного точки P_i (многоугольником близости, ячейкой Дирихле, многоугольником Тиссена) называется геометрическое множество точек плоскости, которые находятся к точке P_i ближе, чем к любой другой точке P_j , где $j \neq i$. Многоугольник Вороного для точки P_i обозначим $V(i)$.

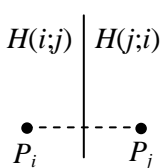


Рис. 1

Для двух фиксированных точек P_i и P_j множество точек, более близких к точке P_i , чем к точке P_j , есть полуплоскость $H(i, j)$, границей которой является серединный перпендикуляр к отрезку $P_i P_j$ (рис. 1).

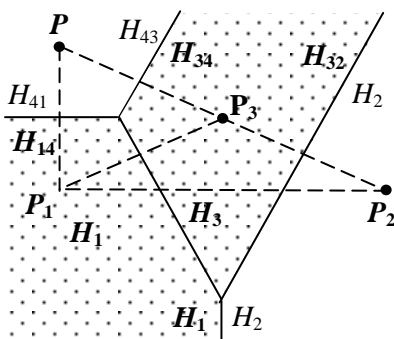


Рис. 2

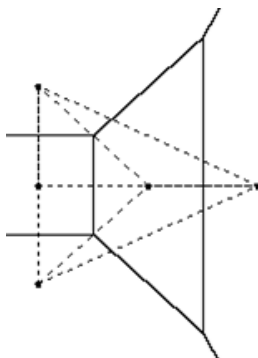


Рис. 3

Для системы точек $S = \{P_1, P_2, \dots, P_n\}$ многоугольник Вороного $V(i)$ является пересечением $(n-1)$ полуплоскостей, т.е. $V(i) = \bigcap_{j \neq i} H(i, j)$. Многоугольник Вороного является выпуклым

многоугольником, имеющим не более $(n-1)$ сторон (рис. 2). Если система $S = \{P_1, P_2\}$ состоит из двух точек, то многоугольниками Вороного являются полуплоскости $H(1,2)$ и $H(2,1)$, т.е. $V(1) = H(1,2)$, $V(2) = H(2,1)$.

Объединяя границы многоугольников, получим сеть, которая называется диаграммой Вороного (рис. 3). Диаграмма названа в честь русского ученого Георгия Феоодосьевича Вороного (1868—1908). Диаграмму Вороного для системы S точек обозначим $V(S)$ или V , если имеется единственная система точек.

Построение диаграммы Вороного имеет важное практическое применение. Если S — система супермаркетов в городе, то ячейки Вороного — области территорий, расположенных ближе к соответствующим магазинам.

Моделирование построения ячеек Вороного можно осуществить по определению, но само построение точек может оказаться медленным процессом при следующей реализации.

Пусть задана область $x_N \leq x \leq x_V$, $y_N \leq y \leq y_V$ для исследования и $M(x,y)$ — произвольная точка из этой области. Для построения ячейки $V(i)$ нужно провести сравнение длин отрезков MP_i и MP_j или определить знак выражения $MP_i^2 - MP_j^2$ для произвольной точки P_j системы, где $j \neq i$.

Пусть точка P_i фиксирована. Организуем в цикле просмотр остальных точек системы S . Если $MP_i^2 - MP_j^2 > 0$ для некоторых значений $j \neq i$, то точка M не принадлежит ячейке $V(i)$. Изменяя значение индекса i , найдем такое его значение, при котором $MP_i^2 - MP_j^2 < 0$ для всех значений $j \neq i$. В этом случае построим точку M на экране компьютера.

Следующая программа «Ячейки Вороного» строит ячейки для системы, состоящей из четырех точек.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
Dim P(4, 2): Dim D(4)
P(1, 1) = 2000: P(1, 2) = 2000: P(2, 1) = 3500: P(2, 2) = 2000
P(3, 1) = 3000: P(3, 2) = 3000: P(4, 1) = 2000: P(4, 2) = 3500
DrawWidth = 4: For i = 1 To 4: PSet (P(i, 1), P(i, 2)): Next i
DrawWidth = 1
```

```

For x = 1000 To 4000
  For y = 1000 To 4000
    For i = 1 To 4
       $D(i) = (x - P(i, 1))^2 + (y - P(i, 2))^2$ 
    Next i
    If D(1) < D(2) And D(1) < D(3) And D(1) < D(4) Then PSet (x, y), QBColor(13)
    If D(2) < D(1) And D(2) < D(3) And D(2) < D(4) Then PSet (x, y), QBColor(11)
    If D(3) < D(1) And D(3) < D(2) And D(3) < D(4) Then PSet (x, y), QBColor(14)
    If D(4) < D(1) And D(4) < D(2) And D(4) < D(3) Then PSet (x, y), QBColor(10)
  Next y, x
  DrawWidth = 4: For i = 1 To 4: PSet (P(i, 1), P(i, 2)): Next i
End Sub

```

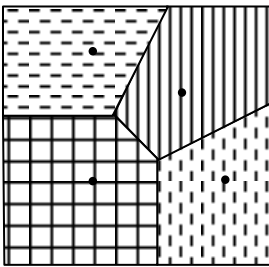


Рис. 4

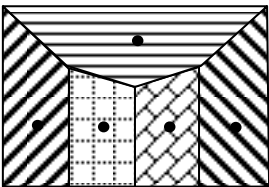


Рис. 5

Замечание. Закраска областей в компьютерной программе осуществляется с помощью различных цветов. Закраска областей на рис. 4 изменена на различную штриховку, чтобы при распечатке журнальной статьи или книги рисунок получался более контрастным.

Эта программа содержит блок условных операторов. При увеличении числа точек в системе S этот блок приходится значительно изменять вручную. Можно автоматизировать изменение условных операторов, если заменить этот блок двумя циклами. В следующей программе «Ячейки Вороного циклом» осуществляется построение областей для системы, содержащей 5 точек (рис. 5).

```

Private Sub Command1_Click()
  Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
  Dim P(6, 2): Dim D(6)
  P(1, 1) = 2000: P(1, 2) = 2000: P(2, 1) = 2500: P(2, 2) = 2000
  P(3, 1) = 3000: P(3, 2) = 2000: P(4, 1) = 3500: P(4, 2) = 2000
  P(5, 1) = 2750: P(5, 2) = 2750: P(6, 1) = P(1, 1): P(6, 2) = P(1, 2)
  DrawWidth = 4: For i = 1 To 5: PSet (P(i, 1), P(i, 2)): Next i
  DrawWidth = 1
  For x = 1750 To 3750
    For y = 1500 To 3000
      For i = 1 To 6
         $D(i) = (x - P(i, 1))^2 + (y - P(i, 2))^2$ 

```



```

Next i
For i = 1 To 5
  For j = 1 To 5
    If j = i Then j = j + 1
    If D(i) > D(j) Then GoTo 1
  Next j
  PSet (x, y), QBColor(i)
1
Next i
Next y, x
DrawWidth = 4: For i = 1 To 5: PSet (P(i, 1), P(i, 2)): Next i
End Sub

```

Для большинства задач не требуется выполнять закраску ячеек Вороного, а нужно указать диаграмму Вороного.

Предыдущие программы легко видоизменить для новой задачи, накладывая проверку условия $MP_i^2 - MP_j^2 = 0$.

Написав правильную программу для построения таким способом точек диаграммы Вороного, можно не обнаружить таких точек на экране компьютера. Причина в том, что точки, удовлетворяющие такому условию, могут иногда оказаться с координатами, являющимися дробными числами или иррациональными числами, а точки на экране компьютера строятся с целочисленными координатами.

Изменяя представленное условие на условие $MP_i^2 - MP_j^2 < \varepsilon$, где ε — некоторая малая константа, можно получить построение диаграммы Вороного. Оно будет приближенным с теоретической точки зрения, но наглядным.

Программа, реализующая рассмотренный способ, также будет работать медленно, т.к. она тестирует все точки с целочисленными координатами из данной области исследования. Причем в программе осуществляется n^2 операций по просмотру точек для проверки условия $MP_i^2 - MP_j^2 < 0$.

Рассмотрим другой способ построения диаграммы Вороного, в котором будет тестироваться значительно меньшее количество точек, т.е. только точки на серединных перпендикулярах.

Составим параметрические уравнения серединного перпендикуляра к отрезку P_iP_j . Для середины H_{ij} отрезка P_iP_j получаем

координаты $H_{ij}\left(\frac{x_i + x_j}{2}; \frac{y_i + y_j}{2}\right)$.

Вектор $\overrightarrow{p}(y_i - y_j, x_j - x_i)$ перпендикулярен вектору $\overrightarrow{P_iP_j}(x_j - x_i; y_j - y_i)$, т.к. $\overrightarrow{p} \cdot \overrightarrow{P_iP_j} = 0$, поэтому вектор \overrightarrow{p} является направляющим вектором для серединного перпендикуляра.

Параметрические уравнения серединного перпендикуляра:

$$x = \frac{x_i + x_j}{2} + (y_i - y_j)t, y = \frac{y_i + y_j}{2} + (x_j - x_i)t. \quad (1)$$

Программа «**Диаграмма Вороного**» строит диаграмму для четырех точек (рис. 6—13).

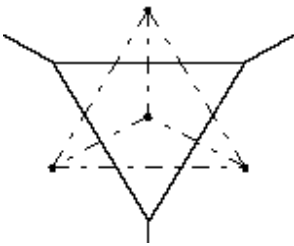


Рис. 6

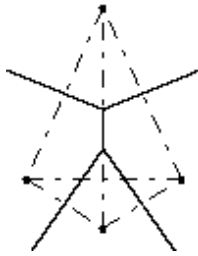


Рис. 7

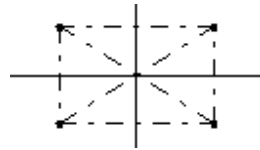


Рис. 8

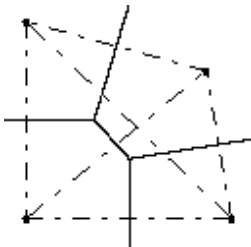


Рис. 9

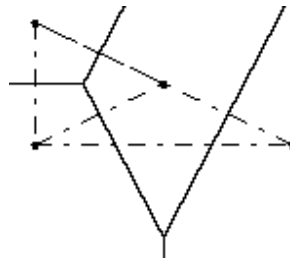


Рис. 10

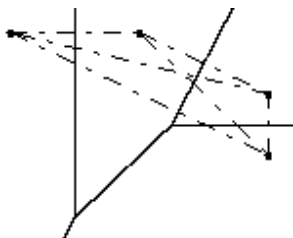


Рис. 11

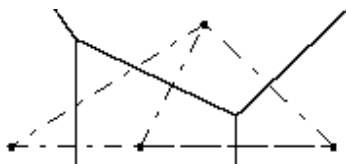


Рис. 12

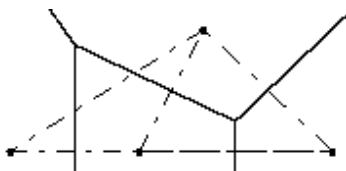


Рис. 13

Две диаграммы назовем топологически эквивалентными, если одну из диаграмм можно преобразовать в другую диаграмму непрерывным преобразованием. Под непрерывным преобразованием будем понимать растяжение или сжатие отрезков, поворот отрезков или лучей, при которых не происходит склеивания или разрыва. Очевидно, что число отрезков и лучей, выходящих из точки, является топологическим инвариантом. Такая величина является степенью вершины (узла) диаграммы.

Замкнутый контур после непрерывного преобразования должен остаться замкнутым контуром.

1.53. На рис. 6—13 выделите топологически эквивалентные диаграммы. Можно ли к ним добавить диаграмму, топологически отличную от представленных диаграмм. Постройте на компьютере такую диаграмму.

На рис. 6—13 программа построила все отрезки, соединяющие точки данной системы. В некоторых случаях эти отрезки имеют общие точки только в точках данной системы S . В этом случае будем говорить, что получен двойственный планарный граф к диаграмме Вороного. Для системы точек выполнена триангуляция, т.е. разбиение выпуклой оболочки данной системы точек на треугольники. Соединяя отрезками те точки, ячейки Вороного

которых соприкасаются хотя бы углами, получаем триангуляцию Делоне.

В каких из рассмотренных случаев на рисунке представлена триангуляция с помощью пунктирных линий?

Для четырех точек системы существует 6 серединных перпендикуляров. Для произвольной точки на каждом серединном перпендикуляре проверяется условие $MP_i^2 - MP_j^2 < 0$, поэтому программа содержит 6 циклов по проверке необходимого условия. При увеличении числа точек системы S в эту программу необходимо вносить существенные изменения. Аналогично предыдущим программам осуществим проверку этих условий с помощью вложенных циклов.

1.54. На рис. 14—26 с помощью программы «Многоугольник близости» построены диаграммы Вороного для различных систем из пяти точек. Определите среди них топологически эквивалентные диаграммы. В каких случаях пунктирные рисунки изображают триангуляцию выпуклой оболочки данной системы точек?

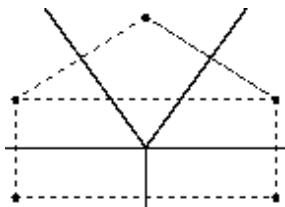


Рис. 14

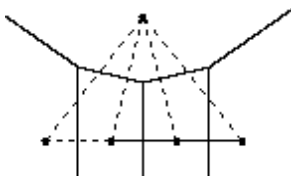


Рис. 15

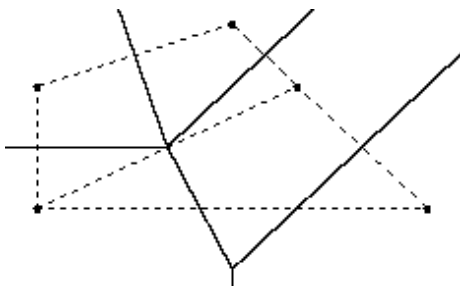


Рис. 16

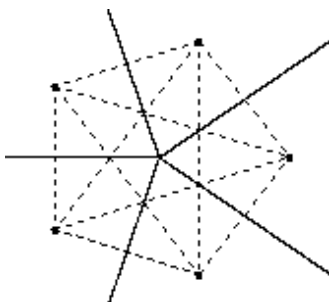


Рис. 17

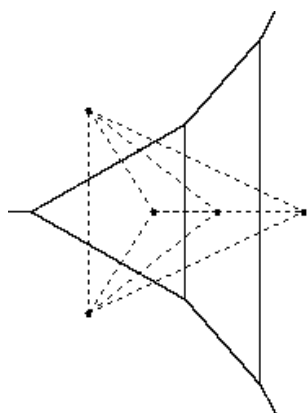


Рис. 18

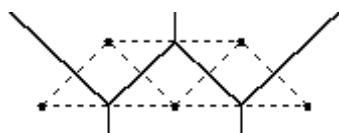


Рис. 19

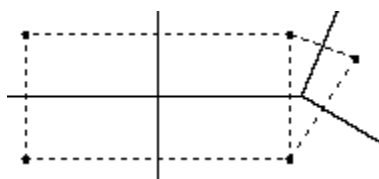


Рис. 20

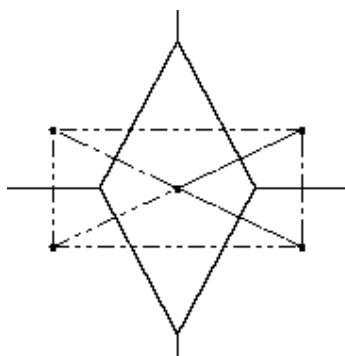


Рис. 21

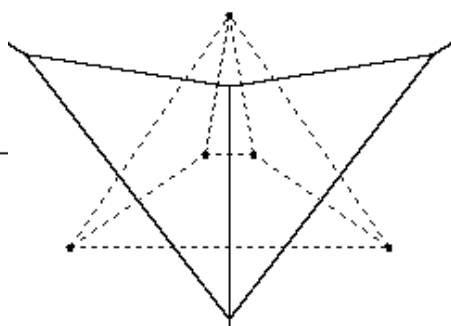


Рис. 22

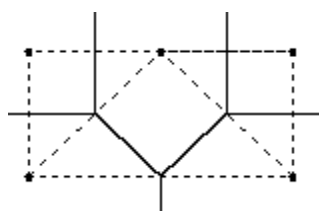


Рис. 23

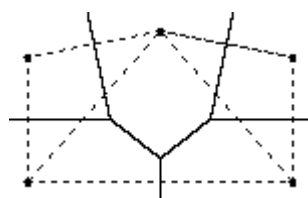


Рис. 24

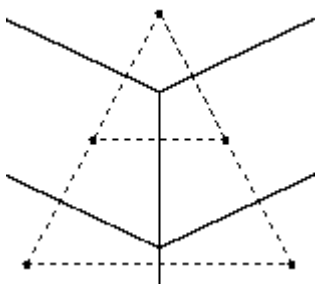


Рис. 25

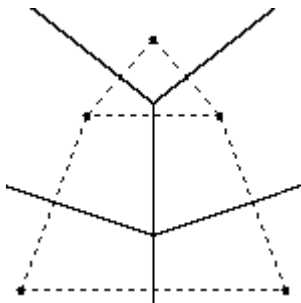


Рис. 26

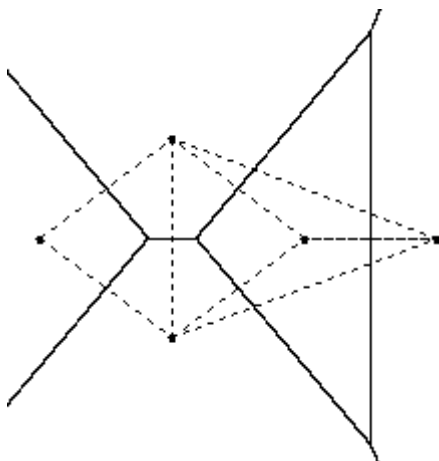


Рис. 27

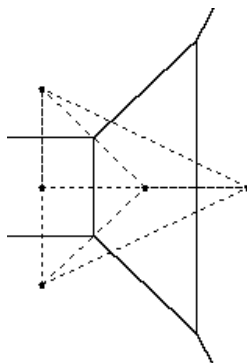


Рис. 28

1.55. Все ли топологически различные случаи диаграммы Вороного представлены на рис. 14—28 для системы из пяти точек?

1.56. Для системы из n точек всего существует C_n^2 отрезков, соединяющих эти точки. На рис. 6—13 построены все такие отрезки. На рис. 14—28 построены в основном те отрезки, для которых существуют линии раздела ячеек Вороного. На каких из рисунков 14—28 построены только отрезки, соответствующие ребрам диаграммы Вороного?

1.57. В [14, с. 257] доказано, что диаграмма Вороного множества из n точек имеет не более $2n - 5$ вершин и $3n - 6$ ребер. Подсчитайте на каждом из рис. 14—28 число вершин и число

ребер диаграмм Вороного и сравните полученные числа с верхними оценками $2n-5, 3n-6$ для $n=5$.

1.58. Пусть n_V — произвольное натуральное число, удовлетворяющее условию $1 \leq n_V \leq 5$. Существует ли диаграмма Вороного для системы из пяти точек, имеющая n_V вершин?

1.59. Пусть m_V — произвольное натуральное число, удовлетворяющее условию $5 \leq m_V \leq 9$. Существует ли диаграмма Вороного для системы из пяти точек, имеющая m_V ребер?

1.60. Постройте несколько различных диаграмм Вороного для системы из шести точек. Познакомьтесь с другими методами разработки алгоритмов Вороного (алгоритм Форчуна, рекурсивный алгоритм) в пособии [16] и в Интернете для большого числа точек.

Для системы, состоящей из трех неколлинеарных точек, вершина диаграммы Вороного может быть быстро построена, т.к. вершина является центром описанной окружности.

Рассмотрим уравнение окружности

$$(x - x_0)^2 + (y - y_0)^2 = R^2.$$

Точки $P_1(x_1; y_1), P_2(x_2; y_2), P_3(x_3; y_3)$ принадлежат окружности, поэтому получаем равенства

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = R^2, \quad (2)$$

$$(x_2 - x_0)^2 + (y_2 - y_0)^2 = R^2, \quad (3)$$

$$(x_3 - x_0)^2 + (y_3 - y_0)^2 = R^2. \quad (4)$$

Вычитая из равенств (3) и (4) равенство (2), получим

$$2(x_2 - x_1)x_0 + 2(y_2 - y_1)y_0 = x_2^2 - x_1^2 + y_2^2 - y_1^2, \quad (5)$$

$$2(x_3 - x_1)x_0 + 2(y_3 - y_1)y_0 = x_3^2 - x_1^2 + y_3^2 - y_1^2. \quad (6)$$

Равенства (5) и (6) в действительности являются уравнениями серединных перпендикуляров P_1P_2, P_1P_3 относительно переменных x_0, y_0 .

Решая систему из уравнений (5) и (6), получим

$$x_0 = \frac{\begin{vmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 & y_2 - y_1 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 & y_3 - y_1 \end{vmatrix}}{2 \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}}, \quad (7)$$

$$y_0 = \frac{\begin{vmatrix} x_2 - x_1 & x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ x_3 - x_1 & x_3^2 - x_1^2 + y_3^2 - y_1^2 \end{vmatrix}}{2 \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}}. \quad (8)$$

Координаты x_0, y_0 определяются однозначно, если $\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} \neq 0$. Это условие означает, что три данные точки не расположены на одной прямой.

Выражение

$$2 \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}$$

имеет простой геометрический смысл — это учетверенная площадь треугольника $P_1P_2P_3$.

Формулы (7) и (8) можно записать в другом виде:

$$x_0 = \frac{(x_1^2 + y_1^2)(y_2 - y_3) + (x_2^2 + y_2^2)(y_3 - y_1) + (x_3^2 + y_3^2)(y_1 - y_2)}{2[(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)]}, \quad (9)$$

$$y_0 = -\frac{(x_1^2 + y_1^2)(x_2 - x_3) + (x_2^2 + y_2^2)(x_3 - x_1) + (x_3^2 + y_3^2)(x_1 - x_2)}{2[(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)]}. \quad (10)$$

После определения координат x_0, y_0 радиус окружности определяется из равенства (2).

Рассмотрим второй способ получения координат центра описанной окружности.

Для серединного перпендикуляра P_1P_2 имеем параметрические уравнения:

$$x = \frac{x_2 + x_1}{2} + (y_2 - y_1)t, y = \frac{y_2 + y_1}{2} - (x_2 - x_1)t . \quad (11)$$

Для серединного перпендикуляра P_1P_3 имеем параметрические уравнения:

$$x = \frac{x_3 + x_1}{2} + (y_3 - y_1)u, y = \frac{y_3 + y_1}{2} - (x_3 - x_1)u . \quad (12)$$

Найдем значение параметра для точки пересечения серединных перпендикуляров из системы, образованной уравнениями (11) и (12):

$$t = \frac{(x_3 - x_2)(x_3 - x_1) + (y_3 - y_2)(y_3 - y_1)}{2[(y_2 - y_1)(x_3 - x_1) - (x_2 - x_1)(y_3 - y_1)]} . \quad (13)$$

Подставляя значение параметра t в уравнения (11), получим координаты центра описанной окружности.

Замечание. В пособии [16] уравнение окружности, проходящей через три точки $P_1(x_1; y_1), P_2(x_2; y_2), P_3(x_3; y_3)$, рассматривается в виде:

$$\begin{vmatrix} x^2 + y^2 & x & y & 1 \\ x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ x_2^2 + y_2^2 & x_2 & y_2 & 1 \\ x_3^2 + y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0$$

или $a(x^2 + y^2) - bx + cy - d = 0$, где

$$a = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, b = \begin{vmatrix} x_1^2 + y_1^2 & y_1 & 1 \\ x_2^2 + y_2^2 & y_2 & 1 \\ x_3^2 + y_3^2 & y_3 & 1 \end{vmatrix}, \quad (14)$$

$$c = \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix}, d = \begin{vmatrix} x_1^2 + y_1^2 & x_1 & y_1 \\ x_2^2 + y_2^2 & x_2 & y_2 \\ x_3^2 + y_3^2 & x_3 & y_3 \end{vmatrix}. \quad (15)$$

Координаты центра и радиус окружности вычисляются по формулам:

$$x_0 = \frac{b}{2a}, y_0 = -\frac{c}{2a}, R = \frac{b^2 + c^2 - 4ad}{4a^2}. \quad (16)$$

Для определения координат центра и радиуса можно использовать группы формул (7), (8), (2) или (13), (11), (2) или (14), (15), (16).

Сравните число необходимых операций в каждом случае и выберите наиболее рациональный способ.

1.61. Напишите программу построения диаграммы Вороного для системы из трех точек, используя формулы (7), (8) или (13), (11). Сравните ее с программой «**Диаграмма Вороного 3 точки**».

В этой программе реализуется построение диаграммы Вороного тремя способами. Программа определяет время, затраченное для построения диаграммы каждым способом, и печатает полученные значения.

Определение параметров окружности, проходящей через три данные точки, имеет большое значение для металлообрабатывающей и деревообрабатывающей промышленности.

При обработке деталей типа «тело вращения» необходимо контролировать параметры круглости и соосности, радиального и торцевого биений. Компьютеризованный контроль точности обработки деталей должен определять меру отклонения точек контура деталей от заданной формы.

Иногда применяют следующий алгоритм определения параметров окружности, проходящей через три данные точки $P_1(x_1; y_1), P_2(x_2; y_2), P_3(x_3; y_3)$:

1) длины сторон треугольника

$$a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad b = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2},$$

$$c = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2};$$

2) площадь треугольника

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \text{ где } p = (a+b+c)/2;$$

3) радиус описанной окружности

$$R = \frac{abc}{4S};$$

4) координаты центра окружности

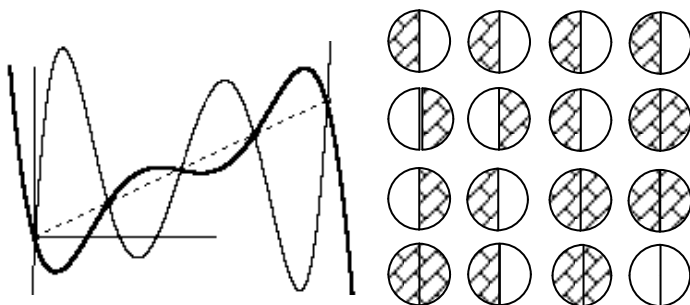
$$x_0 = \frac{1}{4S} \begin{vmatrix} x_1^2 + y_1^2 & y_1 & 1 \\ x_2^2 + y_2^2 & y_2 & 1 \\ x_3^2 + y_3^2 & y_3 & 1 \end{vmatrix}, y_0 = \frac{1}{4S} \begin{vmatrix} x_1^2 + y_1^2 & x_1 & 1 \\ x_2^2 + y_2^2 & x_2 & 1 \\ x_3^2 + y_3^2 & x_3 & 1 \end{vmatrix}.$$

Значения этих определителей можно вычислять по формулам (9), (10).

Сравните этот метод с приведенными выше методами определения параметров окружности, проходящей через три точки.

Глава 2

ПОИСК ФУНКЦИОНАЛЬНОЙ ЗАВИСИМОСТИ



Что нас ожидает в этой главе?

1. Построение линий третьего порядка комбинацией линий второго и первого порядков.
2. Метод нахождения явного задания функции, заданной рекуррентным отношением.
3. Восстановление графика многочлена Лагранжа и многочлена Ньютона по заданному множеству точек.
4. Поиск булевой функции от трех переменных, удовлетворяющих матричному тесту.

§ 13. Линии уровня функции

Пусть в прямоугольной декартовой системе координат линия γ_1 задана общим уравнением $F_1(x, y) = 0$, а линия γ_2 задана общим уравнением $F_2(x, y) = 0$. Объединение $\gamma_1 \cup \gamma_2$ этих линий состоит из точек, каждая из которых принадлежит хотя бы одной из линий. Объединение $\gamma_1 \cup \gamma_2$ определяется уравнением $F_1(x, y) \cdot F_2(x, y) = 0$.

Рассмотрим семейство линий γ_C , заданное уравнением $F_1(x, y) \cdot F_2(x, y) = C$, где $C = \text{const}$ и $C \in \mathbb{R}$.

Линия $\gamma_1 \cup \gamma_2$ оказалась включенной в семейство линий при значении параметра $C = 0$.

Семейство линий $F_1(x, y) \cdot F_2(x, y) = C$ называется линиями уровня для функции $z = F_1(x, y) \cdot F_2(x, y)$, т.к. при фиксированном значении C линия семейства является множеством точек плоскости Oxy , в которых функция принимает постоянное значение $z = C$.

Линии уровня имеют применение в военной топографии при изображении возвышенностей или впадин на географической карте, в метеорологии (изображение пунктов на линии с одинаковой температурой, например, изотермы — линии, соединяющие точки с равными температурами), а также в экономике.

Пример 1. Рассмотрим горизонтальную прямую γ_1 , заданную уравнением $y = 0$, и прямую γ_2 , заданную уравнением $y = \sqrt{3}x$. Прямая γ_2 образует с прямой γ_1 угол, равный 60° .

Объединение этих кривых определяется уравнением $y(\sqrt{3}x - y) = 0$.

На рис. 1 построено семейство линий $y(\sqrt{3}x - y) = C$ с помощью компьютерной программы. Если точка $M(x; y)$ принадлежит одной из линий семейства, то точка $M(-x; -y)$ также принадлежит этой линии, поэтому при построении точек линий семейства использована симметрия относительно начала координат.

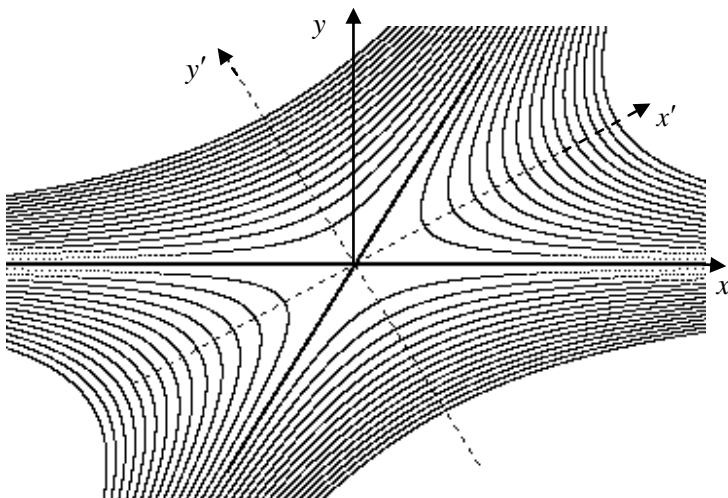


Рис. 1

Программа «Линии уровня из гипербол».

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
x0 = 4000: y0 = 4000: b = 2000
For c = 5 To 3 * b ^ 2 / 4 Step b ^ 2 / 20
For Y = 1 To b
X = (c + Y ^ 2) / (Y * Sqr(3))
PSet (X + x0, Y + y0): PSet (-X + x0, -Y + y0)
Next Y, c
For c = -3 * b ^ 2 / 4 To 5 Step b ^ 2 / 20
For Y = 1 To b
X = (c + Y ^ 2) / (Y * Sqr(3))
PSet (X + x0, Y + y0): PSet (-X + x0, -Y + y0)
Next Y, c
DrawStyle = 2
Line (-b * Sqr(3) / 2 + x0, -b / 2 + y0)-(b * Sqr(3) / 2 + x0, b / 2 + y0)
Line (-b / 2 + x0, b * Sqr(3) / 2 + y0)-(b / 2 + x0, -b * Sqr(3) / 2 + y0)
DrawWidth = 2
Line (-2 * b + x0, y0)-(3 * b + x0, y0)
Line (-b / 2 + x0, -b * Sqr(3) / 2 + y0)-(b / 2 + x0, b * Sqr(3) / 2 + y0)
End Sub
```

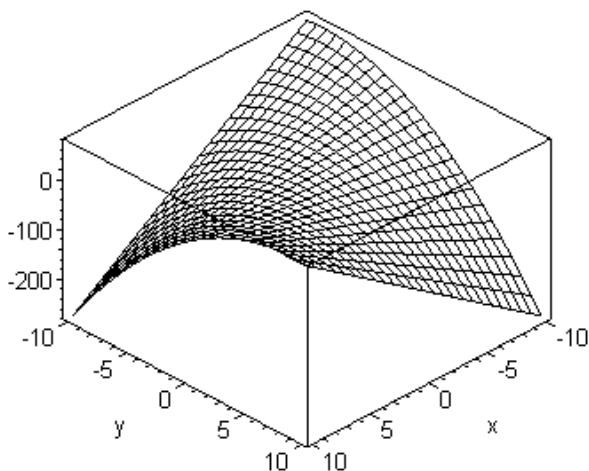



Рис. 2

На рис. 2 в математическом пакете MAPLE построена поверхность, заданная уравнением $z = y(\sqrt{3x} - y)$.

Программа построения графика функции:

```
> plot3d(y*(sqrt(3)*x-y),x=-10..10,y=-10..10).
```

Биссектрисы углов между прямыми (рис. 1) являются осями симметрии для построенных прямых, при условии, что линии уровня можно продолжить. По внешнему виду эти линии похожи на гиперболы. Докажем это, переходя к новой системе координат, оси которой совпадают с биссектрисами углов.

Рассмотрим поворот системы координат на угол 30° .

Формулы связи между координатами точки в различных системах координат:

$$\begin{cases} x = \cos \varphi x' - \sin \varphi y', \\ y = \sin \varphi x' + \cos \varphi y', \end{cases} \text{ или } \begin{cases} x = \sqrt{3}/2 x' - 1/2 y', \\ y = 1/2 x' + \sqrt{3}/2 y'. \end{cases}$$

Подставляя переменные x, y в уравнение семейства линий уровня, получим

$$x'^2 - 3y'^2 = 2C.$$

При $C > 0$ получаем гиперболу с действительной полуосью $\sqrt{2C}$ и мнимой полуосью $\sqrt{2/3 C}$:

$$\frac{x'^2}{\sqrt{2C}^2} - \frac{y'^2}{\sqrt{2/3C}^2} = 1.$$

При $C < 0$ получаем гиперболу с действительной полуосью $\sqrt{-2/3C}$ и мнимой полуосью $\sqrt{-2C}$:

$$-\frac{x'^2}{\sqrt{-2C}^2} + \frac{y'^2}{\sqrt{-2/3C}^2} = 1.$$

Замечание. Выражение $y(\sqrt{3}x - y)$ имеет следующую геометрическую интерпретацию.

Расстояние от точки $M_0(x_0; y_0)$ до прямой γ , заданной уравнением $ax + by + c = 0$, вычисляется по формуле:

$$\rho(M_0, \gamma) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$

Расстояние от произвольной точки $M(x; y)$ до прямой γ_1 , заданной уравнением $y = 0$, равно $\rho(M, \gamma_1) = |y|$.

Расстояние от произвольной точки $M(x; y)$ до прямой γ_2 , заданной уравнением $\sqrt{3}x - y = 0$, равно:

$$\rho(M, \gamma_2) = \frac{|\sqrt{3}x - y|}{2}.$$

Произведение расстояний равно:

$$\rho(M, \gamma_1) \cdot \rho(M, \gamma_2) = \frac{|y(\sqrt{3}x - y)|}{2}$$

или $y(\sqrt{3}x - y) = \pm 2\rho(M, \gamma_1) \cdot \rho(M, \gamma_2)$.

Следовательно, равенство $y(\sqrt{3}x - y) = C$ означает, что определяется множество точек плоскости, произведение расстояний от которых до двух данных прямых равно постоянной величине. Как было показано выше, это множество является гиперболой при $C \neq 0$ или парой данных прямых при $C = 0$ (рис. 3). \square

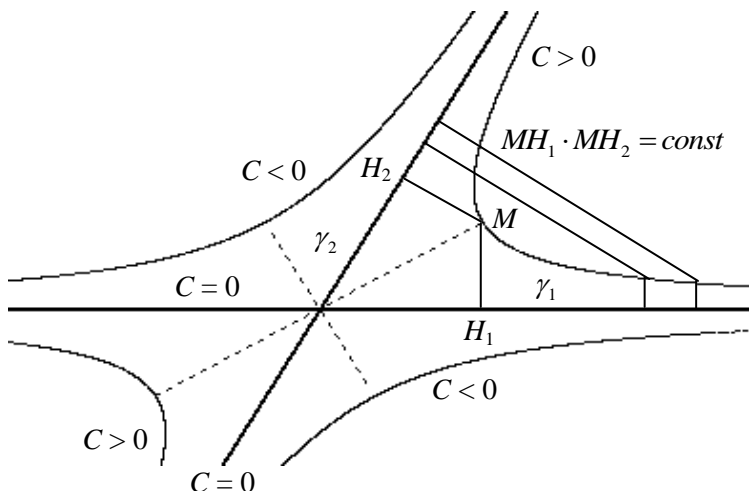


Рис. 3

Пример 2. Рассмотрим прямую γ_1 , заданную уравнением $x=0$, и окружность γ_2 , заданную уравнением $(x-a)^2 + y^2 = r^2$. Объединение этих линий $\gamma_1 \cup \gamma_2$ состоит из прямой и окружности и задается уравнением $x[(x-a)^2 + y^2 - r^2] = 0$.

Включим это объединение в семейство линий:

$$x[(x-a)^2 + y^2 - r^2] = C, \text{ где } C \in \mathbb{R}.$$

Эти линии являются линиями уровня функции

$$z = x[(x-a)^2 + y^2 - r^2].$$

Рассмотрим геометрическую интерпретацию выражения $x[(x-a)^2 + y^2 - r^2]$.

Для произвольной точки $M(x; y)$ и окружности $(x-a)^2 + y^2 = r^2$ величина $(x-a)^2 + y^2 - r^2$ называется степенью точки относительно окружности и обозначается Ст. M .

Пусть произвольная точка $M(x; y)$ расположена вне окружности γ_2 и из точки M проведены касательные MT_1, MT_2 к окружности, тогда $(x-a)^2 + y^2 - r^2 = AM^2 - AT^2 = MT_1 \cdot MT_2$.

Расстояние от точки $M(x; y)$ до прямой $x = 0$ равно $\rho(M, \gamma_1) = |x|$, поэтому $x = \pm \rho(M, \gamma_1)$. Величина x равна расстоянию от точки $M(x; y)$ до прямой, взятому с определенным знаком в зависимости от того, в какой полуплоскости относительно прямой γ_1 расположена эта плоскость.

Расстояние от точки до прямой, взятое с определенным знаком, называется *отклонением* этой точки от прямой. Обозначим его $\delta(M, \gamma_1)$ (рис. 4).

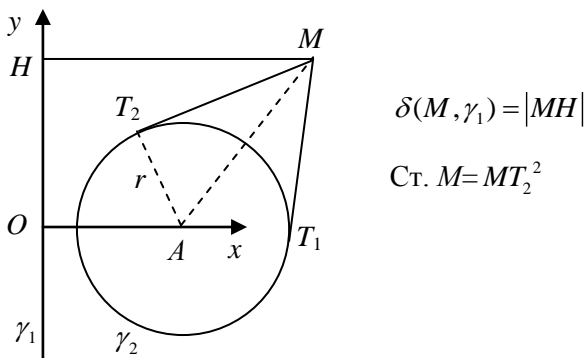
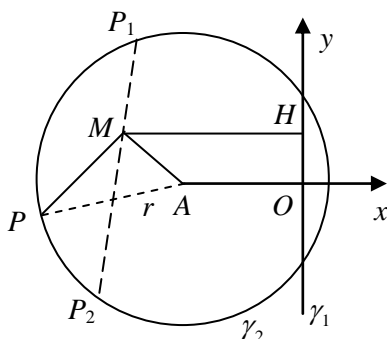


Рис. 4

Для точки M , расположенной вне окружности и в полуплоскости $x > 0$, получаем $x[(x-a)^2 + y^2 - r^2] = \delta(M, \gamma_1) \cdot MT^2$.

Пусть произвольная точка $M(x; y)$ расположена внутри окружности γ_2 и через точку M проведена прямая, перпендикулярная AM и пересекающая окружность в точке P (рис. 5), тогда степень точки равна $(x-a)^2 + y^2 - r^2 = AM^2 - AP^2 = -MP^2$.

Для точки M , расположенной внутри окружности и в полуплоскости $x < 0$, получаем $\rho(M, \gamma_1) = |x| = -x$, $\delta(M, \gamma_1) = x$
 $x[(x-a)^2 + y^2 - r^2] = -\delta(M, \gamma_1) \cdot MP^2$.



$$\text{Ст. } M = -MP^2$$

$$\text{Ст. } M = MP_1 \cdot MP_2$$

$$\delta(M, \gamma_1) = -|MH|$$

Рис. 5

Если точка M расположена вне окружности, то степень точки M относительно окружности равна квадрату длины касательной, проведенной из этой точки к окружности.

Если точка M расположена внутри окружности, то степень точки равна произведению отрезков на хорде, проходящей через точку M , взятому с отрицательным знаком.

Если точка M расположена на окружности, то степень точки равна нулю.

Во всех случаях

$$x[(x-a)^2 + y^2 - r^2] = \pm \delta(M, \gamma_1) \cdot \text{Ст. } M.$$

Точки линии, заданной уравнением $x[(x-a)^2 + y^2 - r^2] = C$, удовлетворяют условию: произведение отклонения точки до данной прямой на степень этой точки относительно данной окружности является постоянной величиной.

Семейство линий уровня представлено на рис. 6.

Семейство линий симметрично относительно оси Ox , т.к. переменная y входит в уравнение семейства в четной степени.

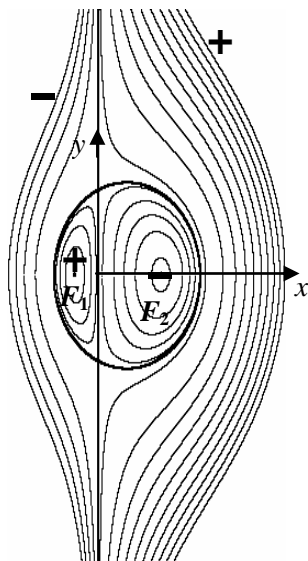


Рис. 6

Функция $z = x[(x-a)^2 + y^2 - r^2]$ на оси Ox принимает значения $f(x) = x[(x-a)^2 - r^2]$ или $f(x) = x^3 - 2ax^2 + (a-r^2)x$.

Определим точки экстремума данной функции:

$$f'(x) = 3x^2 - 4ax + a - r^2.$$

Из уравнения $3x^2 - 4ax + a - r^2 = 0$ определяем критические точки:

$$x_1 = \frac{2a - \sqrt{a^2 + 3r^2}}{3}, x_2 = \frac{2a + \sqrt{a^2 + 3r^2}}{3}.$$

В точке x_1 функция имеет локальный максимум, а в точке x_2 локальный минимум, причем в точках x_i экстремума

$$f(x_i) = x_i^3 - 2ax_i^2 + (a-r^2)x_i = x_i(x_i^2 - 2ax_i + a - r^2),$$

$$f(x_i) = x_i(3x_i^2 - 4ax_i + a - r^2 - 2x_i^2 + 2ax_i).$$

Учитывая равенство $3x_i^2 - 4ax_i + a - r^2 = 0$ в точках экстремума, получаем

$$f(x_i) = 2x_i^2(a - x_i).$$

Для точки $F_1(x_1; 0)$ константа C в уравнении $x[(x-a)^2 + y^2 - r^2] = C$ принимает значение $C_1 = 2x_1^2(a - x_1)$, а для точки $F_2(x_2; 0)$ константа C принимает значение $C_2 = 2x_2^2(a - x_1)$.

Значения констант можно вычислять и непосредственно по формуле определения, т.е. $C_i = x_i[(x_i - a)^2 - r^2]$.

На рис. 6 символами $+$ и $-$ обозначены области, в которых постоянная C принимает положительное или отрицательное значения. Данная прямая γ_1 и данная окружность γ_2 изображены жирными линиями.

Если обратиться к справочнику по кривым третьего порядка [18, с. 7, 20], то, согласно классификации И.Ньютона, семейство этих кривых называется монодиаметральной дефективной гиперболой.

Для рис. 6, на котором окружность и прямая пересекаются, получаем несколько типов линий третьего порядка.

При $f(x_2) < C < 0$ первый тип линий состоит из двух линий. Одна из них расположена в полуплоскости $x < 0$ и асимптотически приближается к прямой $x = 0$. Вторая линия является замкнутой линией, содержащей внутри точку F_2 .

Если C стремится к нулю, то компонента в полуплоскости $x < 0$ приближается к прямой $x = 0$, а компонента в полуплоскости $x > 0$ приближается к окружности γ_2 .

Если C приближается к $f(x_2)$, то замкнутая компонента сжимается к точке F_2 . При $C = f(x_2)$ получаем второй тип кривой третьего порядка, состоящий из бесконечной линии в полуплоскости $x < 0$ и одной точки F_2 . \square

2.1. Напишите программу построения рис. 7 и сравните ее с программой «Поле потенциала».

Замечание. При определении диапазона изменения переменной x необходимо определить область допустимых значений переменной в кубическом уравнении $x^3 - 2ax^2 + (a - r^2)x = C$. Чтобы избежать предварительного решения кубического уравнения, в программе для переменной x указан более широкий диапазон. В результате программа тестирует значения переменной, на которых функция $y(x)$ не определена и строит лишние точки. Для устранения лишних точек в некоторых циклах этой программы наложено условие вида `If y > 1 Then PSet ...`

В каждом из циклов, содержащих такой условный оператор, сузите область изменения переменной x и покажите, что в этом случае можно построить точки без условного оператора.

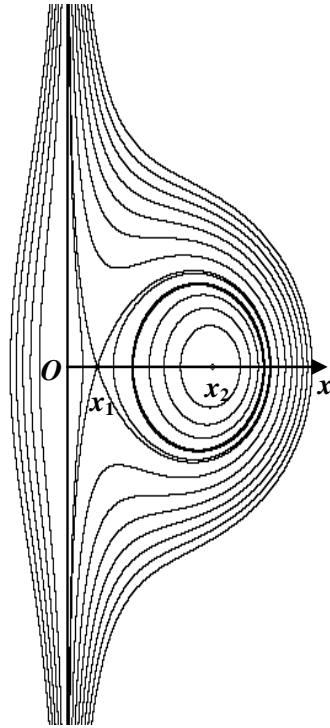


Рис. 7

Напишите подпрограмму решения кубического уравнения и перед началом цикла по переменной x вставьте обращение к этой подпрограмме.

2.2. Постройте линии уровня для функции $z = x^2 + y^2$.

2.3. Какая зависимость между линиями уровня функции $z = f(x, y)$ и линиями уровня функции $z = -f(x, y)$?

2.4. На рис. 1 симметрия построенных линий уровня относительно биссектрис углов становится очевидной, если линии продолжить или обрезать рисунок ножницами по углам. Исправьте создавшуюся ситуацию, т.е. постройте с помощью программы семейство гипербол, представленное на рис. 8. Сравните с программой «Семейство симметричных гипербол».

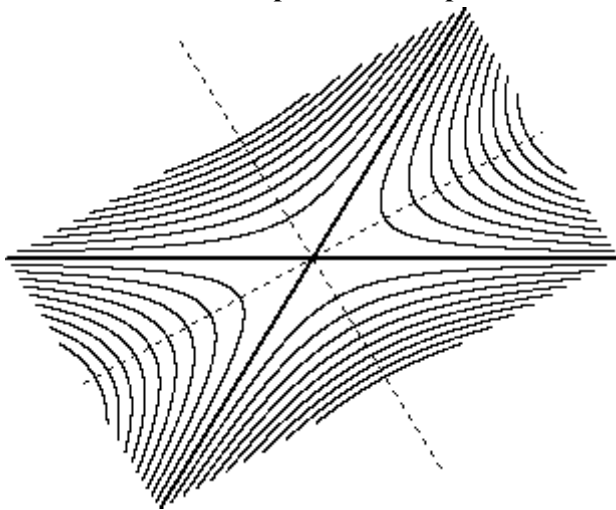


Рис. 8

2.5. На рис. 9 построено два семейства отрезков. Каждое семейство отрезков лежит на пучке прямых, пересекающихся вне данного отрезка. В результате наложения двух семейств иногда получается иллюзия построения новой фигуры, т.е. «муаровый эффект». Напишите программу построения такого рисунка.

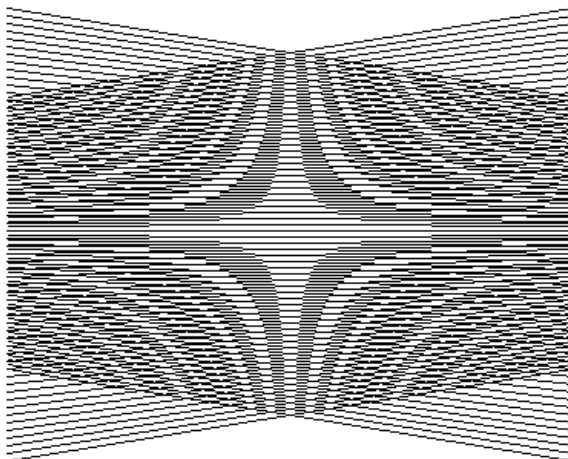


Рис. 9

2.6. На рис. 7 одна из линий получена при $C = 2x_1^2(a - x_1)$ и состоит из одной ветви с петлей. Постройте эту линию при других значения параметров $a, r, a > r$.

2.7. Охарактеризуйте тип линии при значении параметра $C \in (0; f(x_1))$ и $a > r$.

2.8. Для прямой γ_1 , заданной уравнением $x = 0$, и окружности γ_2 , заданной уравнением $(x - r)^2 + y^2 = r^2$, найдите объединение этих линий $\gamma_1 \cup \gamma_2$, включите это множество в семейство линий уровня и постройте на компьютере семейство линий уровня. Охарактеризуйте типы линий третьего порядка в этом случае.

§ 14. Производящая функция

В компьютерных вычислениях значительное число вычислений осуществляется по заданному рекуррентному отношению. Для выяснения асимптотики вычисляемой величины лучше использовать аналитическое задание этой величины как функции натурального аргумента. Этот раздел посвящен методам получения явного задания функции от натурального аргумента, если для нее известно рекуррентное отношение.

$u_{n+1} = F(n, u_n, u_{n+1}, \dots, u_{n+k-1})$ — рекуррентное отношение, которое позволяет вычислять любой член последовательности, если заданы ее k предшествующих членов.

Пример 1. $u_{n+1} = u_n q$, где $q = \text{const} \neq 0$. Рекуррентное отношение определяет геометрическую прогрессию. \square

Пример 2. $u_{n+1} = u_n + d$, где $d = \text{const}$. Рекуррентное отношение определяет арифметическую прогрессию. \square

Пример 3. $F_n = F_{n-1} + F_{n-2}$, $n > 2$, $F_1 = 1$, $F_2 = 1$ — последовательность чисел Фибоначчи, т.е. последовательность чисел

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots$$

Любое натуральное число n имеет единственное представление

$$n = F_{k_1} + F_{k_2} + \dots + F_{k_r}, \quad (1)$$

$$k_1 \geq k_2 + 2, k_2 \geq k_3 + 2, \dots, k_{r-1} \geq k_r + 2,$$

где представление осуществляется с помощью жадного подхода. В качестве F_{k_1} выбирается наибольшее число Фибоначчи, которое меньше либо равно данному числу n , затем в качестве F_{k_2} выбирается наибольшее число, которое меньше либо равно $n - F_{k_1}$ и т.д.

Равенство (1) представим в виде:

$$n = \sum_{k=2}^m b_k F_k, \quad (2)$$

где $b_k = 1$, если число F_k имеется в разложении, и $b_k = 0$, если соответственное число Фибоначчи отсутствует в разложении.

Сопоставим числу (2) машинное слово (b_m, \dots, b_3, b_2) . Получим код числа n в фибоначчевой системе счисления, причем для записи используются только символы 0 или 1.

Числа $F_2, F_3, F_4 \dots$ образуют базис системы счисления Фибоначчи, а равенство (2) является разложением числа n по базису.

Примеры кодов некоторых чисел:

$1_{10} = 1 =$	00000001,	$7_{10} = 5 + 2 =$	00001010,
$2_{10} = 2 =$	00000010,	$8_{10} = 8 =$	00010000,
$3_{10} = 3 =$	00000100,	$9_{10} = 8 + 1 =$	00010001,
$4_{10} = 3 + 1 =$	00000101,	$19_{10} = 13 + 5 + 1 =$	00101001,

$$\begin{array}{llll} 5_{10}=5= & 00001000, & 45_{10}=34+8+3= & 10010100, \\ 6_{10}=5+1= & 00001001, & 54_{10}=34+13+5+2= & 10101010. \end{array}$$

В этой форме представления чисел две единицы не могут быть расположены рядом. Это свойство используется при кодировании информации. \square

Если u_{n+k} линейно выражается через $u_n, u_{n+1}, \dots, u_{n+k-1}$, т.е. $u_{n+k} = c_n u_n + c_{n+1} u_{n+1} + \dots + c_{n+k-1} u_{n+k-1}$, где $c_i - \text{const}$, то рекуррентная последовательность называется *возвратной последовательностью*.

Пусть для возвратной последовательности

$$u_n = au_{n-1} + bu_{n-2}, \quad (3)$$

где $n > 2, a, b - \text{const}$ известны первые два члена последовательности u_1, u_2 .

При $b = 0$ получаем геометрическую прогрессию, для которой $u_n = a^{n-1} u_1$.

Характеристическое уравнение для формулы (3):

$$k^2 - ak - b = 0. \quad (4)$$

Пусть $b \neq 0$ и k_1, k_2 — корни соответствующего характеристического уравнения.

Из рекуррентного соотношения (2) получаем

$$u_n - (k_1 + k_2) u_{n-1} = -k_1 k_2 u_{n-2}$$

или

$$u_n - k_1 u_{n-1} = k_2 (u_{n-1} - k_1 u_{n-2}), \quad (5)$$

$$u_n - k_2 u_{n-1} = k_1 (u_{n-1} - k_2 u_{n-2}). \quad (6)$$

Пусть $k_1 \neq k_2$. Обозначим $v_n = u_n - k_1 u_{n-1}$, $w_n = u_n - k_2 u_{n-1}$, тогда $\{v_n\}, \{w_n\}$ — геометрические последовательности, т.к. $v_n = k_2 v_{n-1}$, $w_n = k_1 w_{n-1}$.

Следовательно, $v_n = k_2^{n-2} v_2$, $w_n = k_1^{n-2} w_2$ или

$$u_n - k_1 u_{n-1} = k_2^{n-2} (u_2 - k_1 u_1), \quad (7)$$

$$u_n - k_2 u_{n-1} = k_1^{n-2} (u_2 - k_2 u_1). \quad (8)$$

Умножая уравнение (7) на $(-k_2)$, а уравнение (8) на k_1 и складывая, получим

$$u_n = \frac{k_1^{n-1}(u_2 - k_2 u_1) - k_2^{n-1}(u_2 - k_1 u_1)}{k_1 - k_2}, \quad (9)$$

$$u_n = \frac{u_2 - k_2 u_1}{k_1(k_1 - k_2)} k_1^n - \frac{u_2 - k_1 u_1}{k_2(k_1 - k_2)} k_2^n.$$

Получаем формулу для общего члена:

$$u_n = C_1 k_1^n + C_2 k_2^n, \quad (10)$$

$$\text{где } C_1 = \frac{u_2 - k_2 u_1}{k_1(k_1 - k_2)}, C_2 = -\frac{u_2 - k_1 u_1}{k_2(k_1 - k_2)}. \quad (11)$$

По формуле (9) или по формулам (10), (11) можно вычислять все члены последовательности.

На практике постоянные C_1, C_2 находят из условий

$$\begin{cases} u_1 = C_1 \alpha + C_2 \beta \\ u_2 = C_1 \alpha^2 + C_2 \beta^2 \end{cases}.$$

Пример 4. Найдем формулу для числа Фибоначчи F_n .

Для рекуррентного отношения $F_n = F_{n-1} + F_{n-2}$ составим характеристическое уравнение $k^2 - k - 1 = 0$.

$$\text{Корни уравнения } k_1 = \frac{1 - \sqrt{5}}{2}, k_2 = \frac{1 + \sqrt{5}}{2}.$$

Для постоянных C_1, C_2 получаем систему:

$$1 = C_1 \frac{1 - \sqrt{5}}{2} + C_2 \frac{1 + \sqrt{5}}{2}, 1 = C_1 \left(\frac{1 - \sqrt{5}}{2} \right)^2 + C_2 \left(\frac{1 + \sqrt{5}}{2} \right)^2.$$

$$\text{Отсюда следует } C_1 = -\frac{1}{\sqrt{5}}, C_2 = \frac{1}{\sqrt{5}}.$$

Формула n -го члена числа Фибоначчи принимает вид:

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right].$$

Замечание. Если для последовательности известны первые два члена u_0, u_1 , то система для определения постоянных C_1, C_2 принимает более простой вид:

$$\begin{cases} u_0 = C_1 + C_2 \\ u_1 = C_1\alpha + C_2\beta \end{cases}.$$

Например, для чисел Фибоначчи можно расширить определение, полагая $F_0 = 0$, тогда константы определяются из следующей системы:

$$\begin{cases} 0 = C_1 + C_2, \\ 1 = C_1 \frac{1-\sqrt{5}}{2} + C_2 \frac{1+\sqrt{5}}{2}. \quad \square \end{cases}$$

Продолжим изучение множества решений рекуррентного соотношения (2) для случая, когда уравнение (2) имеет корень второй кратности, т.е. при $k_1 = k_2 = k$. Уравнения (4) и (5) примут вид:

$$u_n - ku_{n-1} = k(u_{n-1} - ku_{n-2}).$$

Уравнение (6) преобразуется в уравнение

$$u_n - ku_{n-1} = k^{n-2} (u_2 - ku_1).$$

Пусть $u_2 - ku_1 = v_2$, тогда получаем последовательные рекурсии:

$$u_n = ku_{n-1} + k^{n-2}v_2,$$

$$u_{n-1} = ku_{n-2} + k^{n-3}v_2,$$

$$u_{n-2} = ku_{n-3} + k^{n-4}v_2,$$

.....

$$u_3 = ku_2 + kv_2,$$

$$u_2 = ku_1 + v_2.$$

Производя подстановки, получим

$$u_n = k^{n-1}u_1 + (n-1)k^{n-2}v_2 \quad (12)$$

или

$$u_n = k^n \frac{u_1}{k} + (n-1)k^n \frac{v_2}{k^2} = k^n \left(\frac{u_1}{k} - \frac{v_2}{k^2} \right) + nk^n \frac{v_2}{k^2}.$$

Получаем другую форму общего члена последовательности:

$$u_n = C_1 k^n + C_2 n k^n, \quad (13)$$

$$\text{где } C_1 = \frac{u_1}{k} - \frac{v_2}{k^2}, \quad C_2 = \frac{v_2}{k^2}. \quad (14)$$

По формуле (12) или по формулам (13), (14) можно вычислять все члены последовательности.

Постоянные C_1, C_2 находят из условий:

$$\begin{cases} u_0 = C_1 \\ u_1 = C_1 \alpha + C_2 \beta \end{cases}$$

Выражение вида $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n + \dots$ называется *формальным числовым рядом*.

Формальные числовые ряды можно складывать, вычитать, умножать, делить, дифференцировать, составлять их композицию, не беспокоясь о сходимости рядов.

Пусть $\{a_n\} = a_0, a_1, \dots$ — произвольная числовая последовательность, тогда формальный числовой ряд $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n + \dots$ называется *производящей функцией* этой последовательности.

Разложение функций в формальные степенные ряды имеет вид:

$$\frac{1}{1-ax} = 1 + ax + a^2 x^2 + \dots + a^n x^n + \dots \quad (15)$$

При $|ax| < 1$ равенство (6) является формулой суммы геометрической прогрессии.

Если a — фиксированное значение параметра, то равенство (14) выполняется при $x \in (-1/|a|; 1/|a|)$. На множестве $(-1/|a|; 1/|a|)$ равенство (6) выполняется, и этого множества достаточно, чтобы использовать это разложение.

Умножая равенство (15) на x и дифференцируя равенство $\frac{x}{1-ax} = x + ax^2 + a^2 x^3 + \dots + a^n x^{n+1} + \dots$, получим равенство

$$\frac{1}{(1-ax)^2} = 1 + 2ax + 3a^2x^2 + \dots + (n+1)a^n x^n + \dots \quad (16)$$

Перемножая равенства (15) и (16), получим

$$\frac{1}{(1-ax)^3} = 1 + 3ax + 6a^2x^2 + \dots + \frac{(n+1)(n+2)}{2} a^n x^n + \dots \quad (17)$$

Аналогично

$$\frac{1}{(1-ax)^k} = 1 + C_k^1 ax + C_{k+1}^2 a^2 x^2 + \dots + C_{n+k-1}^n a^n x^n + \dots \quad (18)$$

Метод производящих функций — это переход от оперирования с комбинаторными объектами и числовыми последовательностями к действиям с формальными степенными рядами.

Пример 5. Найти формулу общего члена последовательности, заданной рекуррентным отношением $a_n = 3a_{n-1}$ для $n \geq 1$.

Решение. Пусть $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$, тогда

$$3xf(x) = 3a_0x + 3a_1x^2 + 3a_2x^3 + \dots + 3a_{n-1}x^n + 3a_nx^{n+1} + \dots,$$

$$f(x) - 3xf(x) = a_0 + (a_1 - 3a_0)x + (a_2 - 3a_1)x^2 + \dots + (a_n - 3a_{n-1})x^n + \dots$$

Но $a_n - 3a_{n-1} = 0$ при всех $n \geq 1$, поэтому

$$(1-3x)f(x) = a_0, \quad f(x) = \frac{a_0}{1-3x} = a_0 \frac{1}{1-3x},$$

$$f(x) = a_0(1 + 3x + 3^2x^2 + \dots + 3^nx^n + \dots).$$

Сравнивая коэффициенты при x^n в двух разложениях функции $f(x)$, получаем $a_n = a_0 3^n$.

Замечание. В условии данной задачи рассматривается рекуррентное соотношение для геометрической прогрессии со знаменателем $q = 3$. Формула общего члена получается по известной формуле с поправкой на то, что прогрессия имеет первый член a_0 . \square

Пример 6. Найти формулу общего члена последовательности, заданной рекуррентным отношением $a_n = a_{n-1} + 3$ для $n \geq 1$ и $a_0 = 5$.

Решение. Пусть $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$, тогда $xf(x) = a_0x + a_1x^2 + a_2x^3 + \dots + a_{n-1}x^n + a_nx^{n+1} + \dots$.

При сравнении коэффициентов потребуется производящая функция, у которой все коэффициенты содержат число 3, поэтому рассмотрим разложение:

$$\frac{3}{1-ax} = 3(1 + ax + a^2x^2 + \dots + a^nx^n + \dots),$$

$$\begin{aligned} f(x) - xf(x) - \frac{3}{1-x} &= \\ &= a_0 - 3 + (a_1 - a_0 - 3)x + (a_2 - a_1 - 3)x^2 + \dots + (a_n - a_{n-1} - 3)x^n + \dots \end{aligned}$$

Но $a_n - a_{n-1} - 3 = 0$ при всех $n \geq 1, a_0 = 5$, поэтому

$$f(x) - xf(x) - \frac{3}{1-x} = 2, \quad (1-x)f(x) = 2 + \frac{3}{1-x},$$

$$f(x) = \frac{2}{1-x} + \frac{3}{(1-x)^2},$$

$$f(x) = 2(1 + x + x^2 + \dots + x^n + \dots) + 3(1 + 2x + 3x^2 + \dots + (n+1)x^n + \dots)$$

$$f(x) = 5 + 8x + \dots + (3n+5)x^n + \dots$$

Сравнивая коэффициенты при x^n в двух разложениях функции $f(x)$, получаем $a_n = 3n + 5$.

Замечание. В условии данной задачи рассматривается рекуррентное соотношение для арифметической прогрессии с разностью $d = 3$. Формула общего члена арифметической прогрессии получается по известной формуле с поправкой на то, что прогрессия имеет первый член a_0 . \square

Пример 7. Найти формулу общего члена последовательности, заданной рекуррентным отношением $a_n = 3a_{n-1} + 4^n$ для $n \geq 1$ и $a_0 = 1$.

Решение. Пусть $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$, тогда

$$3xf(x) = 3a_0x + 3a_1x^2 + 3a_2x^3 + \dots + 3a_{n-1}x^n + 3a_nx^{n+1} + \dots$$

$$\frac{1}{1-4x} = 1 + 4x + 4^2x^2 + \dots + 4^nx^n + \dots$$

$$f(x) - 3x f(x) - \frac{1}{1-4x} =$$

$$a_0 - 1 + (a_1 - 3a_0 - 4)x + (a_2 - 3a_1 - 4^2)x^2 + \dots + (a_n - 3a_{n-1} - 4^n)x^n + \dots$$

$$f(x) - 3x f(x) - \frac{1}{1-4x} = 0, \quad f(x) = \frac{1}{(1-4x)(1-3x)}.$$

Представим функцию $\frac{1}{(1-4x)(1-3x)}$ в виде суммы элементарных дробей

$$\frac{1}{(1-4x)(1-3x)} = \frac{A}{1-4x} + \frac{B}{1-3x}.$$

Умножая обе части на $(1-4x)(1-3x)$, найдем

$$1 = A(1-3x) + B(1-4x).$$

Подставляя в равенство $x = 1/3$, получаем $B = 3$.

Подставляя в равенство $x = 1/4$, получаем $A = 4$.

Следовательно,

$$f(x) = \frac{4}{1-4x} - \frac{3}{1-3x},$$

$$f(x) = 4(1 + 4x + 4^2 x^2 + \dots + 4^n x^n + \dots) - 3(1 + 3x + 3^2 x^2 + \dots + 3^n x^n + \dots).$$

Сравнивая коэффициенты при x^n в двух разложениях функции $f(x)$, получаем

$$a_n = 4 \cdot 4^n - 3 \cdot 3^n, \quad a_n = 4^{n+1} - 3^{n+1}.$$

Рассмотрим проверку полученного решения.

При $n = 0$ получаем $a_0 = 4^1 - 3^1 = 1$.

Проверим рекуррентное соотношение

$$3a_{n-1} + 4^n = 3(4^n - 3^n) + 4^n = 4^{n+1} - 3^{n+1} = a_n. \quad \square$$

Пример 8. Найти формулу общего члена последовательности, заданной рекуррентным отношением $a_n = 2a_{n-1} + n$ для $n \geq 1$ и $a_0 = 3$.

Пусть $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n + \dots$, тогда

$$2x f(x) = 2a_0 x + 2a_1 x^2 + 2a_2 x^3 + \dots + 2a_n x^{n+1} + \dots,$$

$$\frac{x}{(1-x)^2} = x(1+2x+3x^2+\dots+(n+1)x^n+\dots),$$

$$f(x) - 2x f(x) - \frac{x}{(1-x)^2} =$$

$$= a_0 + (a_1 - 2a_0 - 1)x + (a_2 - 2a_1 - 2)x^2 + \dots + (a_n - 2a_{n-1} - n)x^n + \dots$$

$$(1-2x)f(x) = \frac{x}{(1-x)^2} + 3, \quad f(x) = \frac{x}{(1-2x)(1-x)^2} + \frac{3}{1-2x}.$$

$$f(x) = \frac{5}{1-2x} - \frac{1}{1-x} - \frac{1}{(1-x)^2},$$

$$f(x) = 5(1+2x+2^2x^2+\dots+2^nx^n+\dots) -$$

$$-(1+x+x^2+\dots+x^n+\dots) - (1+2x+3x^2+\dots+(n+1)x^n+\dots).$$

$$a_n = 5 \cdot 2^n - n - 2. \quad \square$$

К возвратным последовательностям можно применять как метод, определенный формулами (4) и (5) с последующим определением констант, так и метод производящей функции.

2.9. Найдите общий член последовательности, заданный рекуррентным отношением:

а) $u_n = 6u_{n-1} - 9u_{n-2}, \quad u_1 = 2, \quad u_2 = 4,$

б) $u_n = -2u_{n-1} - u_{n-2}, \quad u_1 = -1, \quad u_2 = -1,$

в) $u_n = 5u_{n-1} - 6u_{n-2}, \quad u_0 = 0, \quad u_1 = 1,$

г) $u_n = 3u_{n-1} - 2u_{n-2}, \quad u_0 = 1, \quad u_1 = 1,$

д) $u_n = 2u_{n-1} - u_{n-2}, \quad u_0 = 1, \quad u_1 = 2.$

§ 15. Интерполяционные многочлены

При исследовании различных процессов функциональная зависимость между переменной x и функцией y в большинстве случаев неизвестна. В результате эксперимента иногда появляются данные, например, в виде таблицы 1.

Таблица 1

переменная x	x_0	x_1	x_2	x_3	\dots	x_n
функция y	y_0	y_1	y_2	y_3	\dots	y_n

Если изобразить эти данные на координатной плоскости в виде точек $(x_0; y_0), (x_1; y_1), (x_2; y_2), \dots, (x_n; y_n)$, то иногда нам кажется, что функциональная зависимость явно просматривается.

Если для функции все значения переменной перечислены и для нее не требуется искать значения этой же функции при других значениях переменной, то функция задана полностью, как графически, так и таблицей. Конечно, будет лучше, если вместо этих наборов будет найдено аналитическое задание зависимости, например, одной формулой.

К примеру, для данных, представленных в таблице 2, функциональную зависимость можно выразить формулой $y = x/2$, но существуют и другие, более сложные зависимости, удовлетворяющие этим данным.

Таблица 2

x	0	100	200	300	400
y	0	50	100	150	200

На рис. 1 через пять данных точек проходит прямая, заданная уравнением $y = x/2$, а также линия, заданная многочленом пятой степени.

В большинстве случаев аналитическое задание функциональной зависимости сложно определить, и после анализа данных в таблице требуется распространить функциональную зависимость на другие значения переменной, т.е. расширить область определения функции — от n значений перейти к интервалу или отрезку, содержащему эти значения аргумента.

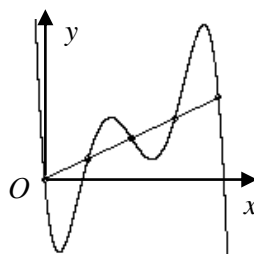


Рис. 1

Необходимость поиска аналитического задания функциональной зависимости объясняется также тем фактом, что при автоматизации вычислений необходимо результат также выводить в виде массива, соответствующего введенному массиву данных величин.

Пусть для n -угольника вводится количество сторон и координаты вершин в виде массива. Требуется вычислить длины сторон и записать значения в виде массива.

Традиция обозначения сторон треугольника представлена на рис. 2 и заключается в том, что для вершины соответствующая сторона расположена напротив этой вершины.

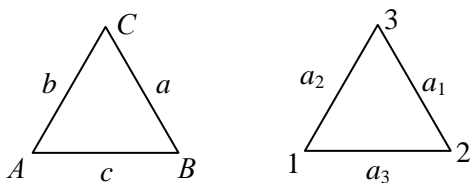


Рис. 2

Представим, что вычисление длин сторон осуществляется в цикле, который последовательно рассматривает пары вершин (1,2), (2,3),(3,1). После каждого вычисления длины стороны значение этой величины нужно обозначить a_3, a_1, a_2 .

Если обход вершин осуществляется в порядке 1, 2, 3, то зависимость номера стороны от номера вершины, рассматриваемого в цикле, представлена в таблице 3:

Таблица 3

n	1	2	3
$f(n)$	3	1	2

Будем искать зависимость в виде многочлена. Три значения аргумента подсказывают нам, что зависимость нужно искать в виде многочлена второй степени с тремя неизвестными коэффициентами:

$$f(n) = c_2x^2 + c_1x + c_0.$$

Подставляя значения переменной и значения функции, получим систему:

$$\begin{cases} c_2 + c_1 + c_0 = 3, \\ 4c_2 + 2c_1 + c_0 = 1, \\ 9c_2 + 9c_1 + c_0 = 2. \end{cases}$$

Решение системы: $c_2 = \frac{3}{2}, c_1 = -\frac{13}{2}, c_0 = 8.$

Функциональная зависимость имеет вид:

$$f(n) = \frac{3}{2}n^2 - \frac{13}{2}n + 8.$$

Заметим, что визуальное наблюдение закономерности обозначения вершин и сторон позволяет нам моментально совершить обозначение сторон для любого второго треугольника, изображенного в любом месте на плоскости. Аналитическая зависимость менее естественна и воспроизвести ее через неделю, вероятно, будет сложнее.

Для четырехугольника понятие «сторона, расположенная напротив вершины» не имеет смысла.

Продолжим исследование поиска зависимости, переходя к пятиугольнику (рис. 3).

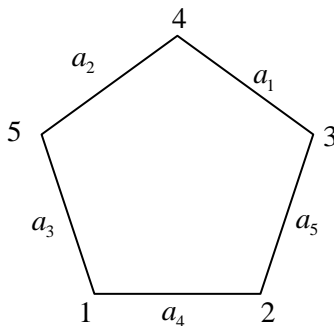


Рис. 3

Таблица 4

n	1	2	3	4	5
$f(n)$	4	5	1	2	3

В следующей таблице 4 представлена зависимость номера стороны, прилежащей к соответствующей вершине:

Первый способ — искать зависимость в виде многочлена четвертой степени:

$$f(n) = c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0.$$

Для этого придется решать систему из пяти уравнений с пятью неизвестными.

А нет ли способа менее трудоемкого? Поиск зависимости по таблице — это типичная ситуация, и для ее решения, вероятно, должны существовать алгоритмы.

Таблица 5

переменная x	x_0	x_1	x_2	x_3	\dots	x_n
функция y	y_0	y_1	y_2	y_3	\dots	y_n

Для функции, заданной таблицей 5, зависимость находится с помощью интерполяционного многочлена Лагранжа по формуле:

$$L_n(x) = \sum_{k=0}^n y_k \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}.$$

Подстановкой легко проверяется, что многочлен Лагранжа удовлетворяет условиям $L_n(x_k) = y_k$ для $k = 0, 1, 2, \dots, n$.

Рассмотрим вычисления по этой формуле для $n = 2$, т.е. применим формулу для поиска зависимости в треугольнике:

$$L_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

Подставляя значения, получим

$$\begin{aligned} L_2(x) &= 3 \frac{(x-2)(x-3)}{(1-2)(1-3)} + 1 \frac{(x-1)(x-3)}{(2-1)(2-3)} + 2 \frac{(x-1)(x-2)}{(3-1)(3-2)} = \\ &= \frac{3}{2}(x^2 - 5x + 6) - (x - 4x + 3) + (x^2 - 3x + 2). \end{aligned}$$

$$L_2(x) = \frac{3}{2}x^2 - \frac{13}{2}x + 8.$$

Для $n = 4$ получаем

$$\begin{aligned} L_4(x) &= 4 \frac{(x-2)(x-3)(x-4)(x-5)}{(1-2)(1-3)(1-4)(1-5)} + 5 \frac{(x-1)(x-3)(x-4)(x-5)}{(2-1)(2-3)(2-4)(2-5)} + \\ &+ 1 \frac{(x-1)(x-2)(x-4)(x-5)}{(3-1)(3-2)(3-4)(3-5)} + 2 \frac{(x-1)(x-2)(x-3)(x-5)}{(4-1)(4-2)(4-3)(4-5)} + \\ &+ 3 \frac{(x-1)(x-2)(x-3)(x-4)}{(5-1)(5-2)(5-3)(5-4)}, \\ L_4(x) &= \frac{1}{24}(-15x^4 + 190x^3 - 825x^2 + 1394x - 648). \end{aligned}$$

Формула Лагранжа интересна тем, что она предоставляет готовый рецепт, по которому можно найти нужную формулу, но математические вычисления по ней являются трудоемким процессом.

Вычисления можно значительно упростить с помощью программы, написанной на языке QBASIC:

```

INPUT "N="; N
DIM X(N), F(N)
FOR I = 0 TO N
PRINT "[X("; I; "),F("; I; : INPUT ")]="; X(I), F(I)
NEXT
INPUT "argument X="; X
FOR j = 0 TO N - 1
FOR I = j + 1 TO N
F(I) = ((X - X(j)) * F(I) - (X - X(I)) * F(j)) / (X(I) - X(j))
NEXT I, j
F1 = F(N)
PRINT "L"; N; "("; X; ")="; F1
END

```

Математические вычисления можно упростить, если использовать другой метод.

Для функции, заданной таблицей, интерполяционный многочлен будем искать в виде:

$P_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \dots + b_{n-1}(x - x_1)(x - x_2) \dots (x - x_{n-1})$ с неизвестными пока коэффициентами. Найдем коэффициенты, последовательно полагая $P(x_0) = y_0, P(x_1) = y_1, \dots, P(x_n) = y_n$.

Для $n = 2$ формула имеет вид:

$$P_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1).$$

Подставляя значения, получим

при $x = 1 \rightarrow 3 = b_0$,

при $x = 2 \rightarrow 1 = 3 + b_1 \cdot 1, \quad b_1 = -2$,

при $x = 3 \rightarrow 2 = 3 - 2 \cdot 2 + b_2 \cdot 1 \cdot 2, \quad b_2 = \frac{3}{2}$.

Итак, $P_2(x) = 3 - 2(x - 1) + \frac{3}{2}(x - 1)(x - 2) = \frac{3}{2}x^2 - \frac{13}{2}x + 8$.

Для $n = 4$ формула имеет вид:

$$P_4(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + b_3(x - x_0)(x - x_1)(x - x_2) + b_4(x - x_0)(x - x_1)(x - x_2)(x - x_3).$$

Подставляя, получим

при $x = 1 \rightarrow 4 = b_0$,

при $x = 2 \rightarrow 5 = 4 + b_1 \cdot 1, \quad b_1 = 1$,

$$\text{при } x = 3 \rightarrow 1 = 4 + 1 \cdot 2 + b_2 \cdot 1 \cdot 2, \quad b_2 = -\frac{5}{2},$$

$$\text{при } x = 4 \rightarrow 2 = 4 + 1 \cdot 3 - \frac{5}{2} \cdot 3 \cdot 2 + b_3 \cdot 3 \cdot 2 \cdot 1, \quad b_3 = \frac{5}{3},$$

$$\text{при } x = 5 \rightarrow 3 = 4 + 1 \cdot 4 - \frac{5}{2} \cdot 4 \cdot 3 + \frac{5}{3} \cdot 4 \cdot 3 \cdot 2 + b_4 \cdot 4 \cdot 3 \cdot 2 \cdot 1, \quad b_4 = -\frac{15}{24}.$$

$$\text{Итак, } P_4(x) = 4 + (x-1) - \frac{5}{2}(x-1)(x-2) + \frac{5}{3}(x-1)(x-2)(x-3) + \\ + \left(-\frac{15}{24}\right)(x-1)(x-2)(x-3)(x-4).$$

$$P_4(x) = \frac{1}{24}(-15x^4 + 190x^3 - 825x^2 + 1394x - 648).$$

Последний способ называется способом интерполяции Ньютона.

Иногда и этот способ можно упростить. Пусть числа $x_1, x_1, x_2, \dots, x_n$ упорядочены в порядке возрастания и находятся на равном расстоянии, т.е. $x_{i+1} - x_i = h = \text{const}$. Величина h называется шагом таблицы. Величины $\Delta y_i = \Delta f_i = f(x_{i+1}) - f(x_i)$ называются конечными разностями первого порядка.

Из разностей первого порядка составим снова конечные разности, которые называются разностями второго порядка:

$$\Delta^2 y_i = \Delta^2 f_i = \Delta f(x_{i+1}) - \Delta f(x_i).$$

Аналогично продолжим

$$\Delta^n y_i = \Delta^n f_i = \Delta^{n-1} f(x_{i+1}) - \Delta^{n-1} f(x_i).$$

Таблица 6

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	
x_2	y_2	Δy_2	$\Delta^2 y_2$		
x_3	y_3	Δy_3			
x_4	y_4				

Конечные разности различных порядков оформим в таблице 6.

Обозначим $\frac{x - x_1}{h} = q$, тогда интерполяционная формула Ньютона имеет вид:

$$P_{n-1}(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!}\Delta^2 y_0 + \dots + \frac{q(q-1)\dots(q-n+1)}{n!}\Delta^{n-1} y_0.$$

Рассмотрим обсуждаемые выше примеры.

При $n = 3$ составим таблицу разностей 7:

Таблица 7

x	y	Δy	$\Delta^2 y$
1	3	-2	3
2	1	1	
3	2		

В нашем случае $h = 1$, $q = x - 1$. По интерполяционной формуле Ньютона получаем

$$P_2(x) = 3 - (x-1) \cdot 2 + \frac{(x-1)(x-2)}{2!} \cdot 3 \text{ или}$$

$$P_2(x) = \frac{3}{2}x^2 - \frac{13}{2}x + 8.$$

Таблица 8

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	4	1	-5	10	-15
2	5	-4	5	-5	
3	1	1	0		
4	2	1			
5	3				

При $n = 5$ составим таблицу разностей 8.

По интерполяционной формуле Ньютона получаем

$$P_4(x) = 4 + (x-1) \cdot 1 + \frac{(x-1)(x-2)}{2!} \cdot (-5) + \frac{(x-1)(x-2)(x-3)}{3!} \cdot 10 + \\ + \frac{(x-1)(x-2)(x-3)(x-4)}{4!} \cdot (-15),$$

$$P_4(x) = \frac{1}{24}(-15x^4 + 190x^3 - 825x^2 + 1394x - 648).$$

Вычисления можно значительно упростить с помощью программы, написанной на языке QBASIC:

```
INPUT "N="; N
FOR I = 0 TO N
PRINT "x("; I; "); Y("; I; : INPUT ")="; X(I), Y(I): NEXT
INPUT "argument X="; X
L = Y(0): S = 1
FOR I = N TO 1 STEP -1
I1 = N - I
FOR k = 0 TO I - 1
Y(k) = (Y(k + 1) - Y(k)) / (X(k + 1 + I1) - X(k))
NEXT
S = S * (X - X(I1)): L1 = L: L = L + Y(0) * S
NEXT I
PRINT "L("; X; ")="; L
END
```

Напишите аналогичную программу на языке Visual Basic для построения графика многочлена Лагранжа и сравните ее с программой **«График многочлена Лагранжа»**.

Итак, мы познакомились с четырьмя способами поиска многочлена наименьшей степени, график которого проходит через заданную систему точек:

- решение системы линейных алгебраических уравнений;
- определение интерполяционного многочлена Лагранжа;
- определение интерполяционного многочлена Ньютона методом неопределенных коэффициентов;
- определение интерполяционного многочлена Ньютона с помощью конечных разностей.

Есть ли необходимость в нахождении такой сложной зависимости для данной геометрической задачи?

Для произвольного нечетного числа n геометрически легко найти зависимость (рис. 4), используя кусочно-линейную зависимость:

$$f(k) = \begin{cases} k + \frac{n+1}{2}, & \text{если } 0 < k \leq \frac{n-1}{2}, \\ k - \frac{n-1}{2}, & \text{если } \frac{n+1}{2} \leq k \leq n, \end{cases}$$

где k — номер вершины, $f(k)$ — номер стороны, следующей за этой вершиной.

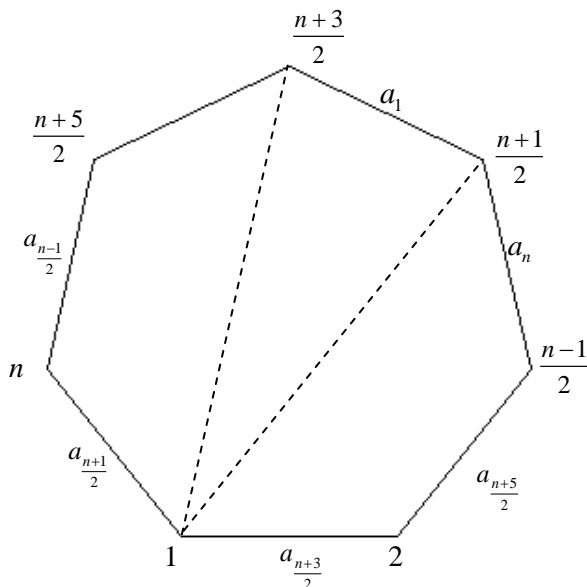


Рис. 4

Читатель, наверное, устал от этих обозначений и поиска зависимости. В действительности традиция обозначать сторону многоугольника символом, связанным с противоположащей стороной, соблюдается только для треугольника. Для $n \geq 4$ нумерация стороны связана с номером предшествующей или последующей вершины следующим образом (рис. 5).

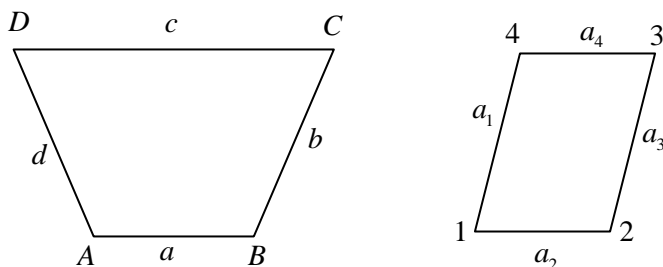


Рис. 5

Интерполяционные многочлены находят в основном в тех случаях, когда функцию, заданную для нескольких значений переменной, нужно распространить на большее число переменных или на некоторый отрезок.

2.10. Найдите интерполяционный многочлен Лагранжа для функций, заданных следующими данными. Постройте графики полученных функций на некотором отрезке, содержащем данные значения переменной x_i . Отметьте на этом графике точки, координаты которых заданы.

а)

x_i	-3	-1	2
y_i	-5	-11	10

б)

x_i	1	2	3
y_i	-3	-2	-1

в)

x_i	-1	1	2	3
y_i	-11	-3	1	13

2.11. Найдите интерполяционный многочлен Ньютона для функций, заданных следующими данными. Постройте графики полученных функций на некотором отрезке, содержащем данные значения переменной x_i . Отметьте на этом графике точки, координаты которых заданы.

а)

x_i	1	2	3
y_i	4	9	18

б)

x_i	1	3	5
y_i	-4	-10	-24

в)

x_i	-1	0	1	2
y_i	-8	-1	0	1

§ 16. Булевы функции от трех переменных, удовлетворяющие условию коммутативности теста

Булевы функции от двух переменных, удовлетворяющие условию коммутативности теста

$$f(f(A, B), f(C, D)) = f(f(A, C), f(B, D)), \quad (1)$$

изучались в пособиях [19, 20]. Они имеют важное значение при разработке матричных тестов по развитию и оценке логического мышления. В настоящей статье рассматривается обобщение этого свойства для функций от трех переменных. Рассмотрим матричный тест, сформированный в виде матрицы A размером 4×4 . В левом верхнем углу расположена матрица аргументов A , размером 3×3 , в каждой клетке которой изображены некоторые символы в произвольном порядке. В трех клетках четвертого столбца и в трех клетках четвертой строки также расположены некоторые символы, но образованные на основе некоторой функциональной зависимости, задуманной автором-составителем теста или выбранной компьютером случайным образом из заданного семейства функций. Тестируемый должен обнаружить функциональную зависимость и заполнить шестнадцатую клетку, расположенную в правом нижнем углу матрицы.

X_1	X_2	X_3	$\rightarrow f(X_1, X_2, X_3)$
X_4	X_5	X_6	$\rightarrow f(X_4, X_5, X_6)$
X_7	X_8	X_9	$\rightarrow f(X_7, X_8, X_9)$
\downarrow $f(X_1, X_4, X_7)$	\downarrow $f(X_2, X_5, X_8)$	\downarrow $f(X_3, X_6, X_9)$	

Если вычислять значение функции от трех аргументов, расположенных в четвертом столбце матрицы, то получим значение

$$f(f(X_1, X_2, X_3), f(X_4, X_5, X_6), f(X_7, X_8, X_9)).$$

Если вычислять значение функции от трех аргументов, расположенных в четвертой строке матрицы, то получим значение

$$f(f(X_1, X_4, X_7), f(X_2, X_5, X_8), f(X_3, X_6, X_9)).$$

Найдем функции $f(X, Y, Z)$, удовлетворяющие условию

$$\begin{aligned} f(f(X_1, X_2, X_3), f(X_4, X_5, X_6), f(X_7, X_8, X_9)) = \\ = f(f(X_1, X_4, X_7), f(X_2, X_5, X_8), f(X_3, X_6, X_9)) \end{aligned} \quad (2)$$

Если матрица аргументов является симметричной, то элементы четвертой строки и четвертого столбца симметричны относительно главной диагонали матрицы A , и поэтому равенство (2) выполняется для любой булевой функции. В дальнейшем требуется выполнение равенства (2) для любого набора аргументов.

Функции, удовлетворяющие условию (2), назовем *функциями коммутативности теста*. Схема вычисления значения булевой функции на тесте не зависит от маршрута следования по тесту. На рис. 1 один маршрут следования при вычислении значения функции на тесте показан сплошными линиями, а другой маршрут — пунктирными линиями.

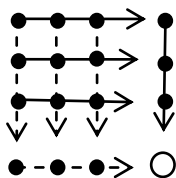


Рис. 1

Тест называется репрезентативным, если среди наборов матрицы аргументов в строках и среди наборов в столбцах представлены все возможные наборы для трех аргументов. Другими словами, тест является репрезентативным, если выборка наборов аргументов является репрезентативной, т.е. содержит все наборы аргументов.

Например, тест $\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & \end{pmatrix}$ не является репрезентативным,

т.к. во всех строках и во всех столбцах матрицы аргументов содержится только два различных набора (101), (010), которым поставлены в соответствие числа 1 и 0. Булеву функцию по этим двум наборам определить однозначно невозможно. Существует 2^6 логических функций, удовлетворяющих этому условию. Даже среди функций, удовлетворяющих условию коммутативности теста, существует шесть логических функций: $X, Z, XZ, X \vee Z, \bar{Y}, \overline{X \oplus Y \oplus Z}$, которые наборам (101), (010) ставят в соответствие числа 1 и 0. Но данный тест можно завершить однозначно, т.к. на основании данной функциональной зависимости набору (101) в четвертой строке и такому же набору в четвертом столбце нужно поставить в соответствие единицу. Поэтому элемент a_{44} теста определяется однозначно. В этом случае задание на определение элемента a_{44} можно ставить перед тестируемым, но нельзя ставить задачу по определению логической функции этого теста.

Следующий тест $\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & \end{pmatrix}$ также не является репрезента-

тивным, т.к. во всех строках и во всех столбцах матрицы аргументов содержится только два различных набора (101), (010), которым поставлено в соответствие число 1. Но данный тест допускает двойственное истолкование.

Первая гипотеза для поиска функциональной зависимости. В четвертой строке и в четвертом столбце нужно записать результат применения функциональной зависимости, используя набор (1,1,1). Для заполнения элемента a_{44} нужна информация о наборе (1,1,1), но ее нет в представленном тесте. В этом случае тест завершить невозможно.

Вторая гипотеза для поиска функциональной зависимости. Всем наборам в строках и наборам в столбцах поставлено в соответствие одно и то же число 1, поэтому и набору (1,1,1) поставим в соответствие число 1. Двойственность истолкования порождена неопределенностью функции.

Чтобы избежать двойственного толкования в матричных тестах, будем рассматривать в дальнейшем только репрезентативные тесты.

Выясним ограничения на формирование репрезентативного теста, а фактически на формирование матрицы аргументов. Если каждая клетка матрицы аргументов содержит только по одному символу, то получаем три набора символов в первых трех строках и три набора символов в первых трех столбцах. Общее количество наборов символов в матрице аргументов равно шести и тест получается не репрезентативным. Пусть в каждой клетке матрицы аргументов содержится по два символа и они расположены так, что из них можно сформировать наборы вдоль строк и вдоль столбцов, тогда получаем всего 12 наборов. Далее будут приведены примеры, которые показывают, что эти символы можно иногда так расположить в клетках матрицы аргументов, что получим восемь различных наборов символов, состоящих из трех символов. В клетках матрицы аргументов должно быть одно и то же число символов (не менее двух), причем наборы аргументов, образованные из них, должны содержать восемь различных наборов.

Множество функций коммутативности теста от n -переменных обозначим K_n . Множество K_2 содержит десять булевых функций: $f_0(A, B) = 0$, $f_1(A, B) = AB$, $f_3(A, B) = A$, $f_5(A, B) = B$, $f_6(A, B) = A \oplus B$, $f_7(A, B) = A + B$, $f_9(A, B) = A \Leftrightarrow B$, $f_{10}(A, B) = \bar{B}$, $f_{12}(A, B) = \bar{A}$, $f_{15}(A, B) = 1$.

Теорема 1. Если функция от трех аргументов реально зависит только от одной переменной или только от двух переменных и эта функция принадлежит множеству K_2 , то она принадлежит и множеству K_3 .

Доказательство. Пусть функция зависит только от одного аргумента, тогда она является одной из следующих функций: $X, Y, Z, \bar{X}, \bar{Y}, \bar{Z}$.

Для функции, равной одной из переменных, например, $f(X, Y, Z) = Z$, доказательство следует из матрицы

X_1	X_2	X_3	X_3
X_4	X_5	X_6	X_6
X_7	X_8	X_9	X_9
X_7	X_8	X_9	X_9

Для функции, равной отрицанию одной из переменных, например, $f(X, Y, Z) = \bar{Y}$, доказательство следует из матрицы

X_1	X_2	X_3	\bar{X}_2
X_4	X_5	X_6	\bar{X}_5
X_7	X_8	X_9	\bar{X}_8
\bar{X}_4	\bar{X}_5	\bar{X}_6	\bar{X}_5

Пусть функция зависит только от двух переменных, например, $f(X, Y, Z) = f(X, Z)$, и $f(X, Z) \in K_2$, т.е.

$f(f(A, B), f(C, D)) = f(f(A, C), f(B, D))$, тогда

X_1	X_2	X_3	$\rightarrow f(X_1, X_3)$
X_4	X_5	X_6	\dots
X_7	X_8	X_9	$\rightarrow f(X_7, X_9)$
\downarrow $f(X_1, X_7)$	\dots	\downarrow $f(X_3, X_9)$	

Условие коммутативности теста в этом случае выполняется, т.к. $f(f(X_1, X_3), f(X_7, X_9)) = f(f(X_1, X_7), f(X_3, X_9))$.

Используя множество K_2 булевых функций от двух переменных, по теореме 1 можно получить некоторые булевы функции:

из множества K_3 : $X \oplus Y, X \oplus Z, Y \oplus Z, X \sim Y, X \sim Z, Y \sim Z, X \vee Y, X \vee Z, Y \vee Z, XY, XZ, YZ$.

Теорема 2. Существует 24 булевы функции, удовлетворяющие условию коммутативности теста:

$0, 1, X, Y, Z, \bar{X}, \bar{Y}, \bar{Z}, X \vee Y, X \vee Z, Y \vee Z, XY, XZ, YZ, X \oplus Y, X \oplus Z, Y \oplus Z, X \sim Y, X \sim Z, Y \sim Z, XYZ, X \vee Y \vee Z,$

$X \oplus Y \oplus Z = X \sim Y \sim Z, \overline{X \oplus Y \oplus Z} = X \oplus Y \oplus Z \oplus 1.$

Доказательство. Множество булевых функций от трех переменных состоит из $2^{2^3} = 2^8 = 256$ функций. Чтобы предъявить 232 функции, не удовлетворяющие условию коммутативности теста, достаточно предъявить хотя бы по одному набору матриц

$$\begin{pmatrix} X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 \\ X_7 & X_8 & X_9 \end{pmatrix},$$

на котором равенство (2) нарушается. Данную матрицу-набор значений удобнее записывать как матрицу-строку из девяти элементов. Также нужно дополнительно указать саму функцию, как набор ее значений в виде строки из восьми значений. Таким образом, нужно перечислить 232 строки, состоящие, по крайней мере, из 17 символов.

Чтобы сократить перечисление таких наборов значений функций и значений элементов матрицы, мы приведем небольшую программу, написанную на языке QBasic, изменяя которую, можно решать различные задачи.

Вначале поясним идею алгоритма моделирования данной задачи по сортировке функций, удовлетворяющих или не удовлетворяющих равенству (2).

Пусть наборы трех аргументов упорядочены в лексикографическом порядке, т.е. расположены следующим образом:

$(000) \prec (001) \prec (010) \prec (011) \prec (100) \prec (101) \prec (110) \prec (111).$

Булеву функцию от трех аргументов на этих наборах можно задать упорядоченным набором значений $(r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7).$

Рассмотреть все значения булевых функций от трех аргументов можно простым перечислением, осуществив восемь вложенных циклов и изменяя булеву переменную от 0 до 1.

Сохраним значения конкретной булевой функции в одномерном массиве $M(8)$, содержащем 8 ячеек, чтобы потом было удобно обращаться с этими значениями булевой функции.

Будем изменять значения элементов X_1, X_2, \dots, X_9 матрицы, организовав девять циклов и изменяя булевы переменные от 0 до 1.

Для вычисления значения булевой функции на упорядоченном наборе $(X_i X_j X_k)$ определим номер этого набора по формуле $4 \cdot X_i + 2 \cdot X_j + X_k$ и вызовем значение ячейки массива с таким номером. Повторяя эту операцию к наборам на матрице и сравнивая значения в равенстве (2), найдем функции, не удовлетворяющие условию коммутативности теста и один из наборов, на котором нарушается это равенство.

Следующая программа перечисляет наборы значений функции, удовлетворяющей условию коммутативности теста на всех наборах матрицы, и указывает номер булевой функции.

SCREEN 12

CLS

DIM f AS SINGLE

DIM M(i = 0 TO 7)

k = 0

FOR r0 = 0 TO 1: FOR r1 = 0 TO 1: FOR r2 = 0 TO 1

FOR r3 = 0 TO 1: FOR r4 = 0 TO 1: FOR r5 = 0 TO 1

FOR r6 = 0 TO 1: FOR r7 = 0 TO 1

nf=r0*2^7+r1*2^6+r2*2^5+r3*2^4+r4*2^3+r5*2^2+r6*2+r7

M(0) = r0: M(1) = r1: M(2) = r2: M(3) = r3

M(4) = r4: M(5) = r5: M(6) = r6: M(7) = r7

FOR X1 = 0 TO 1: FOR X2 = 0 TO 1: FOR X3 = 0 TO 1

FOR X4 = 0 TO 1: FOR X5 = 0 TO 1: FOR X6 = 0 TO 1

FOR X7 = 0 TO 1: FOR X8 = 0 TO 1: FOR X9 = 0 TO 1

f1 = M(4 * X1 + 2 * X2 + X3)

f2 = M(4 * X4 + 2 * X5 + X6)

f3 = M(4 * X7 + 2 * X8 + X9)

fp = M(4 * f1 + 2 * f2 + f3)

f4 = M(4 * X1 + 2 * X4 + X7)

f5 = M(4 * X2 + 2 * X5 + X8)

f6 = M(4 * X3 + 2 * X6 + X9)

fl = M(4 * f4 + 2 * f5 + f6)

IF fp = fl THEN k = k + 1 ELSE GOTO 1

NEXT X9, X8, X7, X6, X5, X4, X3, X2, X1

PRINT r0; r1; r2; r3; r4; r5; r6; r7; "..."; "f"; nf

1

NEXT r7, r6, r5, r4, r3, r2, r1, r0

```

DIM X, Y, Z AS SINGLE
PRINT 0; 0; 0; 0; 1; 1; 1; 1; "..."; "X"
PRINT 0; 0; 1; 1; 0; 0; 1; 1; "..."; "Y"
PRINT 0; 1; 0; 1; 0; 1; 1; 1; "..."; "Z"

```

Приведем примеры функций, указанных программой как удовлетворяющие условию коммутативности теста. Значения функции и номер функции напечатаны компьютером, а формула записи функции дописана потом нами.

```

(0 0 0 0 0 0 0 0) =  $f_0 = 0$ 
(0 0 0 0 0 0 0 1) =  $f_1 = XYZ$ 
(0 0 0 0 0 0 1 1) =  $f_3 = XY$ 
(0 0 0 0 0 1 0 1) =  $f_5 = XZ$ 
(0 0 0 0 1 1 1 1) =  $f_{15} = X$ 
(0 0 0 1 0 0 0 1) =  $f_{17} = YZ$ 
(0 0 1 1 0 0 1 1) =  $f_{51} = Y$ 
(0 0 1 1 1 1 0 0) =  $f_{60} = X \oplus Y$ 
(0 0 1 1 1 1 1 1) =  $f_{63} = X \vee Y$ 
(0 1 0 1 0 1 0 1) =  $f_{85} = Z$ 
(0 1 0 1 1 0 1 0) =  $f_{90} = X \oplus Z$ 
(0 1 0 1 1 1 1 1) =  $f_{95} = X \vee Z$ 
(0 1 1 0 0 1 1 0) =  $f_{102} = Y \oplus Z$ 
(0 1 1 0 1 0 0 1) =  $f_{105} = X \oplus Y \oplus Z$ 
(0 1 1 1 0 1 1 1) =  $f_{119} = Y \vee Z$ 
(0 1 1 1 1 1 1 1) =  $f_{127} = X \vee Y \vee Z$ 
(1 0 0 1 0 1 1 0) =  $f_{150} = \overline{X \oplus Y \oplus Z}$ 
(1 0 0 1 1 0 0 1) =  $f_{153} = Y \sim Z$ 
(1 0 1 0 0 1 0 1) =  $f_{165} = X \sim Z$ 
(1 0 1 0 1 0 1 0) =  $f_{170} = \overline{Z}$ 
(1 1 0 0 0 0 1 1) =  $f_{195} = X \sim Y$ 
(1 1 0 0 1 1 0 0) =  $f_{204} = \overline{Y}$ 
(1 1 1 1 0 0 0 0) =  $f_{240} = \overline{X}$ 
(1 1 1 1 1 1 1 1) =  $f_{255} = 1$ 

```

```

0 0 0 0 1 1 1 1 ...X
0 0 1 1 0 0 1 1 ...Y
0 1 0 1 0 1 1 1 ...Z.

```

После распечатки значений функции нужно определить формулу этой функции. Для этого составим карту Карно этого набора и минимизируем совершенную дизъюнктивную нормальную форму. Например, записываем набор значений $f_{95} = (0\ 1\ 0\ 1\ 1\ 1\ 1)$ на карту Карно.

Объединяя соседние единицы в квадратах, получаем $f_{95} = X \vee Z$.

	X	X	\overline{X}	\overline{X}
Y	1	1	1	
\overline{Y}	1	1	1	
	\overline{Z}	Z		\overline{Z}

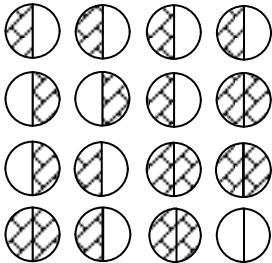


Рис. 2

На рис. 2 представлен тест, построенный на основе функции $f_{95} = X \vee Z$.

Для функции $(1\ 1\ 0\ 0\ 1\ 1\ 0\ 0)$ получаем карту Карно. Объединяя соседние единицы в строке, получаем $f_{204} = \overline{Y}$. На рис. 3 представлен тест, построенный на основе функции $f_{204} = \overline{Y}$.

	X	X	\overline{X}	\overline{X}
Y				
\overline{Y}	1	1	1	1
	\overline{Z}	Z		\overline{Z}

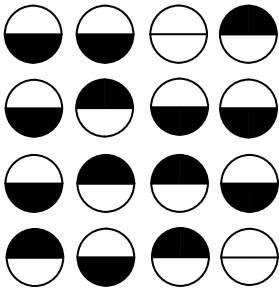


Рис. 3

Для функции $f_{127} = (0\ 1\ 1\ 1\ 1\ 1\ 1\ 1)$ аналогично получаем заполненную карту Карно. Объединяя соседние единицы в квадратах и единицы в первой строке, получаем $f_{95} = X \vee Y \vee Z$.

	X	X	\bar{X}	\bar{X}
Y	1	1	1	1
\bar{Y}	1	1	1	
	\bar{Z}	Z		\bar{Z}

Для функции $f_{90} = (0\ 1\ 0\ 1\ 1\ 0\ 1\ 0)$ получаем карту Карно.

	X	X	\bar{X}	\bar{X}
Y	1		1	
\bar{Y}	1		1	
	\bar{Z}	Z		\bar{Z}

Рис. 4

Объединяя соседние единицы в первом и во втором столбцах, получаем $f_{90} = X\bar{Z} \vee \bar{X}Z = X \oplus Z$. На рис. 4 представлен тест, построенный на основе функции $f = X \oplus Z$. Ответом является пустая шестнадцатая клетка.

Для функции $f_{105} = (0\ 1\ 1\ 0\ 1\ 0\ 0\ 1)$ получаем карту Карно. Единицы в клетках карты Карно нельзя объединить. Выписываем совершенную дизъюнктивную нормальную форму

$f_{105} = XYZ \vee \bar{X}Y\bar{Z} \vee X\bar{Y}\bar{Z} \vee \bar{X}\bar{Y}Z$ или $f_{105} = X \oplus Y \oplus Z$. Отметим, что карта Карно в этом случае не упростила запись. Формулу $f_{105} = XYZ \vee \bar{X}Y\bar{Z} \vee X\bar{Y}\bar{Z} \vee \bar{X}\bar{Y}Z$ можно было записать непосредственно из способа задания функции строкой

$f_{105} = (0\ 1\ 1\ 0\ 1\ 0\ 0\ 1)$.

	X	X	\bar{X}	\bar{X}
Y		1		1
\bar{Y}	1		1	
	\bar{Z}	Z		\bar{Z}

Рис. 5

На рис. 5 представлен тест, построенный на основе функции $f_{105} = X \oplus Y \oplus Z$.

Аналогично находим формулы всех остальных функций, удовлетворяющих условию коммутативности теста.

Двадцать функций от одного или двух аргументов из полученных компьютером функций удовлетворяют условию (2) согласно теореме 2.

Осталось проверить выполнение условия (2) только для функций, существенно зависящих от трех аргументов, т.е. для функций XYZ , $X \vee Y \vee Z$, $X \oplus Y \oplus Z = X \sim Y \sim Z$,

$$\overline{X \oplus Y \oplus Z} = X \oplus Y \oplus Z \oplus 1.$$

Для каждой из этих функций выполняются коммутативный и ассоциативный законы, поэтому перегруппировкой и перестановкой проверяется равенство (2) для каждой из этих функций. Например, для функции $f(X, Y, Z) = X \oplus Y \oplus Z$ получаем тест.

X_1	X_2	X_3	$X_1 \oplus X_2 \oplus X_3$
X_4	X_5	X_6	$X_4 \oplus X_5 \oplus X_6$
X_7	X_8	X_9	$X_7 \oplus X_8 \oplus X_9$
$X_1 \oplus X_4 \oplus X_7$	$X_2 \oplus X_5 \oplus X_8$	$X_3 \oplus X_6 \oplus X_9$	

Используя свойства сложения по модулю 2, получаем

$$\begin{aligned} (X_1 \oplus X_2 \oplus X_3) \oplus (X_4 \oplus X_5 \oplus X_6) \oplus (X_7 \oplus X_8 \oplus X_9) = \\ = (X_1 \oplus X_4 \oplus X_7) \oplus (X_2 \oplus X_5 \oplus X_8) \oplus (X_3 \oplus X_6 \oplus X_9) \end{aligned}$$

Теорема 2 доказана.

§ 17. Компьютерное тестирование с помощью булевых функций от трех переменных

Актуальность использования булевых функций от трех аргументов.

Логические тесты с булевыми функциями $f(X, Y)$ от двух переменных, удовлетворяющие условию коммутативности теста

$$f(f(A, B), f(C, D)) = f(f(A, C), f(B, D)), \quad (1)$$

изучались в работах [20, 21]. В этом параграфе рассматривается компьютерный вариант разработанного теста с использованием функций от трех переменных.

Множество K_2 булевых функций от двух переменных, удовлетворяющих условию коммутативности теста, содержит десять булевых функций: $f_0(X, Y) = 0$, $f_1(X, Y) = XY$, $f_3(X, Y) = X$,

$$f_5(X, Y) = Y, f_6(X, Y) = X \oplus Y, f_7(X, Y) = X \vee Y,$$

$$f_9(X, Y) = X \Leftrightarrow Y, f_{10}(X, Y) = \bar{Y}, f_{12}(X, Y) = \bar{X}, f_{15}(X, Y) = 1.$$

Разработанные ранее программные продукты для функций от двух переменных в логических тестах были нами модифицированы.

На рис. 1 представлены результаты тестирования. В тесте используются функции от двух аргументов. В десятом вопросе операция сложения по модулю 2 на экране компьютера обозначена знаком $+$.

Модификацией этого теста по сравнению с ранее используемыми тестами является введение функции фиксирования времени, затраченного на определение каждой логической операции и ответа на представленное задание. Сравнивая время, затраченное на выполнение всех десяти заданий теста, можно определить скорость выполнения мыслительных операций. Если результаты тестирования оказались низкими, а времени на выполнение операций было затрачено мало, то в этом случае тестируемый выполнял поверхностный анализ условий теста.

На самые простые операции $f_0(X, Y) = 0$ и $f_{15}(X, Y) = 1$ при тестировании затрачено минимальное время.

Интересно сравнить группу операций

$$f_3(X, Y) = X,$$

$$f_{12}(X, Y) = \bar{X} \text{ с группой операций } f_5(X, Y) = Y,$$

$f_{10}(X, Y) = \bar{Y}$. На первую из этих групп, т.е. операции f_3 и f_{12} затрачено меньше времени, чем на вторую группу, хотя они кажутся одинаковыми с точки зрения математической сложности. Это объясняется строением теста. При анализе информации на экране компьютера тестируемый сравнивает результирующий третий

Анализ различных отчетов по затраченному времени показал, что следующей группой по количеству затраченного времени являются функции — $f_1(X, Y) = XY$ и $f_7(X, Y) = X \vee Y$. В зависимости от разреженности теста или, наоборот, от его насыщенности символами, время, затраченное на определение этих функций, может значительно отличаться, что показывает протокол затраченного времени на рис. 1.

При тестировании возникают ошибки двух типов.

Ошибки первого типа

возникают в связи с тем, что неправильно определена функция при анализе информации в первых восьми квадратах.

Ошибки второго типа возникают в тех случаях, когда правильно определена функция при анализе информации в первых восьми квадратах, но в процессе применения найденной функции для заполнения девятого квадрата совершена ошибка. Наблюдение за тестируемым в этом случае показывает, что тестируемый после сравнения своего результата с результатом, представленным компьютером в окне «Правильный результат», иногда произносит: и у меня была получена эта функция. Тестируемый

  	  	  	<p>Ваш ответ: ПРАВИЛЬНЫЙ!</p>
Правильный результат:			
Выход			

Статистика			
Номер вопроса	Ответ	Функция	Время (сек.)
Вопрос 1		0	11
Вопрос 2		X	19
Вопрос 3		X или Y	35
Вопрос 4		X и Y	17
Вопрос 5		1	17
Вопрос 6		X <-> Y	21
Вопрос 7		не X	19
Вопрос 8		Y	40
Вопрос 9		не Y	33
Вопрос 10		X + Y	81

Суммарные результаты	
Всего	9 правильных ответов
Затрачено	293 секунд(ы)
Оценка	5 балл(а)(ов) из 5

Рис. 1

не всегда контролирует свои действия при применении найденной функции для ввода ответа.

Тесты имеют важное значение в развитии и оценке логического мышления. Апробация ранее разработанных тестов показала их эффективность для тренировки логического мышления и выстраивания самостоятельной траектории по освоению операций математической логики. Учащиеся, обучающиеся в классах с углубленным изучением математики и информатики, а также студенты с хорошей математической подготовкой отмечали, что тесты с функциями от двух переменных они быстро осваивают и им требуются более сложные тесты. В связи с этим возникла необходимость усложнения тестов, т.е. разработки компьютерных тестов с функциями от трех переменных.

Отметим позитивную роль компьютера для развития логического мышления. Чтобы сопоставить свой ответ с условием задачи, их нужно расположить в большинстве случаев рядом. Обычно ответ занимает меньше места, чем условие задачи. Ответ нужно «принести» к условию задачи. Если листы книги проклеены или скреплены скрепкой, то, развернуть книгу так, чтобы зафиксировать эти положения для размышления над ответом, невозможно, т.к. после нескольких таких действий листы книги рассыпаются или книга деформируется.

Нахождение на экране компьютера условия теста, представление вводимого ответа и появление правильного ответа после проверки введенного ответа является компактной записью для каждого задания и позволяет проводить работу над ошибками.

Набор тестов, предлагаемый в различных методических пособиях, ограничен. Если ответов к тестам нет, то трудно оценить самостоятельно ответы на тесты, а если ответы есть, то учащиеся смотрят в них раньше, чем будет получен ответ. Поэтому становится актуальной проблема по разработке компьютерного тренинга навыков логического мышления. В компьютере невозможно получить ответ раньше, чем будет введен свой ответ на поставленный вопрос. В предложенном автоматическом тестировании ответ на каждый вопрос всегда появляется на экране после введенного учащимся ответа.

Компьютерное тестирование предлагает большой спектр тестов с различными начальными условиями, причем самое большое

достижение в том, что тестирование можно осуществлять в тренировочном режиме. Сопоставление своего ответа и правильного ответа, предложенного компьютером, позволяет провести анализ совершенных мыслительных действий и внести корректирующие действия при следующем ответе.

В результате осуществляется построение студентом индивидуальной образовательной траектории. Компьютерный тренинг логического мышления становится элементом самосовершенствования в индивидуальной образовательной траектории.

Для автоматизации тиражирования матричных тестов на основе логических функций от трех переменных было создано программное средство со следующим набором функций:

1. Формирование матричных тестов на основе логических функций.
2. Визуализация теста в виде матрицы четвертого порядка.
3. Оценка результатов тестирования.
4. Ведение журнала о ходе тестирования.

В настоящей работе рассматривается автоматизация составления матричных тестов и их оценка. Объективность оценивания обеспечивается, во-первых, тем, что компьютер в случайном порядке предлагает матричные задачи с различными логическими функциями. Во-вторых, компьютер для одной и той же функции заполняет пятнадцать квадратов в случайном порядке, в то время как при стандартном тестировании предлагается ограниченный набор условий тестов. В-третьих, компьютер сам объективно оценивает ответ и только потом сообщает правильный ответ. Отвечающий может сравнить свой ответ с правильным ответом. Компьютерная проверка является более объективной, компактной по времени, лишенной пристрастия и антипатий контролирующего эксперта.

Структура тестов с функцией от трех переменных

Рассмотрим матричный тест, сформированный в виде матрицы A размером 4×4 (рис. 2). В левом верхнем углу расположена матрица аргументов A , размером 3×3

$$\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & L \end{pmatrix},$$

в каждой клетке которой изображены некоторые символы в произвольном порядке. В трех клетках четвертого столбца и в трех клетках четвертой строки также расположены некоторые символы, но образованные на основе некоторой функциональной зависимости, задуманной автором-составителем теста или выбранной компьютером случайным образом из заданного семейства функций. Тестируемый должен обнаружить функциональную зависимость и заполнить шестнадцатую клетку, расположенную в правом нижнем углу матрицы.

<i>A</i>	<i>B</i>	<i>C</i>	$f(A, B, C)$
<i>D</i>	<i>E</i>	<i>F</i>	$f(D, E, F)$
<i>G</i>	<i>H</i>	<i>L</i>	$f(G, H, L)$
$f(A, D, G)$	$f(B, E, H)$	$f(C, F, L)$	

Рис. 2

Если вычислять значение функции от трех аргументов, расположенных в четвертом столбце матрицы, то получим значение

$$f(f(A, B, C), f(D, E, F), f(G, H, L)).$$

Если вычислять значение функции от трех аргументов, расположенных в четвертой строке матрицы, то получим значение

$$f(f(A, D, G), f(B, E, H), f(C, F, L)).$$

Результат в шестнадцатой клетке не должен зависеть от направления вычисления, поэтому найдем функции $f(X, Y, Z)$, удовлетворяющие условию

$$\begin{aligned} &f(f(A, B, C), f(D, E, F), f(G, H, L)) = \\ &f(f(A, D, G), f(B, E, H), f(C, F, L)). \end{aligned} \quad (2)$$

В §16 получены все логические функции от трех аргументов, удовлетворяющие условию (2).

Теорема. Существует 24 булевы функции $f(X, Y, Z)$ от трех аргументов, удовлетворяющие условию коммутативности теста:

$0, 1, X, Y, Z, \bar{X}, \bar{Y}, \bar{Z}, X \vee Y, X \vee Z, Y \vee Z, XY, XZ, YZ,$
 $X \oplus Y, X \oplus Z, Y \oplus Z, X \sim Y, X \sim Z, Y \sim Z, XYZ, X \vee Y \vee Z,$
 $X \oplus Y \oplus Z = X \sim Y \sim Z, \overline{X \oplus Y \oplus Z} = X \oplus Y \oplus Z \oplus 1.$

Компьютерная реализация тестирования

Анализ рис. 3 показывает, что символ в результирующей клетке встречается в том и только в том случае, если он содержится в первой и третьей соответствующих клетках ряда (строки или столбца). Обозначим конъюнкцию двух высказываний A и C через AC (рис. 4).

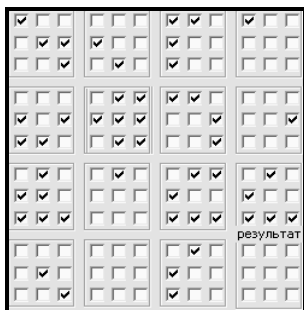


Рис. 3

A	B	C	AC
D	E	F	DF
G	H	L	GL
AG	BH	CL	

Рис. 4

Получаем матрицу с элементами (рис. 4). Используя полученную зависимость, получаем выражение $(AC)(GL)$ при заполнении по столбцу и выражение $(AG)(CL)$ при заполнении по строке для шестнадцатого квадрата. Обсуждаем с учащимися, что эти формулы равносильны, и заполняем шестнадцатый квадрат.

На рис. 5 задана функция, равная конъюнкции трех переменных, т.е. $f(X, Y, Z) = XYZ$. Получаем матрицу с элементами (рис. 5, 6). В шестнадцатый квадрат записываем результат применения конъюнкции к четвертому, восьмому, двенадцатому квадратам или результат применения конъюнкции к 13, 14, 15 квадратам.

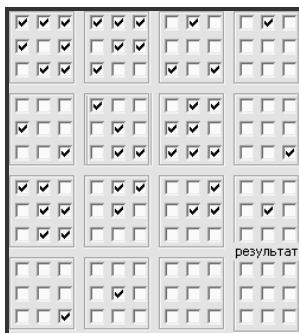


Рис. 5

<i>A</i>	<i>B</i>	<i>C</i>	<i>ABC</i>
<i>D</i>	<i>E</i>	<i>F</i>	<i>DEF</i>
<i>G</i>	<i>H</i>	<i>L</i>	<i>GHL</i>
<i>ADG</i>	<i>BEH</i>	<i>CFL</i>	

Рис. 6

Для матричной задачи (рис. 7) замечаем, что общие элементы на одинаковых местах в первом и третьем рядах отбрасываются. Элементы уникальные для каждого ряда, т.е. которые встречаются только в одном ряду, остаются. Если наличие символа в клетке обозначим 1, а отсутствие символа через 0, получаем правило $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $0 \oplus 0 = 0$.

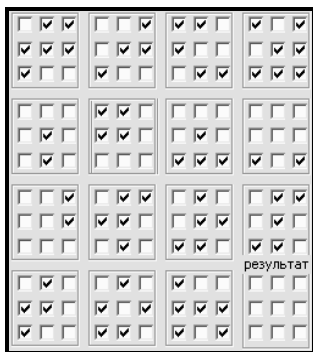


Рис. 7

<i>A</i>	<i>B</i>	<i>C</i>	$A \oplus C$
<i>D</i>	<i>E</i>	<i>F</i>	$D \oplus F$
<i>G</i>	<i>H</i>	<i>L</i>	$G \oplus L$
$A \oplus G$	$B \oplus H$	$C \oplus L$	

Рис. 8

Функция, построенная по такому правилу, называется *операцией сложения по модулю 2* (не равносильностью) и обозначается $A \oplus B$.

На рис. 7 задана функция, равная сумме по модулю 2 для двух переменных, т.е. $f(X, Y, Z) = X \oplus Z$. Получаем матрицу с элементами (рис. 8). В шестнадцатый квадрат записываем результат

применения суммы по модулю 2 к четвертому и двенадцатому квадратам или результат применения суммы по модулю 2 к тринадцатому и пятнадцатому квадратам.

На рис. 9 задана функция, равная сумме по модулю 2 для трех переменных, т.е. $f(X,Y,Z)=X \oplus Y \oplus Z$. Получаем матрицу с элементами (рис. 10). В шестнадцатый квадрат записываем результат $A \oplus B \oplus C \oplus D \oplus E \oplus F \oplus G \oplus H \oplus L$.

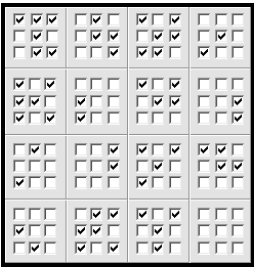


Рис. 9

	B	C	$A \oplus B \oplus C$
D	E	F	$D \oplus E \oplus F$
G	H	L	$G \oplus H \oplus L$
$A \oplus D \oplus G$	$B \oplus E \oplus H$	$C \oplus F \oplus L$	

Рис. 10

На рис. 11 задана функция от трех аргументов, равная второму аргументу, т.е. $f(X,Y,Z)=Y$. Получаем матрицу с элементами (рис. 12). Шестнадцатый квадрат заполняется переменной E .

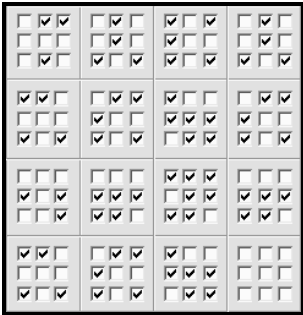


Рис. 11

A	B	C	B
D	E	F	E
G	H	L	H
D	E	F	E

Рис. 12

На рис. 13 задана функция от трех аргументов, равная отрицанию второго аргумента, т.е. $f(X,Y,Z)=\overline{Y}$. Получаем матрицу

с элементами (рис. 14). В шестнадцатый квадрат записываем выражение \overline{E} .

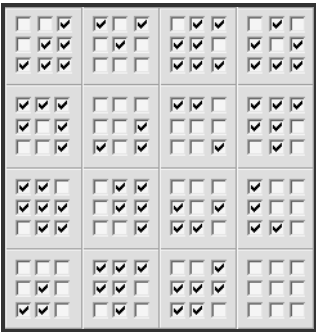


Рис. 13

<i>A</i>	<i>B</i>	<i>C</i>	\overline{Y}
<i>D</i>	<i>E</i>	<i>F</i>	\overline{E}
<i>G</i>	<i>H</i>	<i>L</i>	\overline{H}
\overline{D}	\overline{E}	\overline{F}	\overline{E}

Рис. 14

На рис. 15 задана функция от трех аргументов, принимающая нулевые значения на всех наборах переменных, т.е. $f(X,Y,Z)=0$. Получаем матрицу с элементами (рис. 16). Не заполняя клетки шестнадцатого квадрата, нужно указать компьютеру на проверку результата, нажав клавишу «Проверить».

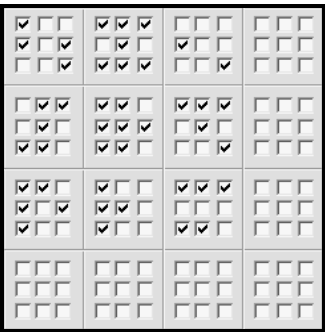


Рис. 15

<i>A</i>	<i>B</i>	<i>C</i>	0
<i>D</i>	<i>E</i>	<i>F</i>	0
<i>G</i>	<i>H</i>	<i>L</i>	0
0	0	0	0

Рис. 16

На рис. 17 задана функция от трех аргументов, принимающая значения на всех наборах переменных, равные 1, т.е. $f(X,Y,Z)=1$. Получаем матрицу с элементами (рис. 18). Все клетки шестнадцатого квадрата следует заполнить данным

символом. На рис. 17 шестнадцатый квадрат представлен до заполнения при решении этой задачи.

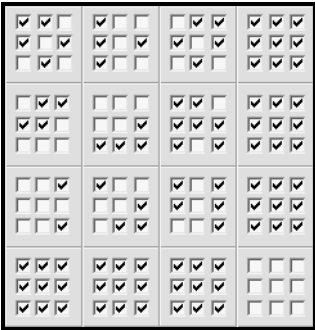


Рис. 17

<i>A</i>	<i>B</i>	<i>C</i>	1
<i>D</i>	<i>E</i>	<i>F</i>	1
<i>G</i>	<i>H</i>	<i>L</i>	1
1	1	1	1

Рис. 18

На рис. 19 задана функция от трех аргументов, равная дизъюнкции второго и третьего аргументов, т.е. $f(X,Y,Z)=Y \vee Z$. Получаем матрицу с элементами (рис. 20). В шестнадцатый квадрат вносим значение $E \vee F \vee H \vee L$.

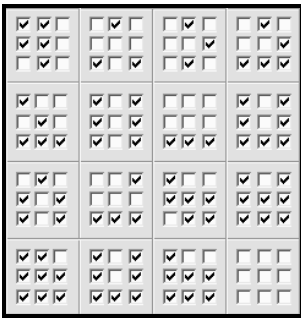


Рис. 19

<i>A</i>	<i>B</i>	<i>C</i>	$B \vee C$
<i>D</i>	<i>E</i>	<i>F</i>	$E \vee F$
<i>G</i>	<i>H</i>	<i>L</i>	$H \vee L$
$D \vee G$	$E \vee H$	$F \vee L$	$(E \vee F) \vee (H \vee L) = (E \vee H) \vee (F \vee L)$
$D \vee G$			

Рис. 20

На рис. 21 задана функция от трех аргументов, равная конъюнкции первого и второго аргументов, т.е. $f(X,Y,Z) = XY$. Получаем матрицу с элементами (рис. 22).

В шестнадцатый квадрат вносим значение $ABDE$. Этот тест полностью аналогичен тесту на рис. 3, т.к. эти два теста используют конъюнкцию для двух аргументов, но эти функции применяются к различным аргументам.

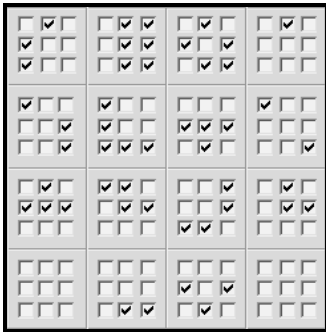


Рис. 21

<i>A</i>	<i>B</i>	<i>C</i>	<i>AB</i>
<i>D</i>	<i>E</i>	<i>F</i>	<i>DE</i>
<i>G</i>	<i>H</i>	<i>L</i>	<i>GH</i>
<i>AD</i>	<i>BE</i>	<i>CF</i>	<i>(AB)(DE)=(AD)(BE)</i>

Рис. 22

На рис. 23 задана функция от трех аргументов, являющаяся эквивалентностью второго и третьего аргументов, т.е. $f(X,Y,Z) = Y \Leftrightarrow Z$. Получаем матрицу с элементами (рис. 24).

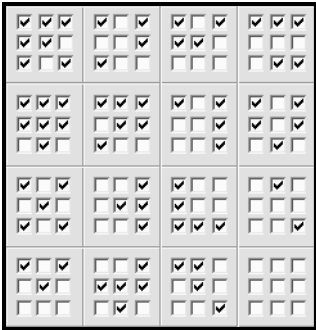


Рис. 23

A	B	C	$B \Leftrightarrow C$
D	E	F	$E \Leftrightarrow F$
G	H	L	$H \Leftrightarrow L$
$D \Leftrightarrow G$	$E \Leftrightarrow H$	$F \Leftrightarrow L$	$(E \Leftrightarrow F) \Leftrightarrow (H \Leftrightarrow L) =$ $= (E \Leftrightarrow H) \Leftrightarrow (F \Leftrightarrow L)$

Рис. 24

Следующая функция от трех аргументов $f(X, Y, Z) = (X \Leftrightarrow Y) \Leftrightarrow Z = (X \Leftrightarrow Y) \Leftrightarrow Z$ удовлетворяет условию $(X \Leftrightarrow Y) \Leftrightarrow Z = X \Leftrightarrow (Y \Leftrightarrow Z)$ и совпадает с функцией $f(X, Y, Z) = X \oplus Y \oplus Z$. Действительно, дважды применяя равенство $X \Leftrightarrow Y = X \oplus Y \oplus 1$, получаем

$$(X \Leftrightarrow Y) \Leftrightarrow Z = (X \oplus Y \oplus 1) \Leftrightarrow Z = X \oplus Y \oplus 1 \oplus Z \oplus 1 = X \oplus Y \oplus Z$$

Функция $f(X, Y, Z) = (X \Leftrightarrow Y) \Leftrightarrow Z$ принимает истинное значение, когда количество истинных высказываний среди высказываний X, Y, Z является нечетным.

Функция $f(X, Y, Z) = \overline{X \oplus Y \oplus Z}$ или равносильная функция $f(X, Y, Z) = \overline{(X \Leftrightarrow Y) \Leftrightarrow Z}$ (рис. 25, 26) принимает истинное значение, когда количество истинных высказываний среди высказываний X, Y, Z является четным.

Рис. 25

A	B	C	$\overline{A \oplus B \oplus C}$
D	E	F	$\overline{D \oplus E \oplus F}$
G	H	L	$\overline{G \oplus H \oplus L}$
$\overline{A \oplus D \oplus G}$	$\overline{B \oplus E \oplus H}$	$\overline{C \oplus F \oplus L}$	

Рис. 26

На рис. 27 задана функция от трех аргументов, равная дизъюнкции трех аргументов, т.е. $f(X, Y, Z) = X \vee Y \vee Z$. Получаем матрицу с элементами (рис. 28).

Рис. 27

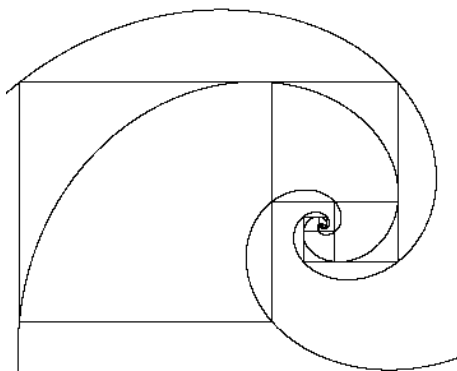
A	B	C	$A \vee B \vee C$
D	E	F	$D \vee E \vee F$
G	H	L	$G \vee H \vee L$
$A \vee D \vee G$	$B \vee E \vee H$	$C \vee F \vee L$	

Рис. 28

Тест с тремя переменными Test. 3×3.exe находится на CD и его можно запустить для автоматического тестирования.

Глава 3

МАТЕМАТИЧЕСКАЯ И КОМПЬЮТЕРНАЯ СМЕСЬ



Что нас ожидает в этой главе?

1. Последовательность квадратов, приводящая к логарифмической спирали.
2. Новая замечательная точка треугольника.
3. Моделирование периодичности траектории математического бильярда в треугольнике.
4. Метод восстановления кривой по заданному радиусу кривизны.

§ 18. Моделирование семейства педальных треугольников

Часто компьютер применяется для упрощения вычислений. Покажем, что применение компьютера для автоматизации геометрических построений также является актуальным направлением. Рассмотрим известную геометрическую задачу о построении семейства педальных треугольников для данного треугольника и ее обобщение для произвольного многоугольника [11, с. 34—37].

Пусть P — произвольная точка внутри данного треугольника ABC и пусть PA_1 , PB_1 , PC_1 — перпендикуляры, опущенные, соответственно, на стороны BC , AC , AB треугольника (рис. 1). Треугольник $A_1B_1C_1$, вершинами которого являются основания перпендикуляров, называется *педальным треугольником* для треугольника ABC и точки P .

Если точка P является центром окружности, описанной около данного треугольника ABC , т.е. является точкой пересечения серединных перпендикуляров к сторонам треугольника, то вершинами педального треугольника являются середины сторон и педальный треугольник называется в этом случае *серединным треугольником* к данному треугольнику ABC .

Если точка P является ортоцентром треугольника ABC , т.е. точкой пересечения его высот, то вершинами педального треугольника являются основания высот исходного треугольника и такой треугольник называется *ортотреугольником*.

Требование о том, чтобы точка P находилась внутри треугольника, можно ослабить, потребовав лишь, чтобы точка P не находилась на окружности, описанной около данного треугольника ABC .

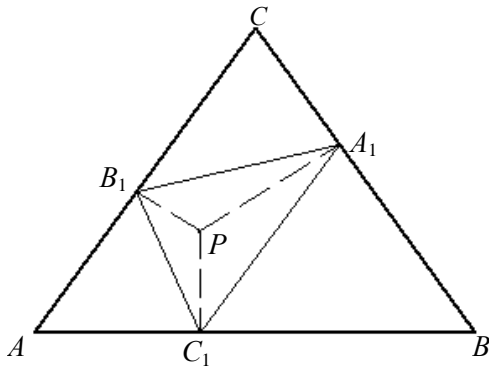


Рис. 1

Для треугольника $A_1B_1C_1$ и той же точки P аналогично построим педальный треугольник $A_2B_2C_2$, который называется *вторым педальным треугольником* для исходного треугольника ABC (рис. 2). Аналогично можно построить третий педальный треугольник $A_3B_3C_3$.

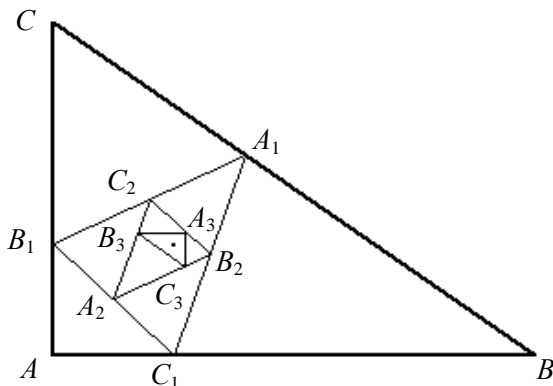


Рис. 2

Известна теорема Нейберга [11, с. 35]: третий педальный треугольник подобен исходному треугольнику.

Доказательство теоремы следует из рис. 3. Треугольник AB_1P является прямоугольным, поэтому центр окружности, описанной около этого треугольника, является серединой отрезка AP . Треугольник AC_1P является прямоугольным, поэтому центр окружности, описанной около этого треугольника, является серединой отрезка AP . Около четырехугольника AB_1C_1P с двумя противоположными прямыми углами можно описать окружность. Следовательно, окружность, описанная около треугольника C_1AB_1 , проходит через точку P . Аналогично, окружности, описанные около треугольников $A_2B_1C_2, B_3C_2A_3$, проходят через точку P .

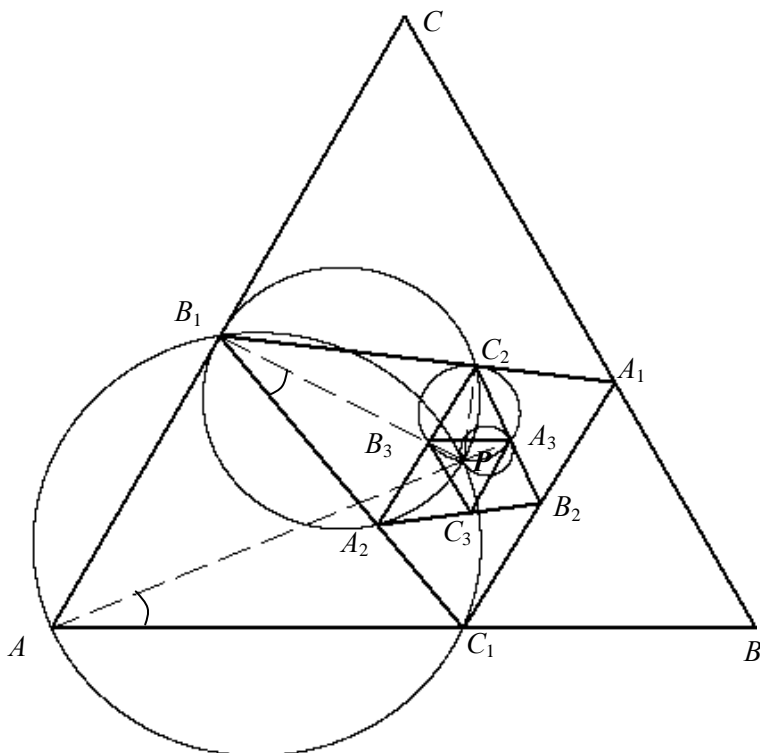


Рис. 3

$\angle C_1AP = \angle C_1B_1P$ как углы, опирающиеся на одну дугу C_1P . Аналогично $\angle C_1B_1P = \angle A_2C_2P$, $\angle A_2C_2P = \angle B_3A_3P$. Следовательно, $\angle BAP = \angle C_1B_1P = \angle A_2C_2P = \angle B_3A_3P$.

Аналогично $\angle BAC = \angle PB_1A_1 = \angle PC_2A_2 = \angle PA_3C_3$.

$\angle A = \angle B_3A_3C_3$, $\angle B = \angle C_3B_3A_3$, $\angle C = \angle A_3C_3B_3$.

Треугольники ABC и $A_3B_3C_3$ подобны и теорема доказана.

Чтобы заметить указанную закономерность в теореме, необходимо построить несколько чертежей с различными треугольниками и произвольной точкой P . Для построения рис. 2 нужно построить девять перпендикуляров. Для построения одного перпендикуляра с помощью циркуля и линейки необходимо выполнить четыре операции, поэтому при построении исходного треугольника

и девяти перпендикуляров необходимо выполнить 39 операций по построению окружностей или отрезков. Построения первых трех перпендикуляров являются независимыми построениями. Построение следующих трех перпендикуляров использует стороны первого педального треугольника, поэтому требования к точности построения являются достаточно высокими.

Если данный треугольник ABC и точку P изменить, то все геометрические построения приходится повторять, т.е. снова выполнить 39 операций. Очевидно, что математическое и компьютерное моделирование могут намного упростить построение семейства педальных треугольников и быстрее варьировать исходными данными.

Вначале решим простую вспомогательную задачу.

В прямоугольной декартовой системе координат даны точка $P(x_0; y_0)$ и уравнение $ax + by + c = 0$ прямой l . Требуется определить координаты точки H , являющейся проекцией точки P на прямую l .

Из уравнения прямой следует, что вектор $\bar{n}(a; b)$ является нормальным вектором для прямой l .

Вектор $\bar{n}(a; b)$ параллелен прямой PH , поэтому он является для этой прямой направляющим вектором.

Параметрические уравнения прямой PH

$$x = x_0 + at, y = y_0 + bt.$$

Координаты точки H найдем из системы уравнений

$$x = x_0 + at, y = y_0 + bt, ax + by + c = 0.$$

Подставляя значения x, y в третье уравнение системы, получаем

$$t = -\frac{ax_0 + by_0 + c}{a^2 + b^2},$$

$$x = x_0 - a \frac{ax_0 + by_0 + c}{a^2 + b^2}, y = y_0 - b \frac{ax_0 + by_0 + c}{a^2 + b^2}. \quad (1)$$

Координаты проекции $H \left(x_0 - a \frac{ax_0 + by_0 + c}{a^2 + b^2}; y_0 - b \frac{ax_0 + by_0 + c}{a^2 + b^2} \right)$.

Продолжим моделирование семейства педальных треугольников.

Пусть $P(x_0, y_0)$ — произвольная точка внутри данного треугольника с вершинами $A(x_1; y_1), B(x_2; y_2), C(x_3; y_3)$. Уравнение прямой AB , проходящей через две данные точки, имеет вид:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \text{ или } (y_2 - y_1)(x - x_1) - (x_2 - x_1)(y - y_1) = 0.$$

Используя формулы (1), получаем для проекции C_1 координаты

$$x = x_0 - (y_2 - y_1) \frac{(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1}{(x_2 - x_1)^2 + (y_2 - y_1)^2},$$

$$y = y_0 + (x_2 - x_1) \frac{(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1}{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Аналогично получаем координаты точек B_1 и C_1 .

Следующая компьютерная программа, написанная на языке Visual Basic, строит данный треугольник, данную точку и семейство педальных треугольников:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
' переход от экранной системы координат к декартовой системе координат
X1 = 1000: Y1 = 1000: X2 = 7000: Y2 = 1000: X3 = 1000: y3 = 5000
' координаты вершин треугольника ABC
x0 = 2500: y0 = 2000 ' координаты точки P
PSet (x0, y0) 'построение точки P
For i = 1 To 4
Line (X1, Y1)-(X2, Y2): Line (X2, Y2)-(X3, y3): Line (X3, y3)-(X1, Y1)
' построение сторон данного треугольника при первом шаге цикла и далее педальных треугольников
d3 = (X2 - X1) ^ 2 + (Y2 - Y1) ^ 2
u3 = x0 - (Y2 - Y1) * ((Y2 - Y1) * x0 - (X2 - X1) * y0 + X2 * Y1 - Y2 * X1) / d3
v3 = y0 + (X2 - X1) * ((Y2 - Y1) * x0 - (X2 - X1) * y0 + X2 * Y1 - Y2 * X1) / d3
' вычисление координат точки C1
d1 = (X3 - X2) ^ 2 + (y3 - Y2) ^ 2
u1 = x0 - (y3 - Y2) * ((y3 - Y2) * x0 - (X3 - X2) * y0 + X3 * Y2 - y3 * X2) / d1
v1 = y0 + (X3 - X2) * ((y3 - Y2) * x0 - (X3 - X2) * y0 + X3 * Y2 - y3 * X2) / d1
' вычисление координат точки A1
d2 = (X1 - X3) ^ 2 + (Y1 - y3) ^ 2
u2 = x0 - (Y1 - y3) * ((Y1 - y3) * x0 - (X1 - X3) * y0 + X1 * y3 - Y1 * X3) / d2
v2 = y0 + (X1 - X3) * ((Y1 - y3) * x0 - (X1 - X3) * y0 + X1 * y3 - Y1 * X3) / d2
```

' вычисление координат точки B1

$X1 = u1$; $Y1 = v1$; $X2 = u2$; $Y2 = v2$; $X3 = u3$; $y3 = v3$

' пере обозначение при выборе полученного педального треугольника в качестве исходного треугольника для дальнейших построений

Next i: End Sub

Теорема Нейберга обобщена на произвольный многоугольник. Рисунок 4 демонстрирует, что четвертый педальный четырехугольник подобен данному четырехугольнику. На рис. 5 пятый педальный пятиугольник подобен данному пятиугольнику.

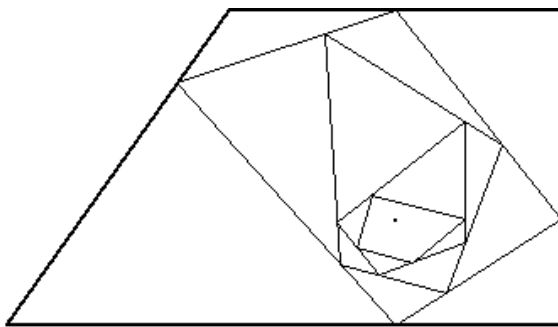


Рис. 4

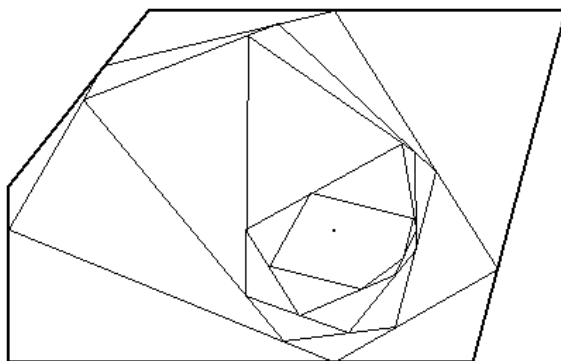


Рис. 5

Из листа бумаги изготовьте треугольник, выберите произвольную точку внутри треугольника и постройте семейство педальных треугольников методом сгибания.

§ 19. Последовательности квадратов и спирали

В математике существуют интересные соотношения, настоящие шедевры. Среди них знаменитая компактная формула $e^{\pi i} + 1 = 0$. В ней аккумулированы сведения о различных понятиях. Найдем другие интересные зависимости.

В квадрате со стороной a расположено n^2 равных кругов (рис. 1), причем каждый круг касается других кругов и (или) сторон квадрата. Какое число n нужно выбрать, чтобы сумма площадей всех кругов составляла наибольшую часть от площади квадрата?

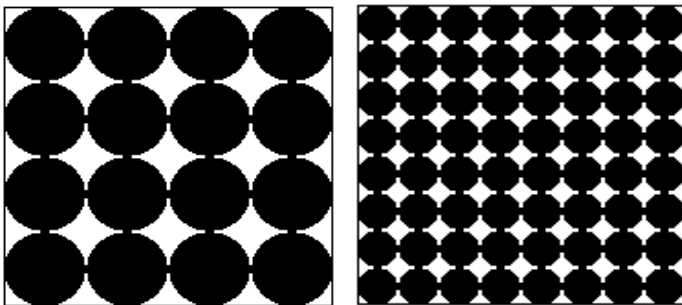


Рис. 1

Ответ покажется странным, но он не зависит от выбранного числа, т.е. для любого числа n отношение суммы площадей всех кругов к площади квадрата является постоянной величиной.

Действительно, радиус круга равен $R = a/(2n)$.

Сумма площадей кругов равна $S = n^2 \pi R^2 = n^2 \frac{a^2}{4n^2} = \frac{\pi a^2}{4}$.

Отношение площади, занятой кругами, к площади квадрата равно $\frac{\pi}{4} \approx 0,785398$ или $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots$.

А какая зависимость для n^3 шаров, расположенных в кубе?

В элективном курсе можно аналогично показать связь математики и информатики, различных разделов из математики: геометрии, комбинаторики, числовых последовательностей, числовых рядов.

Пусть на прямой последовательно отложены отрезки AA_1 , A_1A_2 , A_2A_3 , ..., длина каждого из которых вдвое меньше длины предыдущего отрезка (рис. 2). Пусть длина первого отрезка AA_1 равна 1, тогда длина отрезка AA_n приближается к 2, если число n неограниченно возрастает. Это утверждение следует из равенства $AA_n = AB - A_nB = 2 - \frac{1}{2^{n-1}}$, т.к. точка A_n каждый раз делит отрезок $A_{n-1}B$ пополам.

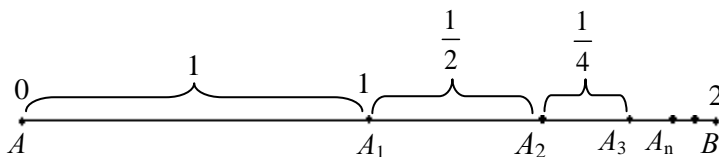


Рис. 2

Второе аналитическое обоснование этого утверждения следует из выражения суммы членов геометрической прогрессии

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = \frac{1}{1 - 1/2} = 2.$$

Последнее равенство в двоичной системе принимает следующий вид:

$$1_2 + 0,1_2 + 0,01_2 + 0,001_2 + \dots = 1,111\dots_2 = 10_2 = 2_{10}.$$

Для точного и быстрого построения последовательности отрезков приведем программу:

```
Private Sub Command1_Click()
a = 2600: xc = 100: Line (xc, 400)-(xc + 2 * a, 400)
DrawWidth = 3: PSet (xc, 400)
For i = 1 To 4
xc = xc + a: Circle (xc, 400), 3: a = a / 3
Next i
End Sub
```

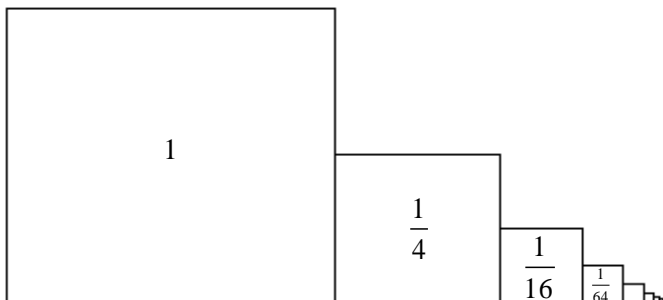


Рис. 3

Рассмотрим ступенчатую фигуру, образованную при построении квадратов на рассмотренной ранее последовательности отрезков (рис. 3).

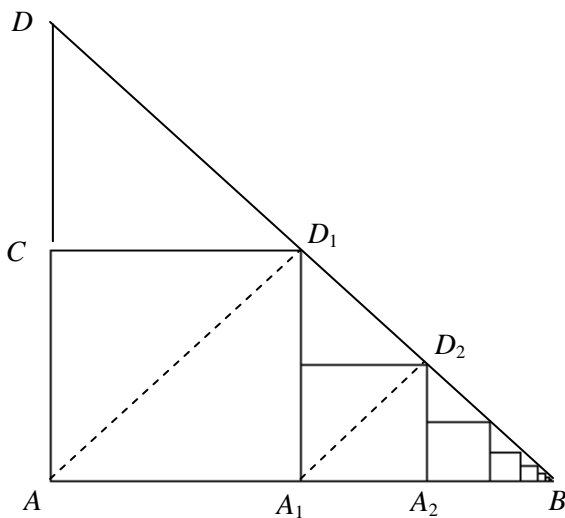


Рис. 4

Площадь ступенчатой фигуры непосредственно вычислить сложно (если читатель не увидел связь с геометрической прогрессией). Если построить дополнительно треугольник ABD (рис. 4), то легко заметить, что каждый квадрат расположен в трапеции, основания которой являются вертикальными отрезками, причем площадь квадрата составляет две третьих от площади

соответствующей трапеции. Если суммировать площади всех трапеций, то получим площадь треугольника, равную 2. Следовательно, площадь ступенчатой фигуры, образованной квадратами, равна $2/3 \cdot 2 = 4/3$.

Замечание. Вместо треугольника ABD можно аналогично построить треугольник ABC и провести аналогичные вычисления.

Программа построения последовательности квадратов имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 2600: xc = 100
Line (xc, 400)-(xc + 2 * a, 400)
For i = 1 To 10
Line (xc, 400)-(xc + a, 400 + a), , B
xc = xc + a: a = a / 2
Next i
End Sub
```

Площадь ступенчатой фигуры вычисляется также по формуле суммы членов геометрической прогрессии:

$$1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \dots = \frac{1}{1-1/4} = 1\frac{1}{3}.$$

Это выражение в двоичной системе счисления также принимает интересный вид:

$$1_2 + 0,01_2 + 0,0001_2 + 0,000001 + \dots = 1,010101\dots_2 = 1,(01)_2 = 1\frac{1}{3_{10}}.$$

Переход в записи одного и того же числа в различных системах счисления имеет интересное применение. Он позволяет представить любое число в виде ряда. Например,

$$\frac{1}{5_{10}} = \frac{1}{101_2}.$$

Выполним деление в двоичной системе счисления. Ниже представлен процесс деления. Выделены этапы деления, чтобы продемонстрировать процесс появления нулей в частном на первом этапе, процесс появления повторяющегося числа в частном,

т.е. 1, и процесс осознания того, что далее в частном будет повторяющаяся группа цифр 011, т.е. появляется периодическая дробь.

$$\begin{array}{l} -1_2 \left| \begin{array}{l} 101_2 \\ 0, \cdot 2 \end{array} \right. \Rightarrow \begin{array}{l} -1000_2 \\ \hline 101 \\ -110 \\ \hline 101 \\ \hline 1 \end{array} \left| \begin{array}{l} 101_2 \\ 0,0011 \cdot 2 \end{array} \right. \Rightarrow \begin{array}{l} -1000_2 \\ \hline 101 \\ -110 \\ \hline 101 \\ \hline 1 \end{array} \left| \begin{array}{l} 101_2 \\ 0,(0011)_2 \end{array} \right. \end{array}$$

Точка в частном означает, что процесс деления не окончен. В дальнейшем осуществлять деление рекомендуется в виде одной записи.

Возвращаясь к десятичной системе счисления в представлении числа

$$\frac{1}{5_{10}} = 0,001100110011\dots_2,$$

получаем разложение числа в ряд в десятичной системе счисления

$$\frac{1}{5} = \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^7} + \frac{1}{2^8} + \frac{1}{2^{11}} + \frac{1}{2^{12}} + \dots$$

Правая часть равенства является суммой членов двух геометрических прогрессий. Проверьте, что их сумма действительно равна левой части равенства.

Если перевести $1/5$ в троичную систему счисления, то аналогично получаем

$$\frac{1}{5_{10}} = \frac{1}{12_3} = 0,(0121)_3.$$

Следовательно,

$$\frac{1}{5} = \frac{1}{3^2} + \frac{2}{3^3} + \frac{1}{3^4} + \frac{1}{3^6} + \frac{2}{3^7} + \frac{1}{3^8} + \frac{1}{3^{10}} + \frac{2}{3^{11}} + \frac{1}{3^{12}} + \dots$$

Проверьте, что сумма членов трех геометрических прогрессий в правой части равенства действительно равна левой части равенства.

Сформирован полезный алгоритм:

Чтобы разложить дробь в ряд, члены которого содержат степени числа p в знаменателях, необходимо выразить эту дробь

в системе счисления с основанием p и затем записать полученную дробь в десятичной системе счисления.

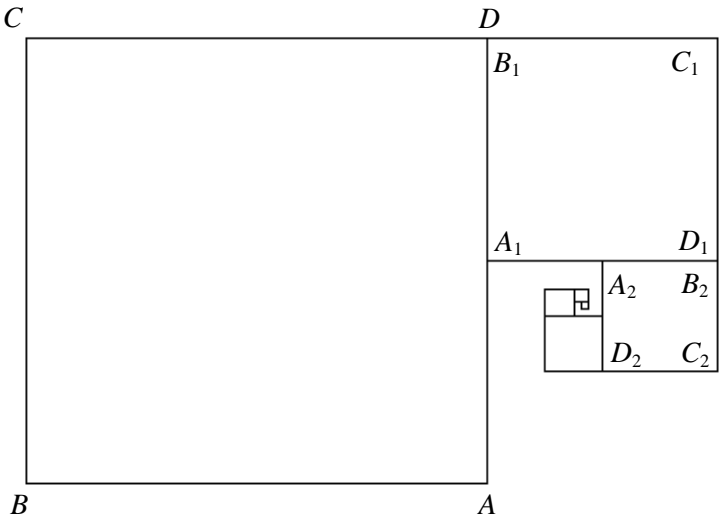


Рис. 5

Рассмотрим фигуру (рис. 5), построенную следующим образом. На стороне AD квадрата $ABCD$ построен квадрат $A_1B_1C_1D_1$, сторона которого вдвое меньше стороны первоначального квадрата. На стороне A_1D_1 снова построен квадрат $A_2B_2C_2D_2$, сторона которого вдвое меньше стороны предыдущего квадрата, и т.д.

Построение последовательности квадратов можно быстро выполнить на компьютере, если использовать цикл для построения последовательности, а каждый квадрат строить одним оператором, используя координаты двух противоположных вершин квадрата.

Анализируем последовательность отрезков $BD, B_1D_1, B_2D_2, \dots$ (рис. 6). Каждые два последовательных отрезка из этого множества перпендикулярны, причем отрезки закручиваются вправо при увеличении номера.

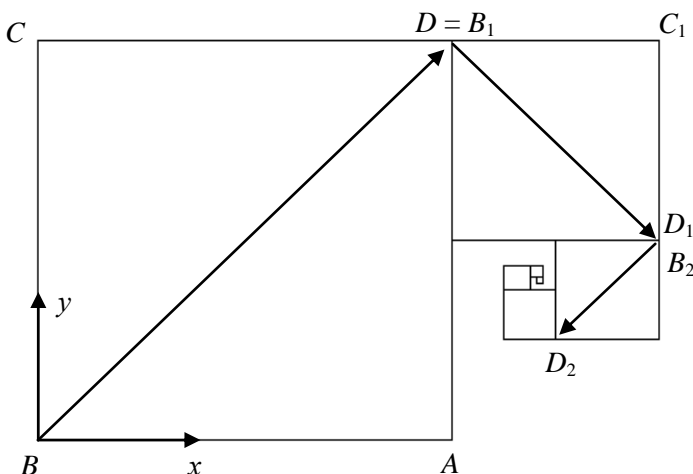


Рис. 6

При повороте на угол (-90°) координаты вектора преобразуются по формулам (рис. 7) $x' = y$, $y' = -x$.

Длина каждого из отрезков BD , B_1D_1 , B_2D_2, \dots вдвое меньше, чем длина предыдущего отрезка, следовательно, координаты полученного вектора необходимо уменьшить вдвое и формулы преобразования векторов, направленных вдоль диагоналей, примут вид:

$$x' = y/2, y' = -x/2.$$

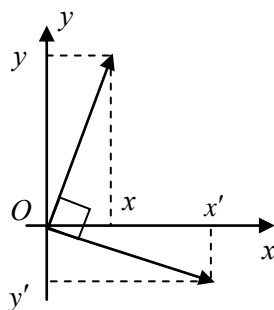


Рис. 7

Если начальная точка вектора имеет координаты $(x_0; y_0)$ и координаты вектора равны $(a; a)$, то конечная точка этого вектора $(x_0 + a; y_0 + a)$ будет начальной точкой следующего вектора, направленного вдоль диагонали. Длина стороны следующего квадрата уменьшается вдвое, а следовательно, и координаты следующего вектора также уменьшаются вдвое.

Программа построения последовательности закрученных квадратов имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 4000 'длина стороны первоначального квадрата
x0 = 100: y0 = 100 'координаты угла начального квадрата
xv = a: yv = a 'координаты вектора на диагонали
For i = 1 To 5
Line (x0, y0)-(x0 + xv, y0 + yv), , B 'построение квадрата
x0 = x0 + xv: y0 = y0 + yv 'переход к начальной точке следующего квадрата
b = xv: xv = yv / 2: yv = -b / 2 'формулы поворота и сжатия вектора
Next i: End Sub
```

Расположение последовательности квадратов имеет форму закрученной спирали. Найдем точку, к которой закручиваются квадраты.

Каждый следующий квадрат в последовательности получается поворотом на (-90^0) . Композиция двух поворотов является симметрией относительно центра поворота. Центр поворота расположен на прямых AA_2 , BB_2 , CC_2 , DD_2 .

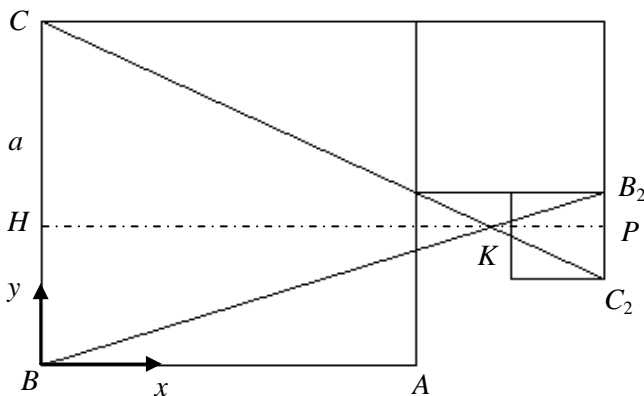


Рис. 8

Пусть прямые BB_2 и CC_2 пересекаются в точке K , тогда треугольники BCK и B_2C_2K подобны (рис. 8). Проведем через точку

K отрезок HP , параллельный отрезку BA , тогда $HP = \frac{3a}{2}$,

$$\frac{HK}{KP} = \frac{BC}{B_2C_2} = 4, \quad HK = \frac{6a}{5}, \quad KP = \frac{3a}{10}, \quad BH = \frac{2a}{5}.$$

Для системы координат с началом в точке B получаем координаты точки $K\left(\frac{6a}{5}; \frac{2a}{5}\right)$.

Логарифмическая спираль с центром в начале координат имеет уравнение $r = r_0 a^\varphi$, где r, φ — полярные координаты произвольной точки спирали. Найдем параметры r_0, a .

При повороте квадрата $A_2B_2C_2D_2$ вокруг центра на 180° с растяжением сторон в четыре раза, получаем квадрат $ABCD$. Точка P при таком преобразовании должна перейти в точку H . Пусть точке P на спирали соответствовал угол φ_P , тогда точке H соответствует угол $\varphi_P + \pi$.

Подставляя координаты этих точек в уравнение спирали, получаем

$$\frac{3a}{10} = r_0 a^{\varphi_P} \text{ — для точки } H, \quad \frac{6a}{5} = r_0 a^{\varphi_P + \pi} \text{ — для точки } P.$$

Разделив эти равенства, получаем $4 = a^\pi, a = 4^{\frac{1}{\pi}}$.

Уравнение спирали частично уточнилось $r = r_0 4^{\frac{\varphi}{\pi}}$.

Точки A, A_1, A_2, A_3, \dots расположены на аналогичной спирали,

т.к.
$$\frac{KA}{KA_1} = \frac{KA_1}{KA_2} = \frac{KA_2}{KA_3} = \dots = 2 \text{ (рис. 9).}$$

Точки B, B_1, B_2, B_3, \dots и точки C, C_1, C_2, C_3, \dots расположены на аналогичных спиралях.

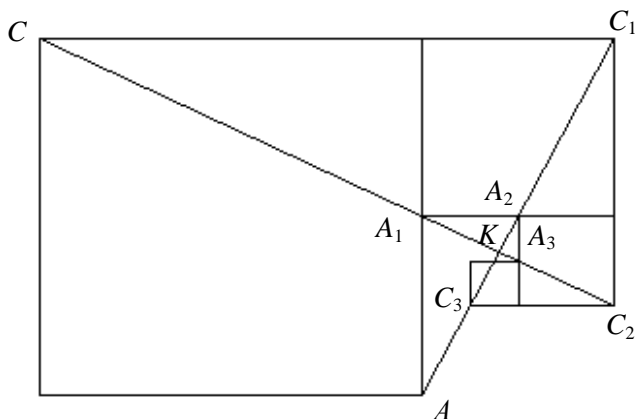


Рис. 9

Чтобы построить спирали, проходящие через последовательности точек A, A_1, A_2, A_3, \dots , B, B_1, B_2, B_3, \dots , C, C_1, C_2, C_3, \dots , для каждой из спиралей необходимо выбрать соответствующее значение множителя r_0 .

Последовательность квадратов совершает бесконечное число оборотов вокруг центра K , но на экране уже на втором обороте вокруг этой точки квадраты трудно различимы визуально.

Будем считать, что первоначальный квадрат расположен на спирали при третьем обороте вокруг центра K .

Для расчета множителя в спирали, проходящей через точки C, C_1, C_2, C_3, \dots получаем $KC = \frac{3a\sqrt{5}}{5}$.

Вектор \overline{KC} образует с направлением оси Ox угол, равный $\pi/2 + \arctg 2$.

Подставляя значения $r = \frac{3a\sqrt{5}}{5}$, $\varphi = 2\pi + \pi/2 + \arctg 2$, получаем $\frac{3a\sqrt{5}}{5} = r_C 4^{\frac{4,5\pi + \arctg 2}{\pi}}$ или $r_C = \frac{3a\sqrt{5}}{5} 4^{-4,5 - \frac{\arctg 2}{\pi}}$.

Аналогично рассчитываются множители r_A, r_B для других спиралей.

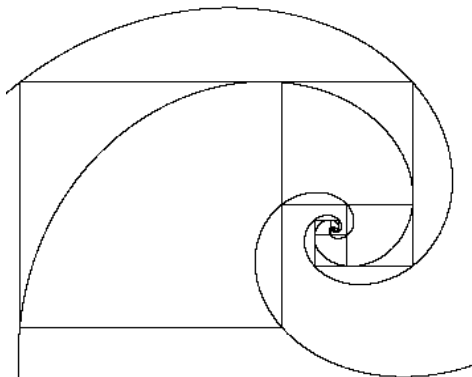


Рис. 10

Программа «**Последовательность квадратов**» построения квадратов и спиралей (рис. 10) имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 3000 'длина стороны первоначального квадрата
x0 = 1000: y0 = 1000: xv = a: yv = a
For i = 1 To 6
Line (x0, y0)-(x0 + xv, y0 + yv), , B
x0 = x0 + xv: y0 = y0 + yv: b = xv: xv = yv / 2: yv = -b / 2
Next i
pi = 4 * Atn(1): rc = 3 * a / (Sqr(5) * 4 ^ (5 / 2 + Atn(2) / pi))
rb = 2 * a * Sqr(10) / (5 * 4 ^ (3 + Atn(1 / 3) / pi))
ra = a / (Sqr(5) * 4 ^ (3 + Atn(2) / pi))
For t = -pi To 4 * pi Step 0.001
xc1 = rc1 * 4 ^ (t / pi) * Cos(t): yc1 = rc1 * 4 ^ (t / pi) * Sin(t)
PSet (xc1 + 6 * a / 5 + 1000, yc1 + 2 * a / 5 + 1000)
xb = rb * 4 ^ (t / pi) * Cos(t): yb = rb * 4 ^ (t / pi) * Sin(t)
PSet (xb + 6 * a / 5 + 1000, yb + 2 * a / 5 + 1000)
xa = ra * 4 ^ (t / pi) * Cos(t): ya = ra * 4 ^ (t / pi) * Sin(t)
PSet (xa + 6 * a / 5 + 1000, ya + 2 * a / 5 + 1000)
Next t
End Sub
```

3.1. Постройте с помощью одного цикла две логарифмические спирали с одним и тем же центром и семейство отрезков, соединяющих точки логарифмических спиралей с одинаковыми значениями параметра (рис. 11). Сравните с программой «**Крыло птицы**».

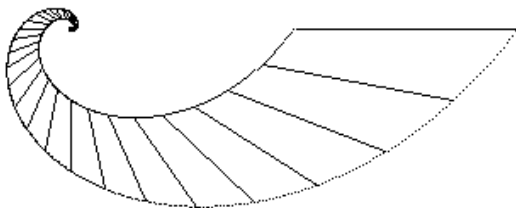


Рис. 11

3.2. Напишите программу построения последовательности квадратов, стороны которых изменяются в k раз (рис. 12).

а) При каком значении k квадраты последовательности начинают накладываться на первоначальный квадрат?

б) Как изменятся уравнения спиралей, проходящих через соответствующие вершины квадратов?

в) В пособии [21, с. 249—262] изучите алогичные спирали, связанные с золотым отношением. При каком значении k получается последовательность квадратов в золотом прямоугольнике?

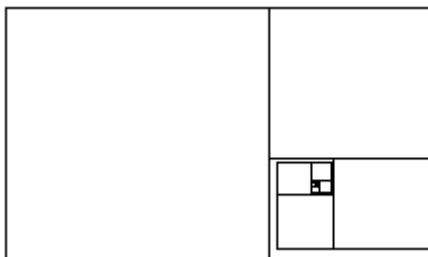


Рис. 12

3.3. Разработайте аналогичный проект для последовательности закрученных правильных треугольников, в которой стороны изменяются в k раз.

§ 20. Окружности Монжа и цветовая композиция от трех источников света

При изучении системы образования цветов RGB в информатике учащиеся не решают задач о наложении одних цветов на другие и образовании новых цветов.

Для этого можно применять шаблоны с расположением источников света различных цветов и препятствий на пути распространения света.

Интересная композиция получается при использовании трех круглых препятствий в виде цилиндрических башен. Сводя задачу к плоскому случаю, можно считать, что заданы три источника света с основными цветами: красный, зеленый и синий, а также заданы три окружности, которые препятствуют распространению света, если какой-то свет падает на них. Будем считать, что тень от каждой окружности полностью покрывает другую окружность. С математической точки зрения будем считать, что во внешних центрах подобия каждой пары окружностей расположены источники света (рис. 1).

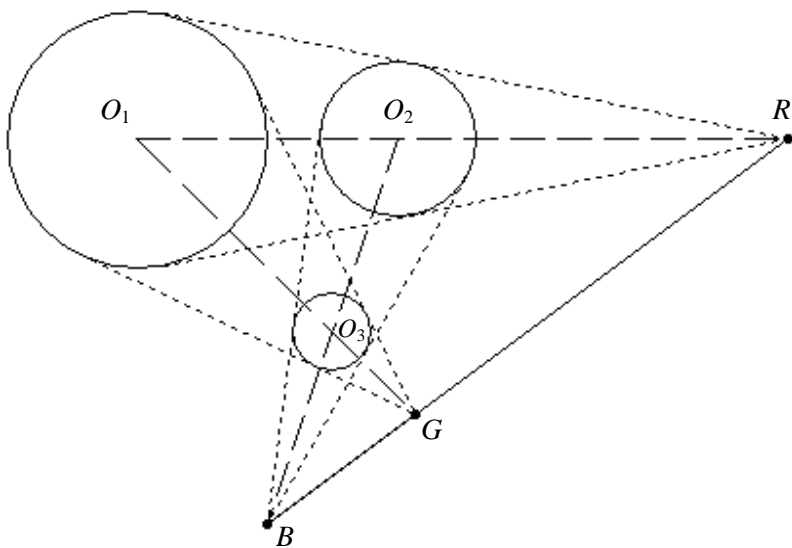


Рис. 1

За основу расположения возьмем известное математическое утверждение о трех окружностях, которые называются *окружностями Монжа* [9, с. 135, 143—145].

Теорема 1. Центры гомотетии трех попарно гомотетичных фигур лежат на одной прямой.

Эта теорема является частным случаем следующей теоремы.

Теорема 2. Произведение двух гомотетий с коэффициентами k_1 и k_2 является гомотетией с коэффициентом k_1k_2 , если $k_1k_2 \neq 1$; параллельным переносом, если $k_1k_2 = 1$, и центральной симметрией, если $k_1k_2 = -1$.

Докажем эту теорему аналитическим способом.

Пусть при гомотетии $H_{M_1}^{k_1}$ с коэффициентом k_1 относительно центра гомотетии $M_1(x_1; y_1)$ произвольная точка $M(x; y)$ отображается в точку $M'(x'; y')$, тогда выполняется равенство

$$\overline{M_1M'} = k_1 \overline{M_1M}.$$

Формулы преобразования координат имеют вид:

$$x' - x_1 = k_1(x - x_1), y' - y_1 = k_1(y - y_1)$$

или

$$x' = k_1x + (1 - k_1)x_1, y' = k_1y + (1 - k_1)y_1.$$

Пусть при гомотетии $H_{M_2}^{k_2}$ с коэффициентом k_2 относительно центра гомотетии $M_2(x_2; y_2)$ точка $M'(x'; y')$ отображается в точку $M''(x''; y'')$, тогда выполняется равенство

$$\overline{M_2M''} = k_2 \overline{M_2M'}.$$

Формулы преобразования координат имеют вид:

$$x'' - x_2 = k_2(x' - x_2), y'' - y_2 = k_2(y' - y_2)$$

или

$$x'' = k_2x' + (1 - k_2)x_2, y'' = k_2y' + (1 - k_2)y_2.$$

Подставляя x', y' , получим

$$\begin{aligned} x'' &= k_2k_1x + k_2(1 - k_1)x_1 + (1 - k_2)x_2, \\ y'' &= k_2k_1y + k_2(1 - k_1)y_1 + (1 - k_2)y_2. \end{aligned} \tag{1}$$

Формулы преобразования координат (1) определяют переход от координат точки $M(x; y)$ к координатам точки $M''(x''; y'')$, т.е. задают композицию отображений $H_{M_2}^{k_2} H_{M_1}^{k_1}$.

Если $k_1 k_2 \neq 1$, то обозначим

$$x_0 = \frac{k_2(1-k_1)x_1 + (1-k_2)x_2}{1-k_1 k_2}, y_0 = \frac{k_2(1-k_1)y_1 + (1-k_2)y_2}{1-k_1 k_2}.$$

Формулы (1) при $k_1 k_2 \neq 1$ можно записать в виде:

$$x'' = k_2 k_1 x + (1 - k_2 k_1) x_0, y'' = k_2 k_1 y + (1 - k_2 k_1) y_0. \quad (2)$$

Формулы (2) определяют гомотетию $H_{M_0}^{k_2 k_1}$ с коэффициентом $k_1 k_2$ и центром $M_0(x_0; y_0)$. При $k_1 k_2 = -1$ гомотетия является центральной симметрией относительно точки

$$x_0 = \frac{(k_2 + 1)x_1 + (1 - k_2)x_2}{2}, y_0 = \frac{(k_2 + 1)y_1 + (1 - k_2)y_2}{2}.$$

Если $k_1 k_2 = 1$, то формулы (1) определяют параллельный перенос. Теорема доказана.

Если рассматриваемые фигуры являются окружностями различных радиусов и для каждой пары окружностей одна лежит вне другой, то точки пересечения общих внешних касательных являются центрами гомотетии этих окружностей. Эти точки расположены на одной прямой, называемой осью подобия окружностей.

Для быстрого изменения расположения центров окружностей и их радиусов, а значит и расположения источников света можно разработать математическую и компьютерную модели данной задачи.

Рассмотрим две вспомогательные задачи.

Задача 1. Требуется найти алгоритм построения касательных, проведенных из точки $M_0(x_0; y_0)$, к окружности ω с центром $O_1(x_1; y_1)$ и радиусом r_1 (рис. 2).

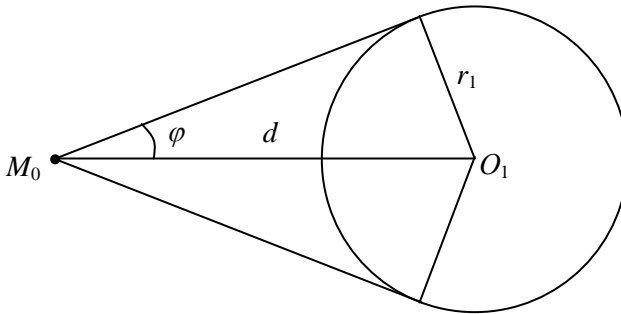


Рис. 2

Пусть при повороте $R_{M_0}^\varphi$ вокруг точки $M_0(x_0; y_0)$ на угол φ точка $M(x; y)$ отображается в точку $M'(x'; y')$, тогда координаты преобразуются по формулам:

$$x' - x_0 = (x - x_0) \cos \varphi - (y - y_0) \sin \varphi,$$

$$y' - y_0 = (x - x_0) \sin \varphi + (y - y_0) \cos \varphi.$$

Пусть угол между прямой M_0O_1 и касательной к окружности равен φ , тогда

$$\cos \varphi = \frac{\sqrt{d^2 - r_1^2}}{d}, \sin \varphi = \frac{r_1}{d}, \text{ где } d = M_0O_1 = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}.$$

При повороте точки $O_1(x_1; y_1)$ вокруг точки $M_0(x_0; y_0)$ на угол φ получим точку на касательной с координатами:

$$x' = x_0 + \frac{\sqrt{d^2 - r_1^2}}{d} (x_1 - x_0) - \frac{r_1}{d} (y_1 - y_0),$$

$$y' = y_0 + \frac{r_1}{d} (x_1 - x_0) + \frac{\sqrt{d^2 - r_1^2}}{d} (y_1 - y_0).$$

Аналогично, если повернем точку $O_1(x_1; y_1)$ вокруг точки $M_0(x_0; y_0)$ на угол $(-\varphi)$, то получим точку на другой касательной с координатами:

$$x' = x_0 + \frac{\sqrt{d^2 - r_1^2}}{d} (x_1 - x_0) + \frac{r_1}{d} (y_1 - y_0),$$

$$y' = y_0 - \frac{r_1}{d} (x_1 - x_0) + \frac{\sqrt{d^2 - r_1^2}}{d} (y_1 - y_0).$$

Чтобы отрезки на касательных были продолжены из точки дальше от точки $M_0(x_0; y_0)$, увеличим в k раз векторы на касательных, тогда получаем следующие точки на касательной с координатами:

$$x' = x_0 + k \left(\frac{\sqrt{d^2 - r_1^2}}{d} (x_1 - x_0) + \frac{r_1}{d} (y_1 - y_0) \right),$$

$$y' = y_0 + k \left(-\frac{r_1}{d} (x_1 - x_0) + \frac{\sqrt{d^2 - r_1^2}}{d} (y_1 - y_0) \right),$$

где коэффициент k выбирается в зависимости от размеров чертежа.

Аналогично

$$x' = x_0 + k \left(\frac{\sqrt{d^2 - r_1^2}}{d} (x_1 - x_0) - \frac{r_1}{d} (y_1 - y_0) \right),$$

$$y' = y_0 + k \left(\frac{r_1}{d} (x_1 - x_0) + \frac{\sqrt{d^2 - r_1^2}}{d} (y_1 - y_0) \right).$$

Для построения касательных нужно провести отрезки с координатами $(x_0; y_0)$ и $(x'; y')$.

Программа построения касательных, написанная на языке Visual Basic, имеет вид:

```
Private Sub Command1_Click()
x0 = 200: y0 = 3000: X1 = 5000: Y1 = 4000: r1 = 2000
d = Sqr((X1 - x0) ^ 2 + (Y1 - y0) ^ 2)
Circle (X1, Y1), r1: Circle (x0, y0), 20
Line (x0, y0)-(X1, Y1)
x = x0 + 2 * (Sqr(d ^ 2 - r1 ^ 2) * (X1 - x0) + r1 * (Y1 - y0)) / d
y = y0 + 2 * (-r1 * (X1 - x0) + Sqr(d ^ 2 - r1 ^ 2) * (Y1 - y0)) / d
Line (x0, y0)-(x, y)
x = x0 + 2 * (Sqr(d ^ 2 - r1 ^ 2) * (X1 - x0) - r1 * (Y1 - y0)) / d
y = y0 + 2 * (r1 * (X1 - x0) + Sqr(d ^ 2 - r1 ^ 2) * (Y1 - y0)) / d
Line (x0, y0)-(x, y)
End Sub
```

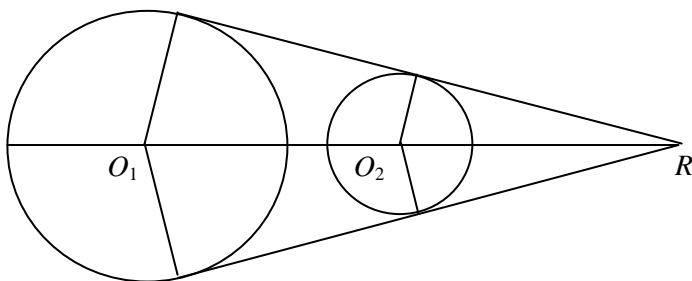


Рис. 3

Задача 2. Даны две окружности с центрами $O_1(x_1; y_1)$ и $O_2(x_2; y_2)$ и с соответствующими радиусами r_1 и r_2 . Требуется определить точку пересечения R общих внешних касательных к окружностям (рис. 3).

Точка пересечения R внешних касательных к окружностям является центром гомотетии этих окружностей. Она делит отрезок O_1O_2 внешним образом в отношении $\lambda = -\frac{r_1}{r_2}$.

Для координат точки, делящей данный отрезок в данном отношении λ , используем формулы:

$$x = \frac{x_1 + \lambda x_2}{1 + \lambda}, y = \frac{y_1 + \lambda y_2}{1 + \lambda},$$

тогда получим координаты для центра гомотетии

$$x_R = \frac{r_1 x_2 - r_2 x_1}{r_1 - r_2}, y_R = \frac{r_1 y_2 - r_2 y_1}{r_1 - r_2}.$$

Аналогично для остальных точек пересечения касательных

$$x_G = \frac{r_1 x_3 - r_3 x_1}{r_1 - r_3}, y_G = \frac{r_1 y_3 - r_3 y_1}{r_1 - r_3}, x_B = \frac{r_2 x_3 - r_3 x_2}{r_2 - r_3}, y_B = \frac{r_2 y_3 - r_3 y_2}{r_2 - r_3}.$$

3.4. Напишите программу построения окружностей Монжа, общих внешних касательных пар окружностей и прямой подобия. Сравните ее с программой «Окружности Монжа».

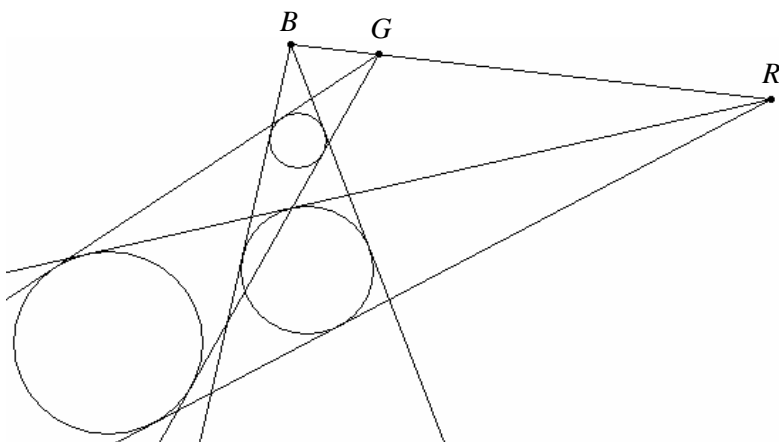


Рис. 4

Данные к рис. 4: $r_1 = 1000$; $r_2 = 700$; $r_3 = 300$; $X_1 = r_1 + 100$;
 $Y_1 = r_1 + 100$; $X_2 = r_1 + 2 * r_2 + 800$; $Y_2 = Y_1 + 800$; $X_3 = X_1 + 2000$; $Y_3 = 3310$

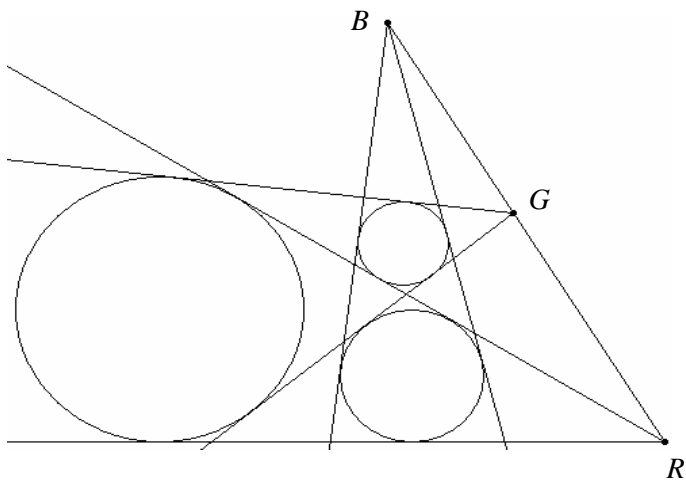


Рис. 5

Данные к рис. 5: $r_1 = 1600$; $r_2 = 800$; $r_3 = 500$; $X_1 = r_1 + 100$;
 $Y_1 = r_1 + 100$; $X_2 = 2800 + X_1$; $Y_2 = Y_1 - 800$; $X_3 = X_1 + 2700$; $Y_3 = 800 + Y_1$

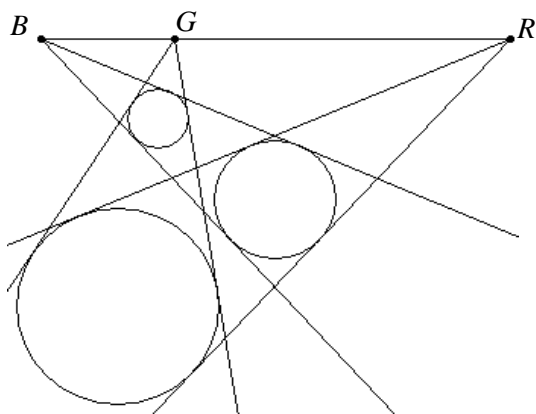


Рис. 6

Данные к рис. 6: $r_1 = 1000$; $r_2 = 600$; $r_3 = 300$; $X_1 = r_1 + 100$; $Y_1 = r_1 + 100$; $X_2 = r_1 + r_2 + 1050$; $Y_2 = Y_1 + 1090$; $X_3 = X_1 + 400$; $Y_3 = 3010$

Закрасьте области на рис. 1, 4—6, используя следующую таблицу смешения цветов. Прямые линии на этих рисунках изображены для лучшего понимания линии цвета. Через прямые свет проходит, а через окружности не проходит.

R-красный	G-зеленый	B-синий	Результирующий
0	0	0	черный
0	0	1	синий
0	1	0	зеленый
0	1	1	голубой
1	0	0	красный
1	0	1	сиреневый
1	1	0	желтый
1	1	1	белый

Замечание 1. Цвет в действительности зависит от пропорции смешивания основных цветов. Составьте новую таблицу и проведите раскраску областей согласно этой таблице.

Замечание 2. Нарисуйте в Word другие рисунки, используя в качестве препятствий прямоугольники, постройте опорные линии

из источников освещения к препятствиям и проведите раскраску полученных областей.

Замечание 3. Используя координаты точек R , G , B , найдите координаты векторов \overline{GR} , \overline{GB} и проверьте их коллинеарность. Это означает, что получено аналитическое доказательство теоремы, т.е. утверждения о том, что эти точки расположены на одной прямой.

Замечание 4. Любые две окружности гомотетичны относительно точки пересечения внутренних общих касательных. Эта точка делит внутренним образом отрезок O_1O_2 в отношении $\lambda = \frac{r_1}{r_2}$. Найдите координаты точек пересечения внутренних касательных и постройте аналогично внутренние касательные к парам окружностей. Любые два внутренних центра подобия и третий внешний центр подобия трех различных пар окружностей снова лежат на одной прямой. Проверьте это утверждение аналитически.

Замечание 5. Мы рассмотрели моделирование, используя в качестве начальных условий задание трех окружностей, т.е. используя параметры x_1 , y_1 , r_1 , x_2 , y_2 , r_2 , x_3 , y_3 , r_3 . Затем определялись точки пересечения внешних касательных, которые расположены на одной прямой.

Рассмотрим моделирование, в основу которого положим выбор центров подобия на заданной прямой l (рис. 7).

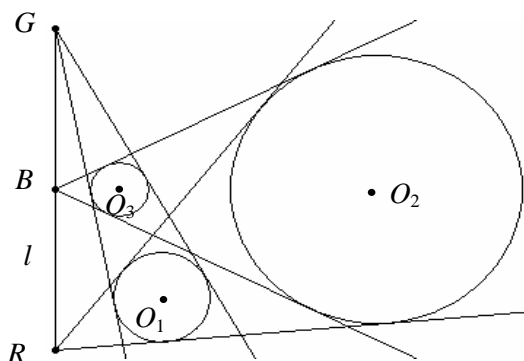


Рис. 7

Пусть прямая линия задана параметрическими уравнениями $x = mt + x_c, y = nt + y_c$, где m, n — координаты направляющего вектора прямой, а x_c, y_c — координаты некоторой точки, через которую проходит прямая.

Для выбора точек R, B на прямой, т.е. будущих центров гомотетии, достаточно зафиксировать параметр t на прямой.

Зададим окружность $\omega_1(O_1, r_1)$ с центром $O_1(x_1, y_1)$ и радиусом r_1 .

Выберем коэффициент k_1 гомотетии $H_R^{k_1}$ с центром в точке R . Применяя гомотетию $H_R^{k_1}$, отобразим окружность $\omega_1(O_1, r_1)$ в окружность $\omega_2(O_2, r_2)$, тогда $r_2 = k_1 r_1$, а для центра второй окружности получаем $x_2 = k_1 x_1 + (1 - k_1) x_R, y_2 = k_1 y_1 + (1 - k_1) y_R$.

Выберем коэффициент k_2 гомотетии $H_B^{k_2}$ с центром в точке B . Применяя гомотетию $H_B^{k_2}$, отобразим окружность $\omega_2(O_2, r_2)$ в окружность $\omega_3(O_3, r_3)$, тогда $r_3 = k_2 r_2$, а для центра второй окружности получаем $x_3 = k_2 x_2 + (1 - k_2) x_B, y_3 = k_2 y_2 + (1 - k_2) y_B$.

Координаты третьего центра гомотетии G определяются с использованием данных задачи для двух окружностей $\omega_3(O_3, r_3)$ и $\omega_1(O_1, r_1)$.

3.5. Напишите программу построения системы окружностей Монжа, их общих внешних касательных для последнего моделирования и сравните ее с программой «Окружности Монжа с данными касат».

§ 21. Многозвенная периодическая траектория математического бильярда в треугольнике

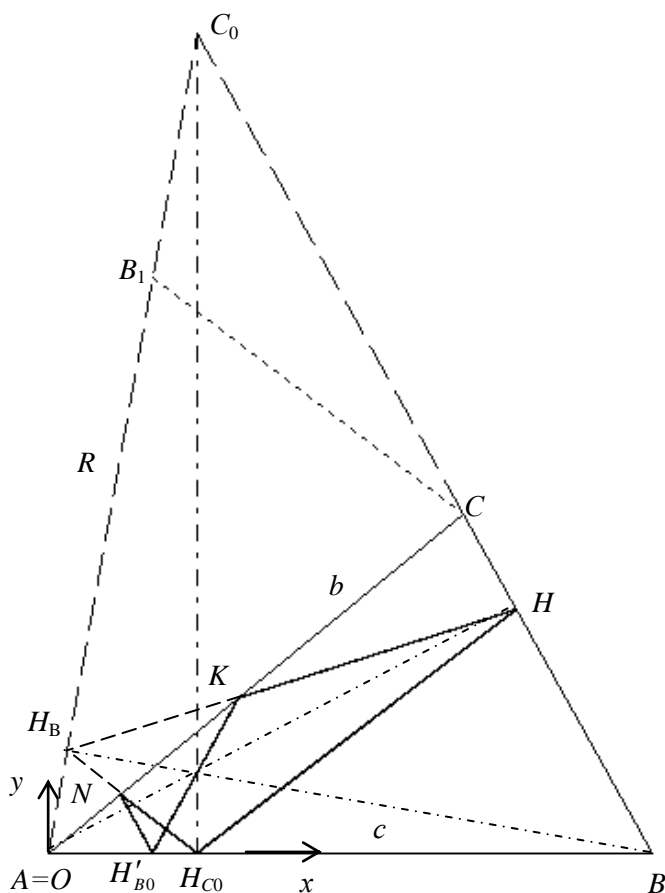


Рис. 1

Трехзвенная периодическая траектория математического бильярда в остроугольном треугольнике проходит через основания высот треугольника [8, 19]. Моделирование такой траектории представлено в [19, с. 51—60]. Периодические траектории с большим числом звеньев изучались для прямоугольника, параллелограмма и куба [21].

Для остроугольного треугольника с углом меньше $\pi/2k$, где k — натуральное число, существует периодическая траектория, состоящая из $2k + 1$ звеньев [7]. Траектория математического бильярда отражается от сторон треугольника по закону упругого удара, т.е. угол падения равен углу отражения.

Для $k = 2$ рассмотрим остроугольный треугольник ABC с элементами $AB = c, AC = b, \angle CAB = u$, где $u < \pi/4$ (рис. 1).

Пусть ACB_1 — треугольник, симметричный к треугольнику ACB относительно прямой AC . Обозначим через C_0 точку пересечения прямых BC и AB_1 . Треугольник ABC_0 — остроугольный, т.к. $\angle BAC_0 = 2u < \pi/2$, $\angle ABC < \pi/2$.

В остроугольном треугольнике существует трехзвенная периодическая траектория $H_{C_0}H_AH_{B_0}$, проходящая через основания высот треугольника ABC_0 . Высота AH_A в треугольнике ABC_0 является высотой данного треугольника ABC , поэтому $H_A \in [BC]$. Точки H_A и H_{B_0} лежат по разные стороны относительно прямой AC , поэтому отрезок $H_AH_{B_0}$ пересекает сторону AC в точке K . Аналогично $H_{C_0}H_{B_0} \cap AC = N$. Отразив часть трехзвенной траектории $KH_{B_0}N$ относительно прямой AC , получим пятизвенную траекторию $H_{C_0}H_AKH'_{B_0}NH_{C_0}$.

Пусть в некоторой прямоугольной декартовой системе координат заданы координаты вершин треугольника: $A(x_1; y_1)$, $B(x_2; y_2)$, $C(x_3; y_3)$.

Движение точки по прямой на экране компьютера будем осуществлять, используя параметрические уравнения прямой $x = x_0 + mt$; $y = y_0 + nt$, где $\vec{p}(m;n)$ — направляющий вектор прямой, $M_0(x_0; y_0)$ — начальная точка траектории.

Найдем правило изменения координат направляющего вектора $\vec{p}(m;n)$ при отражении бильярдной траектории от сторон данного треугольника.

Обозначим

$$e_1 = x_2 - x_1, \quad f_1 = y_2 - y_1,$$

$$e_2 = x_3 - x_2, \quad f_2 = y_3 - y_2,$$

$$e_3 = x_1 - x_3, \quad f_3 = y_1 - y_3.$$

Уравнения сторон l_i треугольника ABC примут вид:

$$l_i : f_i(x - x_i) - e_i(y - y_i) = 0, \quad i = 1, 2, 3.$$

Рассмотрим отражение направляющего вектора от прямой l_i (рис. 2).

Пусть k_i — угловой коэффициент прямой l_i ,

k — угловой коэффициент падающего луча,

k' — угловой коэффициент отраженного луча.

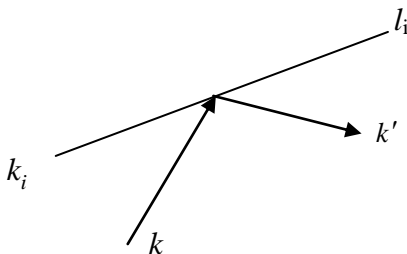


Рис. 2

Из равенства величин ориентированных углов в точке падения получаем

$$\frac{k - k_i}{1 + kk_i} = \frac{k_i - k'}{1 + k_i k'} \quad \text{или} \quad k' = \frac{2k_i - k + kk_i^2}{1 + 2k_i k - k_i^2}.$$

Пусть $k_i = \frac{f_i}{e_i}, k = \frac{n}{m}, k' = \frac{n'}{m'}$, тогда

$$k' = \frac{f_i^2 n - e_i^2 n + 2f_i e_i m}{e_i^2 m - f_i^2 m + 2f_i e_i n}.$$

В качестве направляющего вектора отраженного луча рассмотрим вектор

$$\overline{p'}((e_i^2 - f_i^2)m + 2e_i f_i n; (f_i^2 - e_i^2)n + 2e_i f_i m). \quad (1)$$

Если точка $(x; y)$ после очередного вычисления для бильярдной траектории оказалась вне треугольника, то необходимо сделать шаг назад такого размера, чтобы точка оказалась на границе треугольника, и только потом применить процесс отражения.

Найдем точку пересечения бильярдной траектории со стороной треугольника.

Параметрические уравнения бильярдной траектории:
 $X = x + mt$, $Y = y + nt$, где x, y — координаты последней вычисленной точки траектории.

Параметрические уравнения стороны треугольника:

$$X = x_i + e_i v, Y = y_i + f_i v.$$

Для координат точки пересечения получаем систему

$$\begin{cases} x + mt = x_i + e_i v, \\ y + nt = y_i + f_i v. \end{cases} \text{ или } \begin{cases} mt - e_i v = x_i - x, \\ nt - f_i v = y_i - y. \end{cases}$$

$$t = \frac{e_i(y_i - y) - f_i(x_i - x)}{e_i n - f_i m}. \quad (2)$$

Подставляя значение параметра u , вычисленное по формуле (2), получим точку на границе треугольника. Используя эту точку как начальную и направление траектории (1), получим точку отраженного луча.

Пусть текущая точка $M(x; y)$ траектории и начальная точки $M_0(x_0; y_0)$ траектории оказались по разные стороны относительно стороны $l_i: f_i(x - x_i) - e_i(y - y_i) = 0$ треугольника, тогда выполняется неравенство

$$[f_i(x - x_i) - e_i(y - y_i)][f_i(x_0 - x_i) - e_i(y_0 - y_i)] < 0.$$

Обозначим

$$s_i = f_i(x - x_i) - e_i(y - y_i), \quad s_{i0} = f_i(x_0 - x_i) - e_i(y_0 - y_i).$$

Для проверки того, не вышла ли траектория за пределы данного треугольника, необходимо с помощью условных операторов проверить условия $s_1 s_{10} < 0, s_2 s_{20} < 0, s_3 s_{30} < 0$.

Если выполняется хотя бы одно из этих условий, необходимо текущую точку вернуть назад на границу треугольной области и только потом выполнить изменение направляющего вектора по правилу упругого удара.

Для упрощения вычислений рассмотрим систему координат, начало которой совпадает с вершиной A , а ось абсцисс направлена по лучу AB данного треугольника (рис. 1).

Обозначим длину отрезка AC_0 через R , тогда в выбранной системе координат получаем

$$\begin{aligned}\overline{AC_0}(R \cos 2u; R \sin 2u), \overline{AB}(c; 0), \overline{BC_0} &= \overline{AC_0} - \overline{AB}, \\ \overline{BC_0}(R \cos 2u - c; R \sin 2u), \\ \overline{AC}(bc \cos u; b \sin u), \overline{BC} &= \overline{AC} - \overline{AB}, \overline{BC}(b \cos u - c; b \sin u).\end{aligned}$$

Из коллинеарности векторов $\overline{BC_0}$ и \overline{BC} следует пропорциональность их координат

$$\frac{R \cos 2u - c}{b \cos u - c} = \frac{R \sin 2u}{b \sin u}.$$

Следовательно,

$$R = \frac{cb}{2c \cos u - b}.$$

Для вершин треугольников получаем координаты

$$A(0; 0), B(c; 0), C(b \cos u; b \sin u), C_0(R \cos 2u; R \sin 2u).$$

Найдем координаты направляющего вектора периодической траектории.

Из прямоугольного треугольника ABH_{B_0} следует

$$\overline{AH_{B_0}} = c \cos 2u, \text{ поэтому } \overline{AH_{B_0}}(c \cos^2 2u; c \cos 2u \sin 2u).$$

Из прямоугольного треугольника $AC_0H_{C_0}$ следует

$$\overline{AH_{C_0}} = R \cos 2u, \text{ поэтому } \overline{AH_{C_0}}(R \cos 2u; 0).$$

$$\overline{H_{C_0}H_{B_0}} = \overline{AH_{B_0}} - \overline{AH_{C_0}},$$

$$\overline{H_{C_0}H_{B_0}}(c \cos^2 2u - R \cos 2u; c \cos 2u \sin 2u).$$

В качестве направляющего вектора траектории можно взять вектор $\overline{p}(c \cos 2u - R; c \sin 2u)$, коллинеарный вектору $\overline{H_{C_0}H_{B_0}}$.

В качестве начальной точки траектории можно выбрать точку $H_{C_0}(R \cos 2u; 0)$.

Для наглядности на экране компьютера построены высоты треугольника ABC_0 . Координаты оснований высот найдены следующим образом:

$$\overline{H_{C_0}}(R \cos 2u; 0),$$

$$\overline{AH_{B_0}} = c \cos 2u, \overline{H_{B_0}}(c \cos^2 2u; c \cos 2u \sin 2u),$$

$$\overline{BC_0} = \sqrt{c^2 + R^2 - 2cr \cos 2u},$$

$$\cos \angle B = \frac{BH_{C_0}}{BC_0} = \frac{c - R \cos 2u}{BC_0}, \quad \sin \angle B = \frac{C_0 H_{C_0}}{BC_0} = \frac{R \sin 2u}{BC_0},$$

$$BH_A = c \cos \angle B, \quad H_A (c - BH_A \cos \angle B, BH_A \sin \angle B).$$

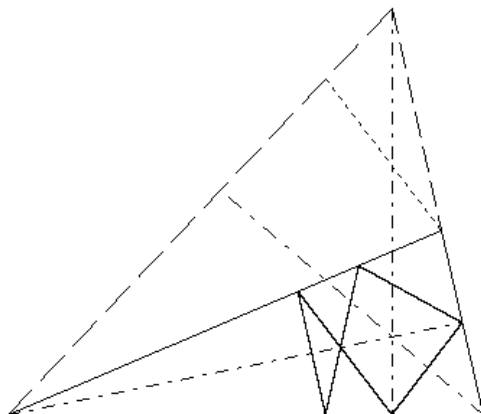


Рис. 3

Если данный треугольник является равнобедренным ($AB = AC$), то периодическая траектория симметрична относительно высоты к основанию треугольника (рис. 3, 4).

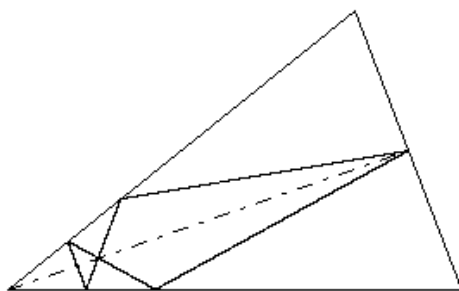


Рис. 4

Если величина угла ACB данного треугольника приближается к $\pi/2$, то длина отрезка $H_A K$ бильярдной периодической траектории стремится к нулю (рис. 5).

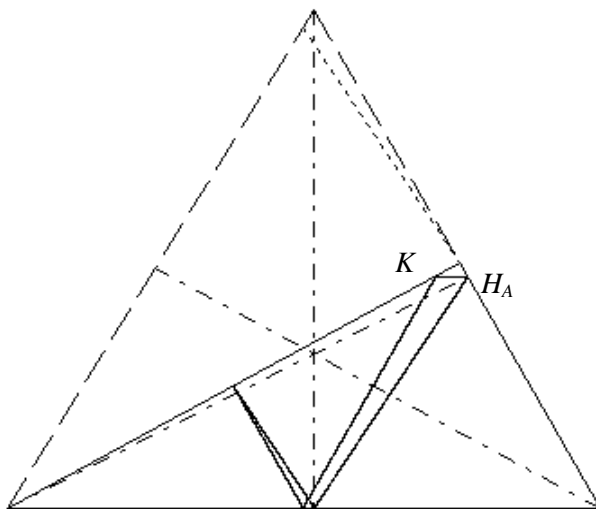


Рис. 5

Если $\angle ACB = \pi/2$, то высота AH_A совпадает со стороной AC треугольника ABC и пятизвенная траектория $H_C H_A K H'_C N H_C$ вырождается в дважды проходимую двухзвенную линию (рис. 6).

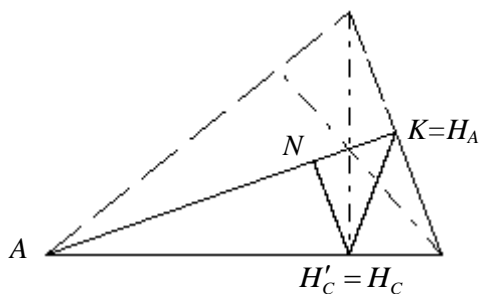


Рис. 6

Программа построения пятизвенной траектории, написанная на языке Visual Basic, имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
b = 4000: c = 4000: u = 30 * Atn(1) / 45
XA = 1000: YA = 1000: XB = XA + c: YB = YA
```

```

XC = XA + b * Cos(u): YC = YA + b * Sin(u)
Line (XA, YA)-(XB, YB): Line (XB, YB)-(XC, YC)
Line (XC, YC)-(XA, YA)
Circle (x0, y0), 20
r = b * c / (2 * c * Cos(u) - b)
m0 = c * Cos(2 * u) - r: n0 = c * Sin(2 * u)
m = m0 / Sqr(m0 ^ 2 + n0 ^ 2): n = n0 / Sqr(m0 ^ 2 + n0 ^ 2)
Xc0 = XA + r * Cos(2 * u): Yc0 = YA + r * Sin(2 * u)
Xb1 = XA + c * Cos(2 * u): Yb1 = YA + c * Sin(2 * u)
Xhc = XA + r * Cos(2 * u): Yhc = YA
Xhb = XA + c * (Cos(2 * u)) ^ 2: Yhb = YA + c * Cos(2 * u) * Sin(2 * u)
BC0 = Sqr(c ^ 2 + r ^ 2 - 2 * c * r * Cos(2 * u))
Bha = c * (c - r * Cos(2 * u)) / BC0
Xha = XA + c - Bha ^ 2 / c: Yha = YA + Bha * Sqr(c ^ 2 - Bha ^ 2) / c
DrawStyle = 1
Line (XA, YA)-(Xc0, Yc0): Line (Xc0, Yc0)-(XC, YC)
DrawStyle = 2
Line (Xb1, Yb1)-(XC, YC)
DrawStyle = 3
Line (Xc0, Yc0)-(Xhc, Yhc): Line (XB, YB)-(Xhb, Yhb)
Line (XA, YA)-(Xha, Yha)
x0 = XA + r * Cos(2 * u) + m: y0 = YA + n
e1 = (XB - XA): f1 = (YB - YA): e2 = (XC - XB): f2 = (YC - YB)
e3 = (XA - XC): f3 = (YA - YC)
x = x0: y = y0
s10 = f1 * (x0 - XA) - e1 * (y0 - YA): s20 = f2 * (x0 - XB) - e2 * (y0 - YB)
s30 = f3 * (x0 - XC) - e3 * (y0 - YC)
For i = 0 To 100000
x = x + m: y = y + n
s1 = f1 * (x - XA) - e1 * (y - YA): s2 = f2 * (x - XB) - e2 * (y - YB)
s3 = f3 * (x - XC) - e3 * (y - YC)
If (s10 * s1) < 0 Then u = (e1 * (YA - y) - f1 * (XA - x)) / (e1 * n - f1 * m):_
x = m * u + x: y = n * u + y: m1 = e1 ^ 2 * m - f1 ^ 2 * m + 2 * e1 * f1 * n:_
n1 = f1 ^ 2 * n - e1 ^ 2 * n + 2 * e1 * f1 * m: r = Sqr(m1 ^ 2 + n1 ^ 2):_
m = m1 / r: n = n1 / r
If (s20 * s2) < 0 Then u = (e2 * (YB - y) - f2 * (XB - x)) / (e2 * n - f2 * m):_
x = m * u + x: y = n * u + y: m1 = e2 ^ 2 * m - f2 ^ 2 * m + 2 * e2 * f2 * n:_
n1 = f2 ^ 2 * n - e2 ^ 2 * n + 2 * e2 * f2 * m: r = Sqr(m1 ^ 2 + n1 ^ 2):_
m = m1 / r: n = n1 / r
If (s30 * s3) < 0 Then u = (e3 * (YC - y) - f3 * (XC - x)) / (e3 * n - f3 * m):_
x = m * u + x: y = n * u + y: m1 = e3 ^ 2 * m - f3 ^ 2 * m + 2 * e3 * f3 * n:_
n1 = f3 ^ 2 * n - e3 ^ 2 * n + 2 * e3 * f3 * m: r = Sqr(m1 ^ 2 + n1 ^ 2):_
m = m1 / r: n = n1 / r
PSet (x, y)
Next i
End Sub

```

Если для введенных параметров b , c , u треугольник ABC не является остроугольным, то траектория не будет периодической. Программу можно дополнить несколькими операторами для вычисления остальных углов данного треугольника и их распечатки на экране компьютера, чтобы контролировать ввод параметров треугольника.

Для произвольного k необходимо внести небольшие изменения.

Из коллинеарности векторов $\overline{BC_0}$ и \overline{BC} следует пропорциональность их координат

$$\frac{R \cos ku - c}{b \cos u - c} = \frac{R \sin ku}{b \sin u}.$$

Следовательно,

$$R = \frac{cb \sin u}{c \sin ku - b \sin(k-1)u}.$$

В качестве направляющего вектора можно взять вектор

$$\overline{p}(c \cos^2 ku - R \cos(ku); c \cos(ku) \sin(ku)).$$

В качестве начальной точки траектории можно выбрать точку $H_{C_0}(R \cos(ku); 0)$.

На рис. 7—11 изображены траектории из $2k+1$ звеньев в одном и том же треугольнике при различных значениях k .

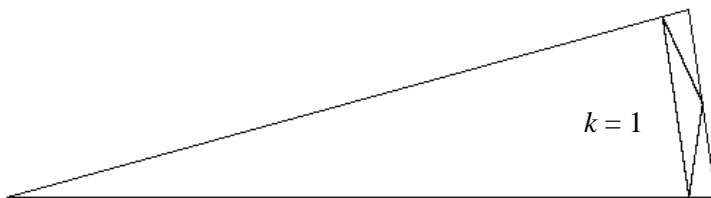


Рис. 7

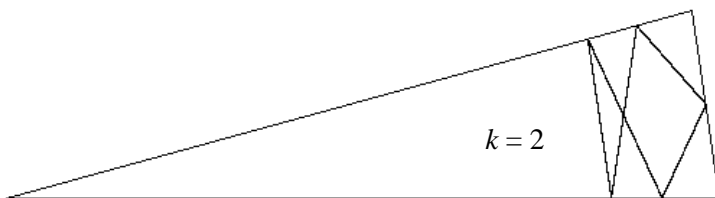


Рис. 8

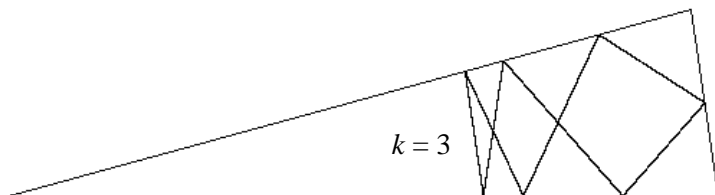


Рис. 9

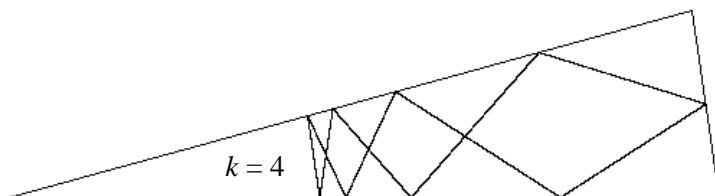


Рис. 10

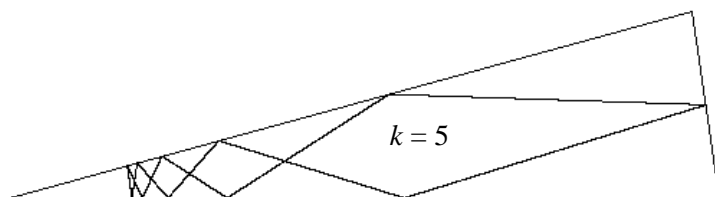


Рис. 11

На рис. 12 эти траектории изображены одновременно и построены все треугольники, получаемые при симметрии. Используя эти вспомогательные треугольники, каждую из периодических траекторий можно «развернуть» до трехзвенной траектории.

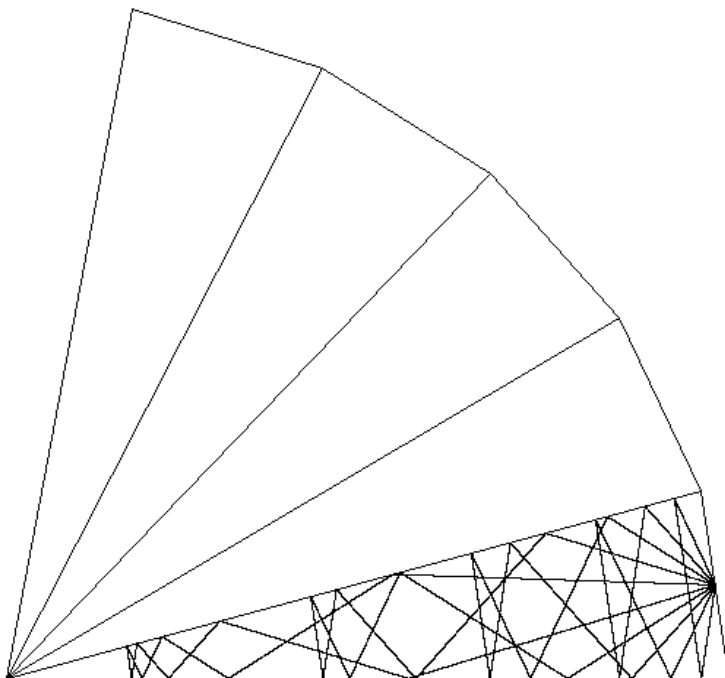


Рис. 12

Напишите программу построения нескольких многозвенных траекторий. Сравните ее с программой «**Бильярд в треугольнике**» построения пяти периодических траекторий в треугольнике.

§ 22. Моделирование замечательных точек T_i треугольника

Существует много замечательных точек для треугольника, но процесс их поиска продолжается и по сегодняшний день. Рассмотрим новую замечательную точку треугольника.

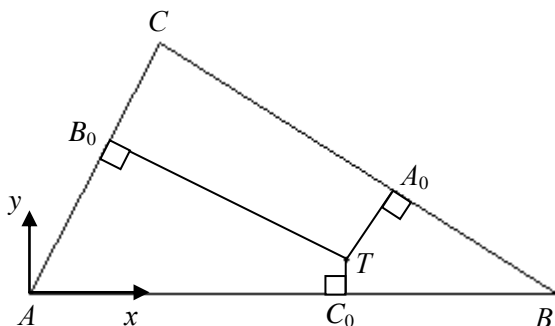


Рис. 1

Задача. Дан треугольник ABC . Найти точку плоскости T , для которой выполняется равенство

$$AB + TC_0 = AC + TB_0 = BC + TA_0, \quad (1)$$

где TC_0 , TB_0 , TA_0 — расстояния от точки T до соответствующих прямых AB , AC , BC .

Если точка T расположена внутри треугольника, то фигура, состоящая из отрезка и перпендикуляра, опущенного на этот отрезок, похожа на букву T . Этим и объясняется название этой замечательной точки треугольника.

Введем следующие обозначения:

$$AB = c, BC = a, AC = b, d = AB + TC_0.$$

Рассмотрим очевидные свойства:

1⁰. Если $a = b = c$, т.е. ABC — равносторонний треугольник, то центр этого треугольника, т.е. точка пересечения медиан треугольника T удовлетворяет условию (1) и

$$d = a + \frac{a\sqrt{3}}{6}.$$

Точка T в этом случае расположена внутри треугольника.

2⁰. Если ABC — равнобедренный треугольник, то точку T будем искать на биссектрисе угла, образованного равными сторонами (рис. 2).

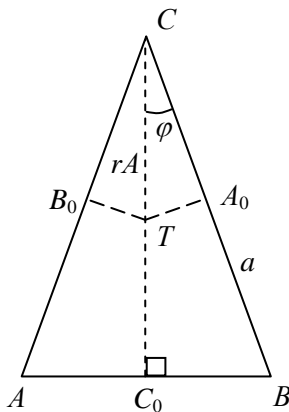


Рис. 2

Выясним расположение точки T в этом частном случае.

Пусть $CA = CB$, $\angle BCC_1 = \varphi$, $CT = r$ (рис. 2), тогда, $TA_0 = r \sin \varphi$, $BC + TA_0 = a + r \sin \varphi$, $CC_0 = a \cos \varphi$, $TC_0 = a \cos \varphi - r$.

Если точка T расположена на отрезке CC_0 , то $AB + TC_0 = 2a \sin \varphi + a \cos \varphi - r$. Из равенства (1) получаем уравнение:

$$r = \frac{a(2 \sin \varphi + \cos \varphi - 1)}{1 + \sin \varphi}.$$

Найдем условие, при котором точка T окажется на стороне AB , т.е. совпадет с точкой C_0 . В этом случае $r = a \cos \varphi$ или

$$\cos \varphi \sin \varphi - 2 \sin \varphi + 1 = 0.$$

Обозначим $\operatorname{tg} \varphi = k$, тогда получаем уравнение:

$$3k^4 - 2k^2 + k^2 - 2k - 1 = 0.$$

На концах отрезка $[0; 2]$ непрерывная функция $f(k) = 3k^4 - 2k^2 + k^2 - 2k - 1$ принимает разные знаки, поэтому на этом отрезке существует решение уравнения.

Методом деления отрезка пополам легко определить приближенное решение этого уравнения $k \approx 1,127$ с заданной точностью p и соответствующее значение угла $\varphi \approx 48,435^\circ$ или $\angle ACB \approx 96,870^\circ$.

Программа решения уравнения, написанная на языке Visual Basic, имеет вид:

```
Private Sub Command1_Click()
a = 0: b = 2: p = 0.0001
Do
c = (a + b) / 2: fa = 3 * a ^ 4 - 2 * a ^ 3 + a ^ 2 - 2 * a - 1
fc = 3 * c ^ 4 - 2 * c ^ 3 + c ^ 2 - 2 * c - 1
If fa * fc < 0 Then b = c Else a = c
Loop While (b - a) / 2 > p
k = (a + b) / 2: u = 45 * Atn(k) / Atn(1)
Print k, 2 * u
End Sub
```

К сожалению, найденная величина угла не является «хорошей» частью прямого угла, т.е. кратной частью прямого угла.

Поиск величины этого угла необходимо было выполнить, но, как иногда случается и в других случаях, рассмотрение частного случая не всегда приводит к общему алгоритму решения задачи.

Рассмотрим моделирование для произвольного треугольника ABC . Пусть система координат на плоскости выбрана согласно рис. 1 и для треугольника заданы: длина стороны $AB = c$, длина стороны $AC = b$ и $\angle BAC = \alpha$.

В компьютерной программе будем использовать следующие элементы треугольника:

$$a = \sqrt{b^2 + c^2 - 2bc \cos \alpha}, \quad \operatorname{tg} \beta = \operatorname{tg} \angle ABC = \frac{b \sin \alpha}{c - b \cos \alpha}.$$

При вычислении расстояний от точки до прямой используются модули некоторых выражений. Для раскрытия каждого из модулей необходимо рассмотреть два случая, а следовательно, условия (1) приводит к необходимости рассмотреть 6 систем.

Очевидно, что необходимы некоторые эвристики, которые сократят число переборов.

Запишем равенство (1) в виде:

$$c + TC_0 = b + TB_0 = a + TA_0. \quad (2)$$

Без ограничения общности будем в дальнейшем считать, что $a \leq c, b \leq c$, тогда из равенства (2) следует $TB_0 \geq c - b, TA_0 \geq c - a$.

Проведем прямые b_1, b_2 , параллельные стороне AC данного треугольника и отстоящие от нее на расстоянии $c - b$ (рис. 3). Точки полосы плоскости, ограниченные этими прямыми, удалены от прямой AC на расстояние не более чем $c - b$.

Проведем прямые a_1, a_2 , параллельные стороне BC данного треугольника и отстоящие от нее на расстоянии $c - a$. Точки полосы плоскости, ограниченные этими прямыми, удалены от прямой BC на расстояние не более чем $c - a$.

Из неравенства $TB_0 \geq c - b, TA_0 \geq c - a$ следует, что искомая точка T должна находиться вне этих полос или на пересечении этих прямых.

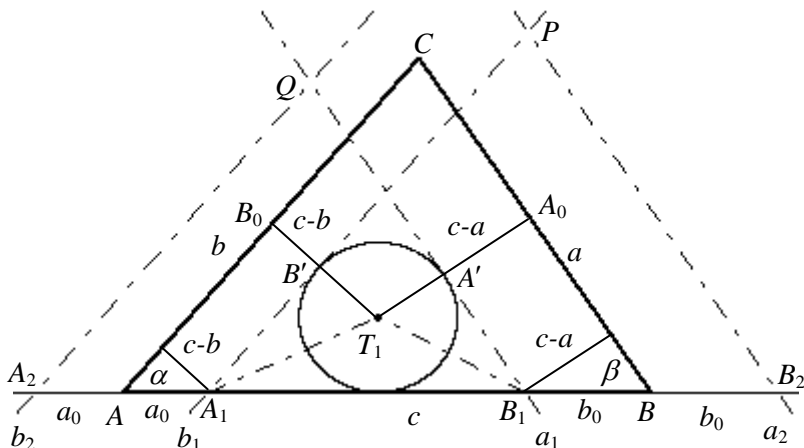


Рис. 3

Из неравенства $TB_0 \geq c - b$ следует, что длину отрезка TB_0 можно представить в виде $c - b + TB'$, где $TB' \geq 0$. Отрезок TB' имеет простой геометрический смысл — это расстояние от точки T до наиболее близкой к ней прямой b_1 или b_2 .

Аналогично, из неравенства $TA_0 \geq c - a$ следует, что длину отрезка TA_0 можно представить в виде $c - a + TA'$, где $TA' \geq 0$. Отрезок TA' имеет простой геометрический смысл — это расстояние от точки T до наиболее близкой к ней прямой a_1 или a_2 .

Равенство (2) примет вид:

$$c + TC_0 = b + (c - b) + TB' = a + (c - a) + TA'$$

или

$$TC_0 = TB' = TA'. \quad (3)$$

Равенства (3) позволяют переформулировать поставленную задачу в следующем виде.

Дан треугольник ABC . Найти точку плоскости T , равноудаленную от прямой AB , прямой b_i и прямой a_j , где каждый из индексов i, j принимает определенное значение из множества $\{1, 2\}$. Точка T является пересечением биссектрис углов, образованных указанными прямыми и, следовательно, центром вписанной окружности.

Пусть прямая b_1 пересекает луч $[AB)$ в точке A_1 , а прямая a_1 пересекает луч $[BA)$ в точке B_1 (рис. 3).

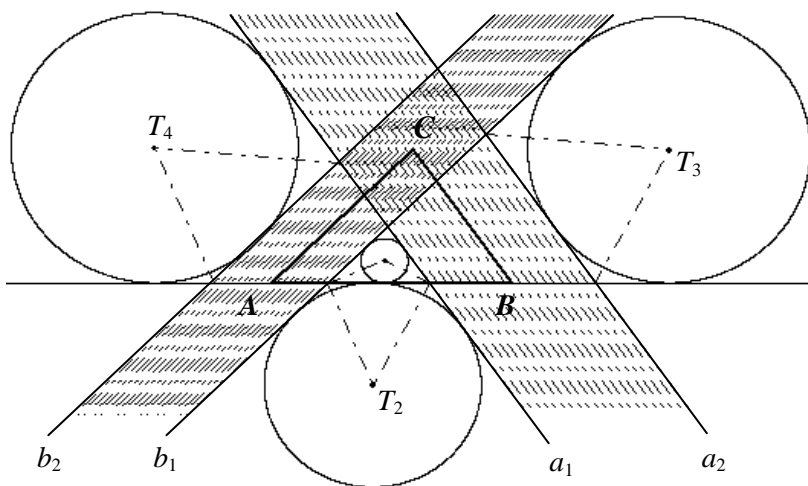


Рис. 4

Если отрезки AA_1 и BB_1 не пересекаются (рис. 3, 4), тогда существует четыре области, ограниченные тройками прямых. На рис. 4 полосы изображены штрихпунктирными линиями. В этом случае существует четыре искомые точки T_1, T_2, T_3, T_4 . Точка T_1 расположена внутри треугольника ABC .

Рассмотрим моделирование этих точек для случая, когда $AA_1 + B_1B < AB$.

Уравнение биссектрисы угла между прямыми A_1B и b_1

$$A_1T_1: y = tg \frac{\alpha}{2} (x - a_0), \text{ где } a_0 = AA_0 = \frac{c-b}{\sin \alpha}.$$

Уравнение биссектрисы угла между прямыми B_1A и a_1

$$B_1T_1: y = -tg \frac{\beta}{2} (x - (c - b_0)), \text{ где } b_0 = BB_0 = \frac{c-a}{\sin \beta}.$$

Для точки T_1 внутри треугольника ABC получаем координаты:

$$x_1 = \frac{a_0 tg(\alpha/2) + (c - b_0) tg(\beta/2)}{tg(\alpha/2) + tg(\beta/2)}, y_1 = tg(\alpha/2) (x_1 - a_0).$$

Аналогично для точки T_2 , в бесконечной области, ограниченной отрезком A_1B_1 и прямыми a_1, b_1 , получаем координаты:

$$x_2 = \frac{a_0 \operatorname{tg}(\alpha/2) + (c - b_0) \operatorname{tg}(\beta/2)}{\operatorname{tg}(\alpha/2) + \operatorname{tg}(\beta/2)}, y_2 = -\operatorname{tg}(\alpha/2) (x_2 - a_0).$$

Для точки T_3 , в бесконечной области, ограниченной отрезком B_2P и прямыми AB, b_1 , получаем координаты:

$$x_3 = \frac{a_0 \operatorname{tg}(\alpha/2) - (c + b_0) \operatorname{ctg}(\beta/2)}{\operatorname{tg}(\alpha/2) - \operatorname{ctg}(\beta/2)}, y_3 = \operatorname{tg}(\alpha/2) (x_3 - a_0).$$

Для наглядности изображения биссектрисы определяем координаты точки $P = b_1 \cap a_2$ из системы

$$\begin{cases} y = \operatorname{tg} \alpha (x - a_0), \\ y = -\operatorname{tg} \beta (x - (c + b_0)), \end{cases}$$

тогда $x_p = \frac{a_0 \operatorname{tg} \alpha + (c + b_0) \operatorname{tg} \beta}{\operatorname{tg} \alpha + \operatorname{tg} \beta}, y_p = \operatorname{tg} \alpha (x_p - a_0).$

Для точки T_4 , в бесконечной области, ограниченной отрезком A_2Q и прямыми AB, a_1 , получаем координаты:

$$x_4 = \frac{a_0 \operatorname{ctg}(\alpha/2) + (c - b_0) \operatorname{tg}(\beta/2)}{-\operatorname{ctg}(\alpha/2) + \operatorname{tg}(\beta/2)}, y_4 = -\operatorname{ctg}(\alpha/2) (x_4 + a_0).$$

Для наглядности изображения биссектрисы определяем координаты точки $Q = a_1 \cap b_2$ из системы

$$\begin{cases} y = \operatorname{tg} \alpha (x + a_0), \\ y = -\operatorname{tg} \beta (x - (c - b_0)). \end{cases}$$

Тогда $x_q = \frac{-a_0 \operatorname{tg} \alpha + (c - b_0) \operatorname{tg} \beta}{\operatorname{tg} \alpha + \operatorname{tg} \beta}, y_q = \operatorname{tg} \alpha (x_q + a_0).$

Следующая программа, написанная на языке Visual Basic, осуществляет построение данного треугольника ABC , вспомогательных прямых a_1, a_2, b_1, b_2 , искомых точек T_1, T_2, T_3, T_4 , окружностей с центрами в этих точках. Программа вычисляет суммы $AB + T_i C_0, AC + T_i B_0, BC + T_i A_0$ от каждой точки T_j и печатает эти значения.

В программе «Точки T_i треугольника» рассматривается случай, когда $a \leq c$, $b \leq c$ и отрезки AA_1 и BB_1 не пересекаются, т.е.

выполняется условие $\frac{c-b}{\sin \alpha} + \frac{c-a}{\sin \beta} < c$.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): x0 = 3400: y0 = 3000: c = 2400: b = 2000: al = 60 * Atn(1) / 45
a = Sqr(b ^ 2 + c ^ 2 - 2 * b * c * Cos(al)): bet = Atn(b * Sin(al) / (c - b * Cos(al)))
xa = 0: ya = 0: xb = c: yb = 0: xc = b * Cos(al): yc = b * Sin(al)
Line (0, y0)-(10000, y0)
DrawStyle = 3
a0 = (c - b) / Sin(al)
For i = -1 To 1 Step 0.1
Line (a0 * i + 2 * b * Cos(al) + x0, 2 * b * Sin(al) + y0) - (a0 * i - b * Cos(al) + x0, -b * Sin(al) + y0)
Next i
b0 = (c - a) / Sin(bet)
For i = -1 To 1 Step 0.1
Line (b0 * i + 2 * b * Cos(al) - c + x0, 2 * b * Sin(al) + y0) - (b0 * i + 2 * c - b * Cos(al) + x0, -b * Sin(al) + y0) 'B1B2
Next i
x = ((c - b0) * Tan(bet / 2) + a0 * Tan(al / 2)) / (Tan(al / 2) + Tan(bet / 2))
y = Tan(al / 2) * (x - a0)
Line (a0 + x0, y0) - (x + x0, y + y0)
Line (c - b0 + x0, y0) - (x + x0, y + y0)
dc = c + Abs(y): db = b + Abs(Sin(al) * x - Cos(al) * y)
da = a + Abs(b * Sin(al) * x + (c - b * Cos(al)) * y - b * c * Sin(al)) / a
FillStyle = 0: Circle (x + x0, y + y0), 20
For t = 0 To 8 * Atn(1) Step 0.001
u = x + Abs(y) * Cos(t): v = y + Abs(y) * Sin(t)
PSet (u + x0, v + y0)
Next t
PSet (100, 500), QBColor(15)
x = ((c - b0) / Tan(bet / 2) + a0 / Tan(al / 2)) / (1 / Tan(al / 2) + 1 / Tan(bet / 2))
y = -(x - a0) / Tan(al / 2)
Line (a0 + x0, y0) - (x + x0, y + y0): Line (c - b0 + x0, y0) - (x + x0, y + y0)
FillStyle = 0
Circle (x + x0, y + y0), 20
dc = c + Abs(y): db = b + Abs(Sin(al) * x - Cos(al) * y)
da = a + Abs(b * Sin(al) * x + (c - b * Cos(al)) * y - b * c * Sin(al)) / a
For t = 0 To 8 * Atn(1) Step 0.001
u = x + Abs(y) * Cos(t): v = y + Abs(y) * Sin(t)
PSet (u + x0, v + y0)
Next t
PSet (100, 1000), QBColor(15)
```

```

Xp = (Tan(al) * a0 + Tan(bet) * (c + b0)) / (Tan(al) + Tan(bet))
Yp = Tan(al) * (Xp - a0)
x3 = (a0 * Tan(al / 2) - (c + b0) / Tan(bet / 2)) / (Tan(al / 2) - 1 / Tan(bet / 2))
y3 = Tan(al / 2) * (x3 - a0)
FillStyle = 0: Circle (x3 + x0, y3 + y0), 20
Line (Xp + x0, Yp + y0)-(x3 + x0, y3 + y0)
Line (c + b0 + x0, y0)-(x3 + x0, y3 + y0)
dc = c + Abs(y3): db = b + Abs(Sin(al) * x3 - Cos(al) * y3)
da = a + Abs(Sin(bet) * x3 + Cos(bet) * y3 - Sin(bet) * c)
For t = 0 To 8 * Atn(1) Step 0.001
u = x3 + Abs(y3) * Cos(t): v = y3 + Abs(y3) * Sin(t)
PSet (u + x0, v + y0)
Next t
PSet (100, 1500), QBColor(15)
Print da; db; dc
Xp = (-Tan(al) * a0 + Tan(bet) * (c - b0)) / (Tan(al) + Tan(bet))
Yp = Tan(al) * (Xp + a0)
x4 = (a0 / Tan(al / 2) + (c - b0) * Tan(bet / 2)) / (Tan(bet / 2) - 1 / Tan(al / 2))
y4 = -(x4 + a0) / Tan(al / 2)
FillStyle = 0: Circle (x4 + x0, y4 + y0), 20
Line (Xp + x0, Yp + y0)-(x4 + x0, y4 + y0): Line (-a0 + x0, y0)-(x4 + x0, y4 + y0)
dc = c + Abs(y4): db = b + Abs(Sin(al) * x4 - Cos(al) * y4)
da = a + Abs(Sin(bet) * x4 + Cos(bet) * y4 - Sin(bet) * c)
For t = 0 To 8 * Atn(1) Step 0.001
u = x4 + Abs(y4) * Cos(t): v = y4 + Abs(y4) * Sin(t)
PSet (u + x0, v + y0)
Next t
PSet (100, 2000), QBColor(15)
Print da; db; dc
DrawWidth = 2
Line (xa + x0, ya + y0)-(xb + x0, yb + y0) 'AB
Line (xb + x0, yb + y0)-(xc + x0, yc + y0) 'BC
Line (xc + x0, yc + y0)-(xa + x0, ya + y0) 'CA
End Sub

```

Если отрезки AA_1 и BB_1 имеют одну общую точку, т.е. выполняется условие $\frac{c-b}{\sin \alpha} + \frac{c-a}{\sin \beta} = c$, то получаем три точки $T_1 = T_2$, T_3 , T_4 (рис. 5, 6). Точка T_1 в этом случае расположена на стороне AB данного треугольника. Рисунок 5 построен для равнобедренного треугольника и угла $\varphi \approx 48,435^\circ$, вычисленного выше с помощью программы.

Если отрезки AA_1 и BB_1 пересекаются по отрезку, т.е. выполняется условие $\frac{c-b}{\sin \alpha} + \frac{c-a}{\sin \beta} > c$, то получаем две точки T_3, T_4 (рис. 7).

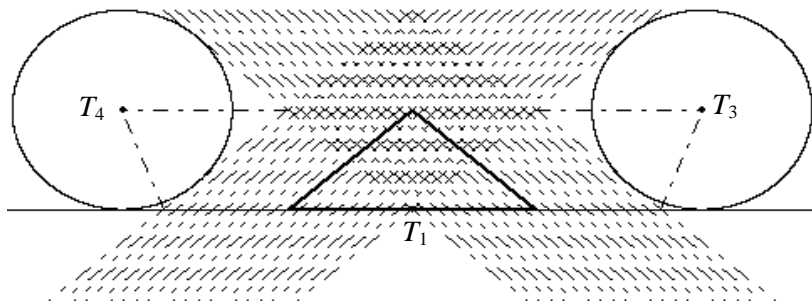


Рис. 5

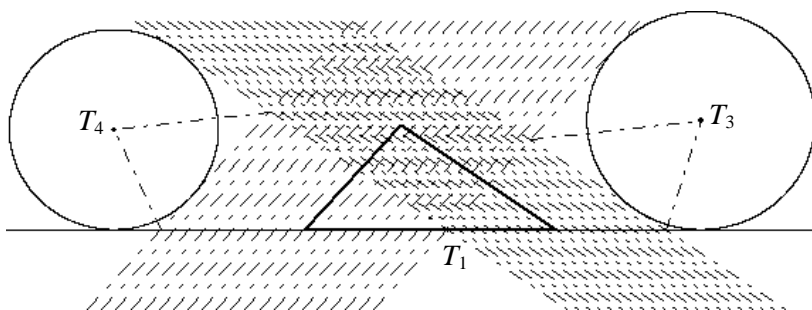


Рис. 6

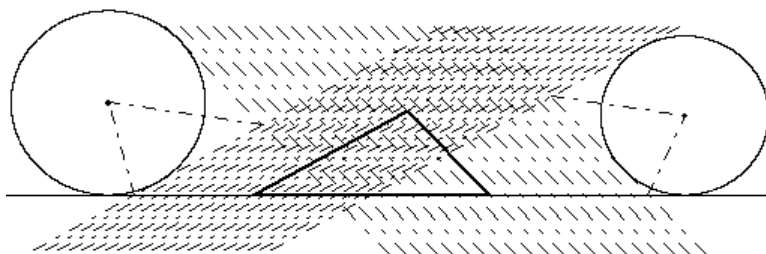


Рис. 7

В программе в этом частном случае нужно удалить два блока для построения точек T_1 и T_2 .

Рассмотрим различные вариации для моделирования с использованием формулы

$$\rho(M_0, l) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

для вычисления расстояния от точки $M_0(x_0; y_0)$ до прямой l , заданной уравнением $ax + by + c = 0$.

А) моделирование для произвольного треугольника ABC , если заданы: длина стороны $AB = c$ и координаты вершины $C(m; h)$, где $h > 0$, тогда $A(0; 0), B(c; 0)$.

Обозначим координаты искомой точки T через x, y , тогда $AB + TC_0 = c + |y|$.

Прямая AC имеет уравнение $hx - my = 0$, поэтому

$$AC + TB_0 = \sqrt{m^2 + h^2} + \frac{|hx - my|}{\sqrt{m^2 + h^2}}.$$

Прямая BC имеет уравнение $hx + (c - m)y - ch = 0$, поэтому

$$BC + TA_0 = \sqrt{(c - m)^2 + h^2} + \frac{|hx + (c - m)y - ch|}{\sqrt{(c - m)^2 + h^2}}.$$

Координаты точки T удовлетворяют системе уравнений

$$c + |y| = \sqrt{m^2 + h^2} + \frac{|hx - my|}{\sqrt{m^2 + h^2}} = \sqrt{(c - m)^2 + h^2} + \frac{|hx + (c - m)y - ch|}{\sqrt{(c - m)^2 + h^2}}. \quad (4)$$

А₁) Для точки T_1 , расположенной внутри треугольника ABC , получаем

$$y > 0, hx - my > 0, hx - (c - m)y - ch < 0.$$

Координаты точки $T_1(x_1; y_1)$ найдем из системы

$$\left\{ \begin{array}{l} c + y = \frac{m^2 + h^2 + hx - my}{\sqrt{m^2 + h^2}} \\ c + y = \frac{(c - m)^2 + h^2 - hx + (c - m)y + ch}{\sqrt{(c - m)^2 + h^2}} \end{array} \right., \text{ тогда}$$

$$y_1 = \frac{2m^2 + 2h^2 + c^2 + ch - 2cm - c(\sqrt{m^2 + h^2} + \sqrt{(c-m)^2 + h^2})}{\sqrt{m^2 + h^2} + \sqrt{(c-m)^2 + h^2} + c},$$

$$x_1 = \frac{(c + y_1)\sqrt{m^2 + h^2} + my_1 - m^2 - h^2}{h}.$$

Аналогично рассматривается упрощение уравнения (4) в остальных случаях.

Б) Моделирование для треугольника с элементами $AB = c$, $AC = b$ и $\angle BAC = \alpha$.

Прямая AC имеет уравнение $\sin \alpha x - \cos \alpha y = 0$, поэтому

$$AC + TB_0 = b + \sin \alpha x - \cos \alpha y.$$

Прямая BC имеет уравнение $\sin \beta x + \cos \beta y - c \sin \beta = 0$, поэтому

$$BC + TA_0 = a - \sin \beta x - \cos \beta y + c \sin \beta.$$

Б₁) Для точки $T_1(x_1; y_1)$, расположенной внутри треугольника ABC , получаем

$$y_1 = \frac{b \sin \beta + a \sin \alpha + c(\sin \alpha \sin \beta - \sin \alpha - \sin \beta)}{\sin(\alpha + \beta) + \sin \alpha + \sin \beta},$$

$$x_1 = \frac{c - b + y_1 + \cos \alpha y_1}{\sin \alpha}.$$

Аналогично рассматривается упрощение уравнения в остальных случаях.

§ 23. Метрика на поверхности и кратчайшие линии на кубе

Задачи о поиске кратчайшей линии на поверхности часто встречаются среди олимпиадных задач и являются предметом изучения в физико-математических школах. Алгоритм нахождения кратчайшей линии на многогранниках сводится к поиску кратчайшей линии на развертке многогранника, причем приходится рассматривать несколько разверток некоторых граней многогранника. На каждой развертке нужно определить кратчайшие

линии, т.е. отрезки, а затем выбрать наименьший из найденных отрезков и построить соответствующую кратчайшую линию на многограннике.

Обучение можно наполнить наглядностью, если изобразить на поверхности куба несколько пар точек, между которыми нужно найти кратчайшие маршруты. Компьютерное моделирование позволяет красочно изобразить соответствующие отрезки на каждой развертке, быстро сравнить их длины и построить кратчайшую линию на поверхности.

Для изучения темы в элективном курсе достаточно знать уравнение прямой, проходящей через две точки, формулы преобразования координат точки при повороте и уметь оперировать координатами в различных системах координат.

В астрофизике известно явление черной дыры, притягивающей планеты небольшой массы к звезде с гигантской массой.

Рассмотрим некоторый аналог в математике, являющийся антиподом указанного явления. На рис. 1 построено семейство кратчайших линий на кубе. Точка M движется в передней грани по параболе, а точка N движется в верхней грани куба по прямой. При движении точек M и N в правую сторону наступает момент, когда кратчайшие линии перескакивают через правый верхний угол куба. Аналогично при движении точек M и N в левую сторону наступает момент, когда кратчайшие линии перескакивают левый верхний угол куба.

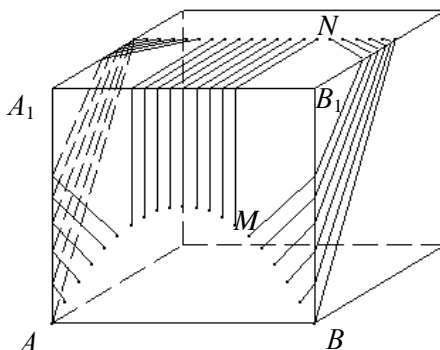


Рис. 1

Вокруг вершин A_1 и B_1 образуются области, не покрытые кратчайшими линиями. Если число кратчайших линий значительно увеличить, то указанные области окажутся незакрашенными, т.е. белыми пятнами на кубе (рис. 2). Для экономных перевозок, т.е. удовлетворяющих условию минимальности длины маршрута, эти области окажутся закрытыми для путешествий.

Среди всех линий, соединяющих две фиксированные точки A и B , выберем линию наименьшей длины, если такая линия существует, и назовем длину этой кратчайшей линии расстоянием между точками A и B на поверхности или метрикой на поверхности.

На плоскости расстояние между двумя данными точками реализуется на отрезке AB . На сфере расстояние между точками реализуется на кратчайшей линии, соединяющей эти точки, т.е. на меньшей дуге большой окружности, проходящей через эти точки. На цилиндре кратчайшей линией является винтовая линия или меньшая дуга окружности, перпендикулярная оси цилиндра, или образующая цилиндра, т.е. прямая, параллельная оси цилиндра.



Рис. 2

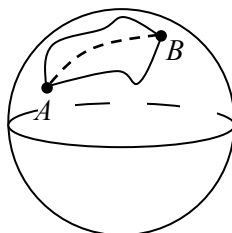


Рис. 3

Существуют поверхности, на которых линия наименьшей длины не существует для двух данных точек поверхности. Например, если из сферы удалить дугу AB большой окружности, оставив концевые точки A и B (рис. 3), то кратчайшей линией является дуга AB , но она не принадлежит поверхности.

Расстоянием между точками A и B на поверхности F называется число $d(A, B) = \inf_{\gamma \in F} s(\gamma(A, B))$, где $\gamma(A, B)$ — произвольная линия с концевыми точками A и B на поверхности, а символ $s(\gamma(A, B))$ означает длину линии.

Если существует линия, где этот \inf достигается, то можно использовать \min .

В дальнейшем будем предполагать, что среди всех линий, соединяющих две точки поверхности, существует линия наименьшей

длины с концами в этих точках, поэтому будем использовать определение

$$d(A, B) = \min_{\forall \gamma \in F} s(\gamma(A, B)). \quad (1)$$

Понятие расстояния между точками на поверхности удовлетворяет аксиомам метрического пространства, поэтому оно по праву называется также метрикой.

Проверка двух первых аксиом метрического пространства очевидна:

- 1) $\rho(x, y) \geq 0, \forall x, y \in X; \rho(x, y) = 0 \Leftrightarrow x = y,$
- 2) $\rho(x, y) = \rho(y, x),$
- 3) $\rho(x, y) + \rho(y, z) \geq \rho(x, z).$

Проверим выполнение третьей аксиомы.

Пусть x, y, z — произвольные точки на поверхности. Пусть для точек x и y существует кратчайшая линия γ_1 , соединяющая эти точки, а для точек y и z — кратчайшая линия γ_2 .

Среди всех линий, соединяющих точки x и z , существует линия xuz , состоящая из объединения линий γ_1 и γ_2 . Если она является кратчайшей линией, то $\rho(x, y) + \rho(y, z) = \rho(x, z)$ (рис. 4).

Если линия xuz , состоящая из объединения линий γ_1 и γ_2 , не является кратчайшей (рис. 5), то $\rho(x, y) + \rho(y, z) > \rho(x, z)$. Аксиома выполняется для введенного расстояния.

Изучим кратчайшие линии на поверхности куба.

Если точки M и N расположены на одной грани куба, то кратчайшей линией, соединяющей эти точки, является отрезок MN .

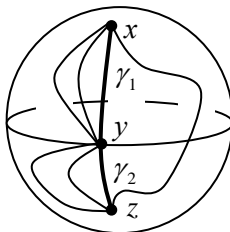


Рис. 4

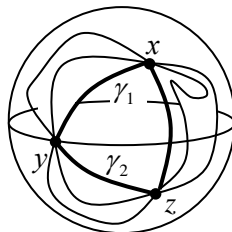


Рис. 5

Пусть точки M и N расположены на смежных гранях, имеющих общее ребро. В дальнейшем будем считать, что точка M расположена на передней грани куба, точка N расположена на верхней грани куба, а ребро куба равно $2a$.

Чтобы разработать математическое и компьютерное моделирование кратчайших линий с концевыми точками на двух смежных гранях куба, нужно понять, какие маршруты могут соединять эти точки, т.е. схему пересечения ребер куба.

Диагональная плоскость ABC_1D_1 разбивает поверхность куба на две симметричные части. Одну половину поверхности куба, содержащую данные точки M и N , назовем положительной, а другую отрицательной. Если ломаная линия с концевыми точками M и N заходит в отрицательную область, то ее можно укоротить так, чтобы ее длина была меньше длины предъявленной линии и укороченная линия находилась в положительной области.

На рис. 6 изображен плоский граф куба, т.е. два concentric-квадрата и четыре отрезка, соединяющие соответствующие точки. Две точки M и N на смежных гранях можно соединить тремя различными маршрутами, которые не пересекают контур ABC_1D_1 .

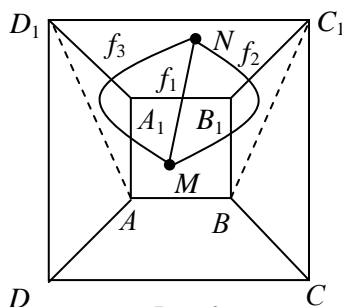


Рис. 6

Таким образом, достаточно рассмотреть ломаные линии с концевыми точками M и N , расположенные в передней, верхней, левой боковой и правой боковой гранях куба.

Повернем верхнюю грань $A_1B_1C_1D_1$ вокруг прямой A_1B_1 так, чтобы она оказалась в плоскости передней грани и эти два квадрата образовывали прямоугольник со сторонами $2a$ и $4a$. На развертке получим прямоугольник ABC_1D_1 (рис. 7).

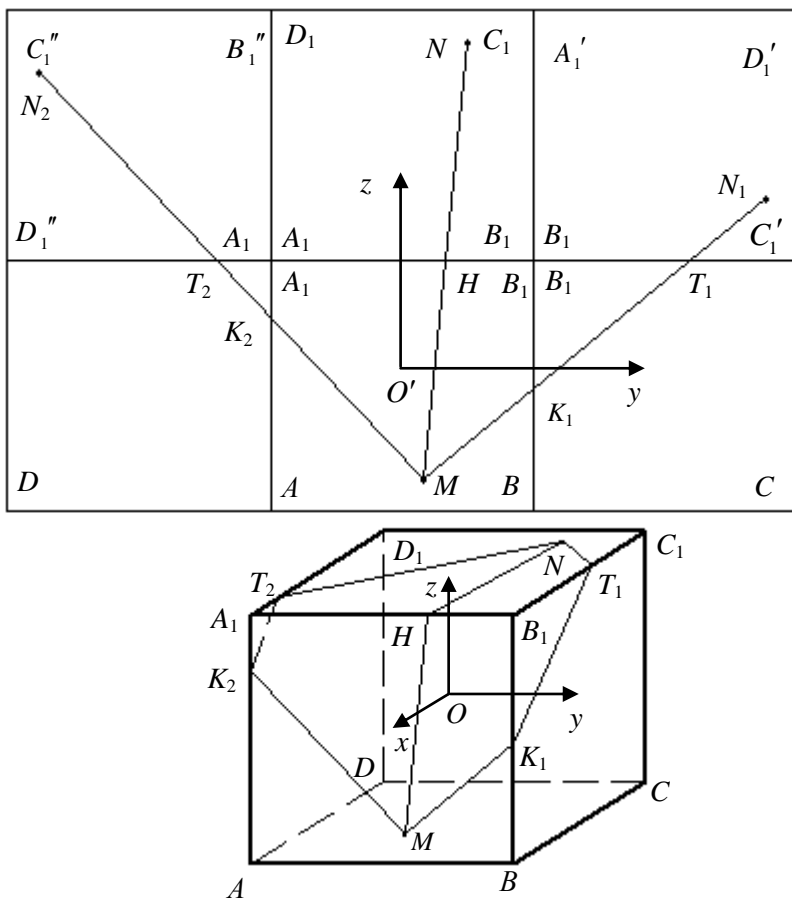


Рис. 7

Длина пространственной линии с концевыми точками M и N по поверхности куба равна длине линии на развертке. Кратчайшей линией на развертке, соединяющей точки M и N , является отрезок MN . Пусть отрезок MN пересекает отрезок A_1B_1 на развертке в точке H , тогда ломаная MHN на поверхности куба будет первым претендентом на кратчайшую линию, соединяющую точки M и N по поверхности куба.

Повернем верхнюю грань куба $B_1C_1D_1A_1$ вокруг прямой B_1C_1 так, чтобы она оказалась в плоскости правой боковой грани и эти два квадрата образовывали прямоугольник со сторонами $2a$ и $4a$.

Затем полученный прямоугольник повернем вокруг прямой BB_1 так, чтобы он оказался в плоскости передней грани. Получим развертку верхней, правой боковой и передней граней, т.е. ступенчатую фигуру $ABCC_1'D_1'A_1B_1A_1A$. Если от развертки вернуться к поверхности куба, то склеиваются следующие точки: A_1' и A_1 , D_1' и D_1 , C_1' и C_1 . Точка N на верхней грани куба при таком развороте окажется в положении N_1 . Кратчайшей линией на развертке в этом случае является отрезок MN_1 .

Возвращаясь от развертки к поверхности куба, получаем ломаную линию MK_1T_1N , которая является вторым претендентом на кратчайшую линию, соединяющую точки M и N по поверхности куба.

Повернем верхнюю грань куба $A_1B_1C_1D_1$ вокруг прямой A_1D_1 так, чтобы она оказалась в плоскости левой боковой грани и эти два квадрата образовывали прямоугольник со сторонами $2a$ и $4a$. Затем полученный прямоугольник повернем вокруг прямой AA_1 так, чтобы он оказался в плоскости передней грани. Получим развертку верхней, левой боковой и передней граней, т.е. ступенчатую фигуру $BADD_1''C_1''B_1''A_1B_1B$. Если от развертки вернуться к поверхности куба, то склеиваются следующие точки: B_1'' и B_1 , C_1'' и C_1 , D_1'' и D_1 . Точка N на верхней грани куба при таком развороте окажется в положении N_2 . Кратчайшей линией на развертке в этом случае является отрезок MN_2 .

Возвращаясь от развертки к поверхности куба, получаем ломаную линию MK_2T_2N , которая является третьим претендентом на кратчайшую линию, соединяющую точки M и N по поверхности куба.

Из трех отрезков MN , MN_1 , MN_2 выберем кратчайший отрезок и потом построим соответствующую кратчайшую ломаную линию на поверхности куба. Идея решения сформулирована и ее можно осуществить на бумажной модели куба и его развертки.

Перейдем к разработке математической модели.

Пусть начало пространственной системы координат $O_{хуз}$ расположено в центре куба, а оси проведены перпендикулярно граням куба, тогда куб определяется неравенствами:

$$-a \leq x \leq a, -a \leq y \leq a, -a \leq z \leq a.$$

Пусть известны пространственные координаты данных точек:

$$M(x_M; y_M; z_M), N(x_N; y_N; z_N).$$

На развертке введем систему координат $O'yz$, начало которой совпадает с центром квадрата ABB_1A_1 , а оси параллельны сторонам квадрата.

Координаты точек на развертке выражаются через пространственные координаты следующим образом:

$$M(y_M; z_M), N(y_N; 2a - x_N).$$

Если точка $Q(y'; z')$ получена из точки $Q(y; z)$ при повороте $R_{M_0}^\varphi$ вокруг точки $Q(y_0; z_0)$ на угол φ , то координаты преобразуются по формулам:

$$y' - y_0 = (y - y_0) \cos \varphi - (z - z_0) \sin \varphi,$$

$$z' - z_0 = (y - y_0) \sin \varphi + (z - z_0) \cos \varphi.$$

При повороте $R_{B_1}^{-90^\circ}$ вокруг точки B_1 на угол $\varphi = -90^\circ$ получим формулы преобразования:

$$y' = z, z' = 2a - y.$$

Точка N при повороте $R_{B_1}^{-90^\circ}$ отображается в точку N_2 , поэтому $N_2(2a - x_N; 2a - y_N)$.

При повороте $R_{A_1}^{90^\circ}$ вокруг точки A_1 на угол $\varphi = 90^\circ$ получим формулы преобразования:

$$y' = -z, z'' = 2a + y.$$

Точка N при повороте $R_{A_1}^{90^\circ}$ отображается в точку N_2 , поэтому $N_2(x_N - 2a; y_N + 2a)$.

Используя уравнение прямой, проходящей через две точки $Q_1(y_1; z_1)$ и $Q_2(y_2; z_2)$, т.е. уравнение $\frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}$, составим уравнения прямых:

$$MN: (y - y_M)(2a - x_N - z_M) = (z - z_M)(y_N - y_M),$$

$$MN_1: (y - y_M)(2a - y_N - z_M) = (z - z_M)(2a - x_N - y_M),$$

$$MN_2: (y - y_M)(2a + y_N - z_M) = (z - z_M)(x_N - 2a - y_M).$$

Прямая MN пересекает прямую A_1B_1 , заданную уравнением $z = a$, в точке H , для которой

$$y = \frac{a(y_M + y_N) - z_M y_N - y_M x_N}{2a - x_N - z_M}, z = a.$$

Прямая MN_1 пересекает прямую B_1C_1' , заданную уравнением $z = a$, в точке T_1 , для которой

$$y = \frac{a(2a + y_M - x_N - 2z_M) - y_M y_N + z_M x_N}{2a - y_N - z_M}, z = a.$$

Прямая MN_1 пересекает прямую BB_1 , заданную уравнением $y = a$, в точке K_1 , для которой

$$y = a, \quad z = \frac{a(2a - y_N + z_M - 2y_M) + y_M y_N - x_N z_M}{2a - x_N - y_M}.$$

Прямая MN_2 пересекает прямую A_1D_1'' , заданную уравнением $z = a$, в точке T_2 , для которой

$$y = \frac{a(-2a + y_M + x_N + 2z_M) + y_M y_N - z_M x_N}{2a + y_N - z_M}, z = a.$$

Прямая MN_2 пересекает прямую AA_1 , заданную уравнением $y = -a$, в точке K_2 , для которой

$$y = -a, \quad z = \frac{a(-2a - y_N - z_M - 2y_M) - y_M y_N + x_N z_M}{-2a + x_N - y_M}.$$

3.6. Разработайте компьютерную программу для рис. 8 на основе проведенного математического моделирования. Исходные параметры для программы: $2a$ — длина ребра куба, d — число кратчайших линий в семействе, x_0, y_0 — координаты центра куба, s_0, w_0 — координаты центра смещения развертки. Программа должна распечатывать значения длин отрезков MN, MN_1, MN_2 . Сравните с программой «Кратчайшая на кубе».

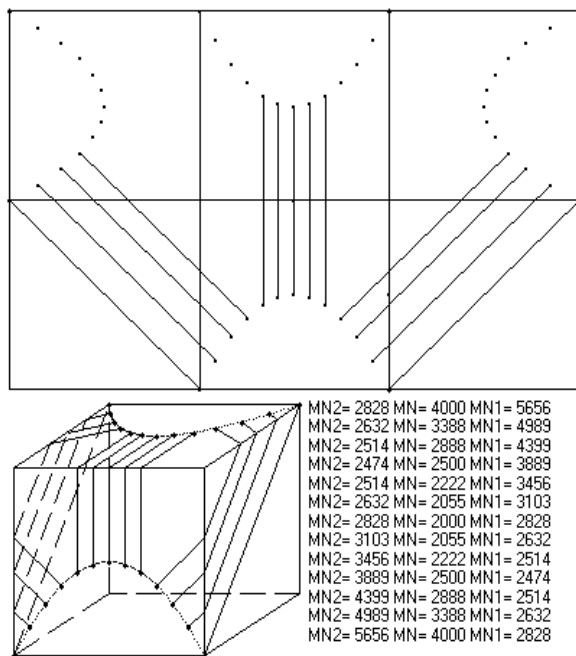


Рис. 8

3.7. Рассчитайте параметры белой области вокруг вершины B_1 (рис. 9), если точка M движется вдоль диагонали A_1B , а точка N движется по диагонали A_1C_1 , причем они движутся с равными скоростями. Длина куба равна $2a$.

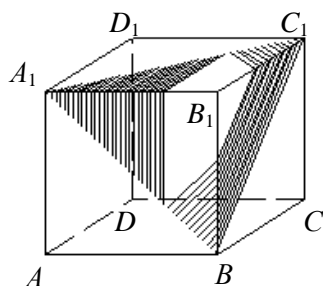


Рис. 9

3.8. Исследуйте закрытые зоны для маршрутов, т.е. белые пятна на кубе, в зависимости от траекторий точек M и N (рис. 10—11). Напишите компьютерные программы построения этих рисунков.

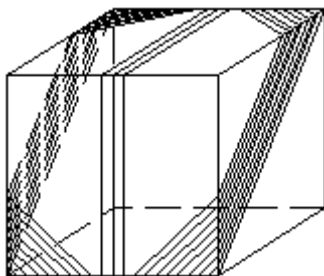


Рис. 10

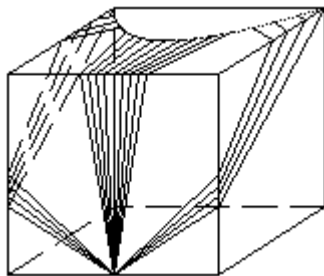


Рис. 11

Изучим кратчайшие линии на поверхности куба с концевыми точками на параллельных гранях.

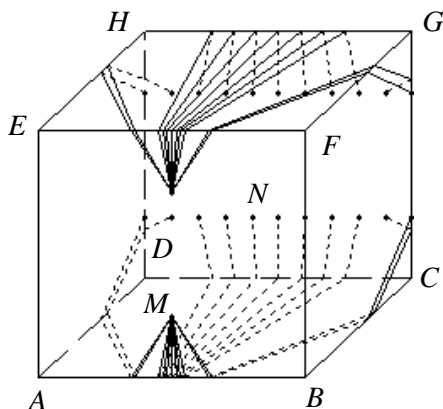


Рис. 12

На рис. 12 построено два семейства кратчайших линий с концевыми точками M и N на параллельных гранях поверхности куба. Целью исследования является математическое моделирование кратчайших линий, если известны законы движения точек M и N .

Рассмотрим куб, ребро которого равно $2a$. Для построения кратчайшей линии, соединяющей две

данные точки M и N на параллельных гранях поверхности куба, необходимо построить развертки поверхности, провести отрезки, соединяющие соответственные точки на развертках, и выбрать из них кратчайший отрезок. Сложив из развертки снова куб, получим ломаную линию, являющуюся кратчайшей линией.

Построим граф куба, чтобы понять, развертки каких граней нужно построить (рис. 13). Задняя грань $CDHG$ куба на плоском графе изображается областью, внешней к квадрату $CDHG$. Точка N на задней грани куба изображена четырьмя точками N на графе, которые нужно отождествить.

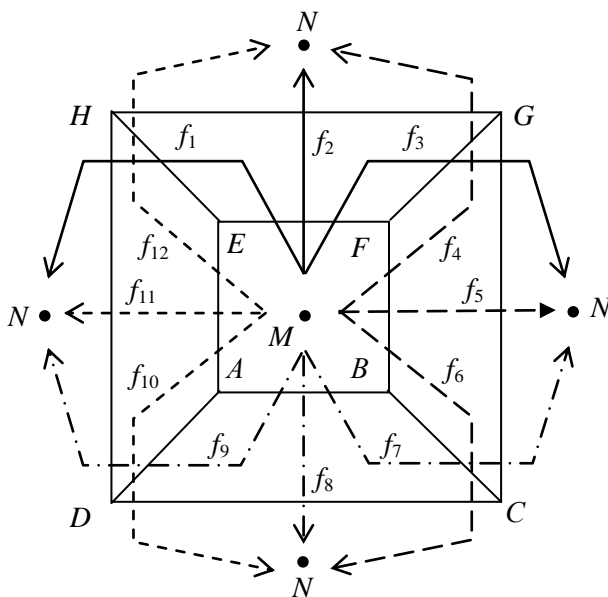


Рис. 13

Чтобы пройти из точки M в точку N , нужно пересечь одну из четырех сторон квадрата $ABFE$. Это первое действие можно выполнить четырьмя способами. Выйти из смежной грани можно тремя способами. Дальнейший путь к точке N по кратчайшему маршруту является строго определенным. По правилу произведения получаем $4 \cdot 3 = 12$ способов, чтобы пройти из M в точку N . На рис. 2 перечислены соответствующие маршруты f_1, f_2, \dots, f_{12} . Рассмотрим соответствующие развертки некоторых граней куба.

Развернув переднюю грань $ABFE$, верхнюю грань $EFGH$ и заднюю грань $HGCD$ куба в одну плоскость передней грани, получим четырехугольник ABC_2D_2 (рис. 14). Аналогично из каждой трех граней куба получим развертки AD_5H_5E , EFG_8H_8 , $BC_{11}G_{11}F$.



Рис. 15

284

указывают направление прохождения квадратов развертки. Аналогично можно построить 8 ступенчатых областей. Нумерация точек в последнем квадрате каждой развертки совпадает с нумерацией маршрута. На рис. 14 построено 12 неполных разверток куба, содержащих только те грани, через которые могут пройти кратчайшие линии. Через некоторые квадраты развертки проходит несколько различных маршрутов.

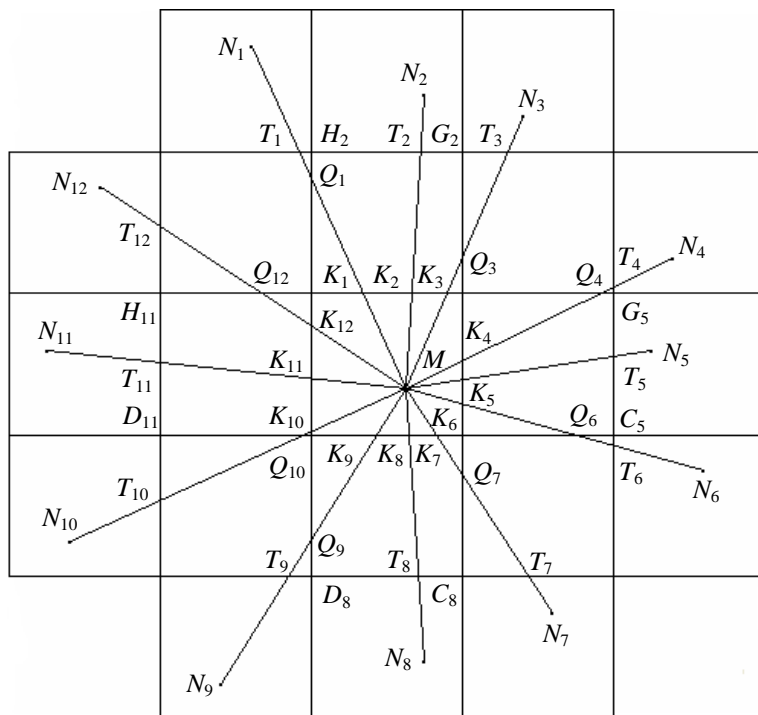


Рис. 16

На каждой из двенадцати разверток появится образ точки N куба, т.е. точки N_1, N_2, \dots, N_{12} .

На рис. 16 изображены 12 отрезков $MN_1, MN_2, \dots, MN_{12}$, среди которых нужно выбрать наименьший. Потом соответствующую развертку с наименьшим отрезком нужно наложить на куб, чтобы построить кратчайшую линию.

Пусть начало координат находится в центре куба, ось Ox сонаправлена с лучом $[CB)$, а ось Oy сонаправлена с лучом $[AB)$. Пусть на развертке начало координат совпадает с центром квадрата, ось Oy сонаправлена с лучом $[AB)$, ось Oz сонаправлена с лучом $[BF)$. Система координат на развертке выбрана согласованной с системой координат в пространстве, т.е. она получена перемещением пространственной системы координат в центр передней грани. Для двух осей пространственной и плоской системы координат выбраны одни те же обозначения, а различать их будем по числу координат каждой точки. Пусть точка M на кубе имеет пространственные координаты x_M, y_M, z_M , тогда на развертке получаем $M(y_M, z_M)$.

Пусть точка N на кубе имеет пространственные координаты x_N, y_N, z_N , тогда на развертке получаем:

$$N_2(y_N; 4a - z_N), N_5(4a - y_N; z_N), N_8(y_N; -4a - z_N), N_{11}(-4a - y_N; z_N).$$

Немного сложнее определяются координаты остальных точек $N_1, N_3, N_4, N_6, N_7, N_9, N_{10}, N_{11}$.

На развертке (рис. 5) квадрат $H_1G_1C_1D_1$ может быть получен из квадрата $H_2G_2C_2D_2$ поворотом вокруг точки H_2 на угол $\pi/2$. Точка N_2 при этом повороте отображается в точку N_1 .

Если точка $Q(y'; z')$ получена из точки $Q(y; z)$ при повороте $R_{M_0}^\varphi$ вокруг точки $Q(y_0; z_0)$ на угол φ , то координаты преобразуются по формулам:

$$y' - y_0 = (y - y_0) \cos \varphi - (z - z_0) \sin \varphi,$$

$$z' - z_0 = (y - y_0) \sin \varphi + (z - z_0) \cos \varphi.$$

При повороте $R_{M_0}^{90^\circ}$ вокруг точки $Q(y_0; z_0)$ на угол $\varphi = 90^\circ$ получим формулы преобразования:

$$y' - y_0 = -(z - z_0), \quad z' - z_0 = (y - y_0).$$

При повороте $R_{M_0}^{90^\circ}$ вокруг точки $Q(y_0; z_0)$ на угол $\varphi = -90^\circ$ получим формулы преобразования:

$$y' - y_0 = (z - z_0), \quad z' - z_0 = -(y - y_0).$$

Подставляя в формулы поворота $R_{H_2}^{90^\circ}$ $y' = -z + 2a, z' = y + 4a$ координаты точки N , получим $N_1(z_N - 2a; y_N + 4a)$.

Точка N_2 при повороте $R_{G_2}^{-90^\circ}$ отображается в точку N_3 , поэтому $N_3(2a - z_N; 4a - y_N)$.

Точка N_5 при повороте $R_{G_5}^{90^\circ}$ отображается в точку N_4 , поэтому $N_4(-z_N + 4a; -y_N + 2a)$.

Точка N_5 при повороте $R_{C_5}^{-90^\circ}$ отображается в точку N_6 , поэтому $N_6(z_N + 4a; y_N - 2a)$.

Точка N_8 при повороте $R_{C_8}^{90^\circ}$ отображается в точку N_7 , поэтому $N_7(z_N + 2a; y_N - 4a)$.

Точка N_8 при повороте $R_{D_8}^{-90^\circ}$ отображается в точку N_9 , поэтому $N_9(-z_N - 2a; -y_N - 4a)$.

Точка N_{11} при повороте $R_{D_{11}}^{90^\circ}$ отображается в точку N_{10} , поэтому $N_{10}(-z_N - 4a; -y_N - 2a)$.

Точка N_{11} при повороте $R_{H_{11}}^{-90^\circ}$ отображается в точку N_{12} , поэтому $N_{12}(z_N - 4a; y_N + 2a)$.

Для отрезков $MN_1, MN_2, \dots, MN_{12}$ найдем их длины:

$$MN_i = \sqrt{(y_{N_i} - y_M)^2 + (z_{N_i} - z_M)^2}, i = 1, 2, \dots, 12.$$

С помощью цикла в компьютерной программе найдем минимальную длину, которой будет соответствовать один или несколько отрезков на развертках. Чтобы наложить эти отрезки на куб, найдем точки пересечения всех отрезков с линиями сетки, порожденной ребрами куба, т.е. со сторонами на соответствующей развертке.

Пусть отрезок MN_2 пересекает отрезки EF и H_2G_2 , соответственно, в точках K_2 и T_2 .

Используя уравнение прямой, проходящей через две точки $Q_1(y_1; z_1)$ и $Q_2(y_2; z_2)$, т.е. уравнение $\frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}$,

составим уравнение прямой MN_2 :

$$MN_2: (y - y_M)(4a - z_N - z_M) = (z - z_M)(y_N - y_M).$$

Точки пересечения прямой MN_2 с сеткой развертки:

$$K_2 \left(\frac{3ay_M + ay_N - z_M y_N - y_M z_N}{4a - z_M - z_N}; a \right),$$

$$T_2 \left(\frac{ay_M + 3ay_N - z_M y_N - y_M z_N}{4a - z_M - z_N}; 3a \right).$$

Аналогично получаем уравнение прямой MN_5 :

$$MN_5: (y - y_M)(z_N - z_M) = (z - z_M)(4a - y_N - y_M).$$

Точки пересечения прямой MN_5 с сеткой развертки:

$$K_5 \left(a; \frac{3az_M + az_N - y_M z_N - y_N z_M}{4a - y_M - y_N} \right),$$

$$T_5 \left(3a; \frac{az_M + 3az_N - y_M z_N - y_N z_M}{4a - y_M - y_N} \right).$$

Уравнение прямой MN_8 :

$$MN_8: (y - y_M)(-4a - z_N - z_M) = (z - z_M)(y_N - y_M).$$

Точки пересечения прямой MN_8 с сеткой развертки:

$$K_8 \left(\frac{3ay_M + ay_N + z_M y_N + y_M z_N}{4a + z_M + z_N}; -a \right),$$

$$T_8 \left(\frac{ay_M + 3ay_N + z_M y_N + y_M z_N}{4a + z_M + z_N}; -3a \right).$$

Уравнение прямой MN_{11} :

$$MN_{11}: (y - y_M)(z_N - z_M) = (z - z_M)(-4a - y_N - z_M).$$

Точки пересечения прямой MN_{11} с сеткой развертки:

$$K_{11} \left(-a; \frac{3ay_M + ay_N + z_M y_N + y_M z_N}{4a + z_M + z_N} \right),$$

$$T_{11} \left(-3a; \frac{ay_M + 3ay_N + z_M y_N + y_M z_N}{4a + z_M + z_N} \right).$$

Пусть отрезок MN_1 пересекает отрезки EF , EH_2 и H_1D_1 , соответственно, в точках K_1 , Q_1 и T_1 .

Уравнение прямой MN_1 :

$$MN_1: (y - y_M)(y_N + 4a - z_M) = (z - z_M)(z_N - 2a - y_M).$$

$$K_1 \left(\frac{-2a^2 + 3ay_M + 2az_M + az_N + y_M y_N - z_M z_N}{4a - z_M + y_N}; a \right),$$

$$T_1 \left(\frac{-6a^2 + ay_M + 2az_M + 3az_N + y_M y_N - z_M z_N}{4a - z_M + y_N}; 3a \right),$$

$$Q_1 \left(-a; \frac{4a^2 + 4ay_M + az_M + ay_N + y_M y_N - z_M z_N}{2a + y_M - z_N} \right).$$

Обращаем внимание читателя на то, что точки M и N на развертке могут быть расположены так (рис. 17), что отрезок MN_1 не принадлежит развертке $ABG_2H_1G_1C_1A_1EA$.

Чтобы исключить такой случай, на координату точки Q_1 нужно наложить условие $a \leq z_{Q_1} \leq 3a$.

Аналогично получаем для остальных прямых.

Уравнение прямой MN_3 :

$$MN_3: (y - y_M)(-y_N + 4a - z_M) = (z - z_M)(-z_N + 2a - y_M).$$

$$K_3 \left(\frac{2a^2 + 3ay_M - 2az_M - az_N - y_M y_N + z_M z_N}{4a - z_M - y_N}; a \right),$$

$$T_3 \left(\frac{6a^2 + ay_M - 2az_M - 3az_N - y_M y_N + z_M z_N}{4a - z_M - y_N}; 3a \right),$$

$$Q_3 \left(a; \frac{4a^2 - 4ay_M + az_M - ay_N + y_M y_N - z_M z_N}{2a - y_M - z_N} \right).$$

Координата точки Q_3 должна удовлетворять условию $a \leq z_{Q_3} \leq 3a$.

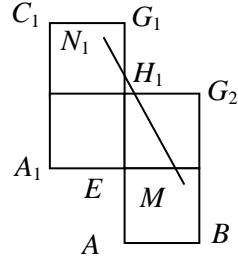


Рис. 17

Уравнение прямой MN_4 :

$$MN_4: (y - y_M)(-y_N + 2a - z_M) = (z - z_M)(-z_N + 4a - y_M).$$

$$K_4 \left(a; \frac{2a^2 - 2ay_M + 3az_M - ay_N + y_M y_N - z_M z_N}{4a - y_M - z_N} \right),$$

$$T_4 \left(3a; \frac{6a^2 - 2ay_M + az_M - 3ay_N + y_M y_N - z_M z_N}{4a - y_M - z_N} \right),$$

$$Q_4 \left(\frac{4a^2 + ay_M - 4az_M - az_N - y_M y_N + z_M z_N}{2a - y_N - z_M}; a \right).$$

Координата точки Q_4 должна удовлетворять условию $a \leq y_{Q_4} \leq 3a$.

Уравнение прямой MN_6 :

$$MN_6: (y - y_M)(y_N - 2a - z_M) = (z - z_M)(z_N + 4a - y_M).$$

$$K_6 \left(a; \frac{-2a^2 + 2ay_M + 3az_M + ay_N - y_M y_N + z_M z_N}{4a - y_M + z_N} \right),$$

$$T_6 \left(3a; \frac{-6a^2 + 2ay_M + az_M + 3ay_N - y_M y_N + z_M z_N}{4a - y_M + z_N} \right),$$

$$Q_6 \left(\frac{4a^2 + ay_M + 4az_M + az_N - y_M y_N + z_M z_N}{2a - y_N + z_M}; -a \right).$$

Координата точки Q_6 должна удовлетворять условию $a \leq y_{Q_6} \leq 3a$.

Уравнение прямой MN_7 :

$$MN_7: (y - y_M)(y_N - 4a - z_M) = (z - z_M)(z_N + 2a - y_M).$$

$$K_7 \left(\frac{2a^2 + 3ay_M + 2az_M + az_N - y_M y_N + z_M z_N}{4a + z_M - y_N}; -a \right),$$

$$T_7 \left(\frac{6a^2 + ay_M + 2az_M + 3az_N - y_M y_N + z_M z_N}{4a + z_M - y_N}; -3a \right),$$

$$Q_7 \left(a; \frac{-4a^2 + 4ay_M + az_M + ay_N - y_M y_N + z_M z_N}{2a - y_M + z_N} \right).$$

Координата точки Q_7 должна удовлетворять условию $-3a \leq z_{Q_7} \leq -a$.

Уравнение прямой MN_9 :

$$\begin{aligned} MN_9: & (y - y_M)(y_N + 4a + z_M) = (z - z_M)(z_N + 2a + y_M). \\ K_9 & \left(\frac{-2a^2 + 3ay_M - 2az_M - az_N + y_M y_N - z_M z_N}{4a + z_M + y_N}; -a \right), \\ T_9 & \left(\frac{-6a^2 + ay_M - 2az_M - 3az_N + y_M y_N - z_M z_N}{4a + z_M + y_N}; -a \right), \\ Q_9 & \left(-a; \frac{-4a^2 - 4ay_M + az_M - ay_N - y_M y_N + z_M z_N}{2a + y_M + z_N} \right). \end{aligned}$$

Координата точки Q_9 должна удовлетворять условию $-3a \leq z_{Q_9} \leq -a$.

Уравнение прямой MN_{10} :

$$\begin{aligned} MN_{10}: & (y - y_M)(y_N + 2a + z_M) = (z - z_M)(z_N + 4a + y_M). \\ K_{10} & \left(-a; \frac{-2a^2 - 2ay_M + 3az_M - ay_N - y_M y_N + z_M z_N}{4a + y_M + z_N} \right), \\ K_{10} & \left(-3a; \frac{-6a^2 - 2ay_M + az_M - 3ay_N - y_M y_N + z_M z_N}{4a + y_M + z_N} \right), \\ Q_{10} & \left(\frac{-4a^2 + ay_M - 4az_M - az_N + y_M y_N - z_M z_N}{2a + y_N + z_M}; -a \right). \end{aligned}$$

Координата точки Q_{10} должна удовлетворять условию $-3a \leq y_{Q_{10}} \leq -a$.

Уравнение прямой MN_{12} :

$$\begin{aligned} MN_{12}: & (y - y_M)(y_N + 2a - z_M) = (z - z_M)(z_N - 4a - y_M). \\ K_{12} & \left(-a; \frac{-2a^2 - 2ay_M - 3az_M - ay_N - y_M y_N + z_M z_N}{z_N - 4a - y_M} \right), \\ T_{12} & \left(-3a; \frac{-6a^2 - 2ay_M - az_M - 3ay_N - y_M y_N + z_M z_N}{z_N - 4a - y_M} \right), \end{aligned}$$

$$Q_{12} \left(\frac{-4a^2 + ay_M + 4az_M + az_N + y_M y_N - z_M z_N}{2a + y_N - z_M}; -a \right).$$

Координата точки Q_{12} должна удовлетворять условию $-3a \leq y_{Q_{12}} \leq -a$.

Замечание. Если какой-то из отрезков MN_1, MN_3, MN_7, MN_9 окажется вертикальным, или какой-то из отрезков $MN_4, MN_6, MN_{10}, MN_{12}$ окажется горизонтальным, то соответствующую точку Q_i нельзя определить, так как знаменатель дроби принимает нулевое значение, но в этом случае эту точку и не нужно строить. Ломаная $K_i Q_i T_i$ превращается в отрезок $K_i T_i$.

Пусть из двенадцати отрезков $MN_1, MN_2, \dots, MN_{12}$ кратчайшим оказался отрезок MN_1 , который нужно наложить на куб, чтобы изобразить кратчайшую линию $MKQT$. Рассмотрим схему пересчета координат от точек, лежащих на развертке, к точкам, лежащим на поверхности куба (рис. 18, 19).

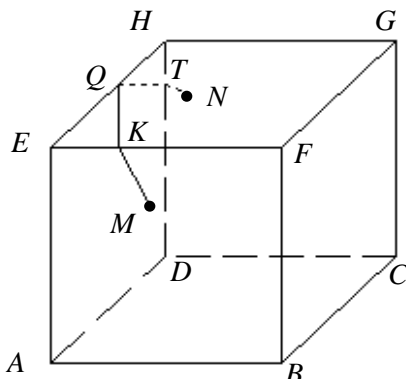


Рис. 18

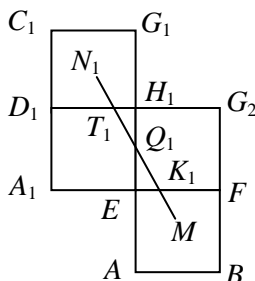


Рис. 19

Для точек $K_1(y_{K_1}; z_{K_1})$ и $K(x_K; y_K; z_K)$ получаем $x_k = a, y_k = y_{k1}, z_k = z_{k1}$.

Точки T и N расположены на задней грани. При развороте кратчайшей их образы будут расположены в квадрате $D_1 H_1 G_1 C_1$. Связи между координатами на поверхности куба и координатами на развертке должны записываться по одному и тому же закону, поэтому для точек $T_1(y_{T_1}; z_{T_1})$ и $T(x_T; y_T; z_T)$ получаем

$$y_{T_1} = z_T - 2a, z_{T_1} = y_T + 4a, z_{T_1} = 3a.$$

Следовательно,

$$x_T = -a, y_T = -a, z_T = 2a + y_{T_1}.$$

Для установления связи между координатами точек $Q_1(y_{Q_1}; z_{Q_1})$ и $Q(x_Q; y_Q; z_Q)$ изобразим оси на развертке передней и верхней граней (рис. 20) и получаем:

$$x_Q = 2a - z_{Q_1}, y_Q = -a, z_Q = a.$$

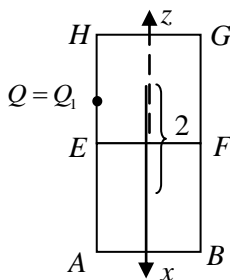


Рис. 20

Две точки на поверхности куба назовем диаметрально противоположными, если они расположены на прямой, проходящей через центр куба. Координаты точек $M(x_M; y_M; z_M)$, $N(x_N; y_N; z_N)$ удовлетворяют условиям $x_N = -x_M, y_N = -y_M, z_N = -z_M$.

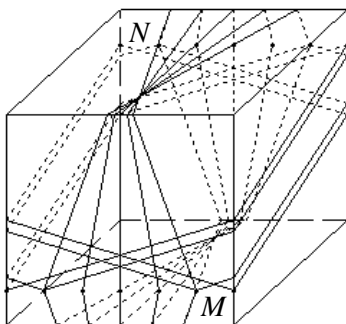


Рис. 21

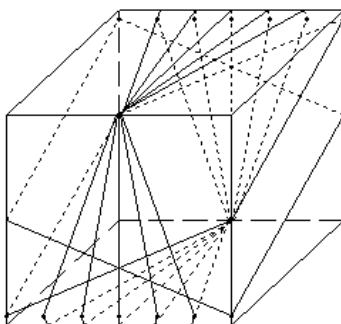


Рис. 22

Для каждой кратчайшей на поверхности куба с концевыми точками в диаметрально противоположных точках существует другая кратчайшая на поверхности куба, полученная из первой кратчайшей симметрией относительно центра куба. Рисунки 21 и 22, построенные с помощью компьютерной программы, подтверждают это высказывание и дополнительно показывают, что для некоторых пар диаметрально противоположных точек на кубе может существовать четыре кратчайшие линии. А шесть или восемь кратчайших линий на поверхности куба с концевыми точками в диаметрально противоположных точках может существовать?

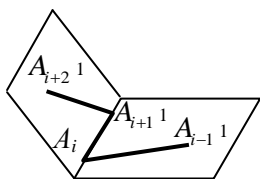


Рис. 23

Ломаная линия называется *плоской*, если она расположена в одной плоскости. Если ломаная линия $A_1 A_2 \dots A_i \dots A_n$ не является плоской (рис. 23), то существует такое звено $A_i A_{i+1}$ этой ломаной линии, что плоскости $A_{i-1} A_i A_{i+1}$ и $A_i A_{i+1} A_{i+2}$ являются различными. В этом случае говорят, что ломаная линия на ребре $A_i A_{i+1}$ имеет кручение.

Если концевые точки кратчайшей линии на поверхности куба являются диаметрально противоположными точками, то линия является плоской?

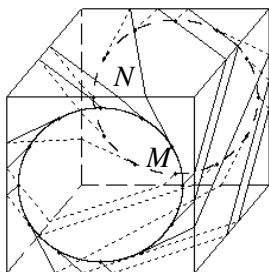


Рис. 24

На рис. 24 построено семейство кратчайших линий. Точка M движется в передней грани по окружности, а точка N движется в задней грани по окружности. Параметры для соответственных точек отличаются на одну и ту же величину, т.е. соответственные точки сдвинуты по окружности на одно и то же расстояние. Кратчайшие линии оказались закрученными.

На рис. 25 построено семейство кратчайших, для которых точка M движется в передней грани по окружности, точка N движется в задней грани по окружности, причем их параметризации не имеют сдвига. На поверхности куба оказалось четыре не закрашенные области. На рис. 26 построено семейство кратчайших, для которых точки M и N имеют сдвиг.

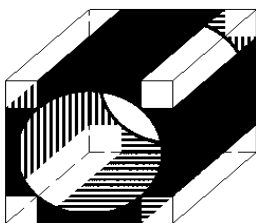


Рис. 25

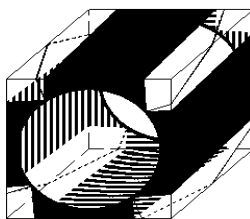


Рис. 26

Снова не закрашено четыре области, но границы каждой области состоят из закрученных ломаных линий.

На рис. 27 построено семейство кратчайших линий. Точка M зафиксирована и находится на передней грани, а точка N движется в задней грани по вертикальному отрезку. Отрезок разделен на 12 частей. Для нумерации точек выбраны нижние индексы слева, так как нумерация с помощью нижних правых индексов использовалась выше для различных изображений одной и той же точки N на развертках. Существуют по две кратчайшие линии, соединяющие пары точек M и $_0N$, M и $_8N$.

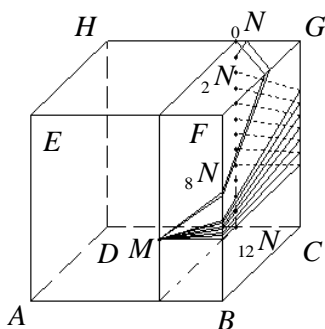


Рис. 27

На рис. 28 построено это же семейство кратчайших, но точка N непрерывно движется по вертикальному отрезку. Непрерывность здесь понимается с точки зрения информатики, т.е. с достаточно малым шагом изменения параметра. Все кратчайшие разбивают точки поверхности на четыре области. Перечислите эти области.

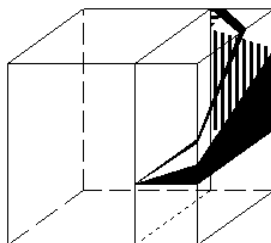


Рис. 28

На рис. 29 точка M расположена в центре передней грани, а точка N движется по диагонали CH задней грани. Это множество точек симметрично относительно диагональной плоскости $BCEH$, а также $AFGD$, поэтому и семейство кратчайших линий симметрично относительно этих плоскостей.

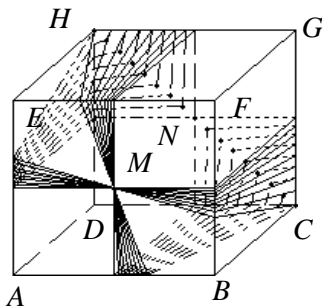


Рис. 29

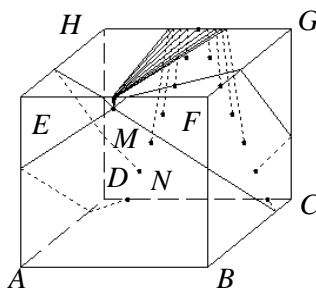


Рис. 30

3.9. Определите элементы симметрии для кратчайших линий на рис. 30.

3.10. На рис. 31 точка M расположена в центре передней грани, а точка N движется по синусоиде в задней грани, проходящей через центр задней грани. Определите элементы симметрии для кратчайших линий на рис. 31.

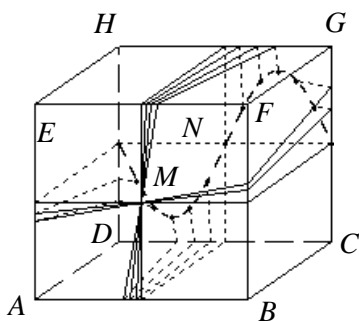


Рис. 31

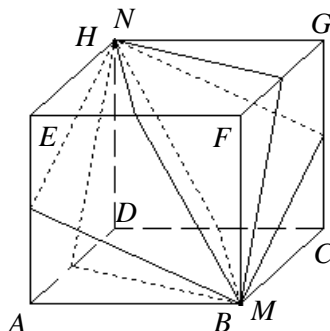


Рис. 32

3.11. Точки M и N (рис. 32) расположены на концах диагонали куба. Перечислите все элементы симметрии для семейства кратчайших линий, соединяющих точки M и N .

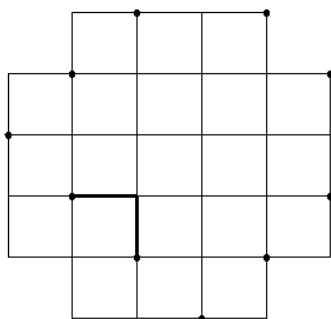


Рис. 33

3.12. На передней и задней гранях куба даны фиксированные точки M и N . Программа построила 12 изображений точки N на всех развертках, но получилось 9 точек. На рис. 33 они изображены жирными точками. Из двенадцати отрезков на развертках программа выбрала кратчайший отрезок. Получилось два отрезка, изображенных жирными линиями. Как расположены точки M и N на поверхности куба? Постройте кратчайшую линию на поверхности куба для этого случая.

Постройте кратчайшую линию на поверхности куба для этого случая.

§ 24. Опорная функция линии и радиус кривизны

Линию на плоскости можно задать графиком явной функции $y = f(x)$; множеством точек, координаты которых удовлетворяют общему уравнению $F(x, y) = 0$, или с помощью параметрических уравнений $x = x(t)$, $y = y(t)$.

Изучим новый метод задания линия на плоскости.

Пусть в каждой точке линии определена касательная и различные точки линии имеют различные касательные. От точки касания линии с касательной можно отложить два единичных вектора, перпендикулярных касательной (рис. 1). В одной точке выберем один из таких векторов, и далее в остальных точках линии будем выбирать один из двух векторов таким образом, чтобы при переходе от одной точки к другой векторы непрерывно изменялись. Вдоль линии получим семейство единичных векторов, которое называется полем нормальных векторов. Например, если линия является замкнутой и выпуклой, то рассмотрим векторы, направленные во внешнюю часть плоскости относительно линии (рис. 2).

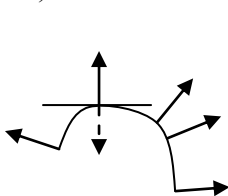


Рис. 1

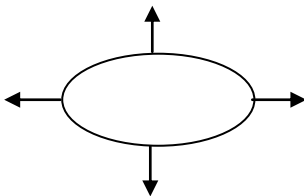


Рис. 2

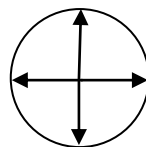


Рис. 3

Если перенести полученные нормальные векторы и отложить от одной точки (рис. 3), то получим множество нормальных векторов для данной линии, которое называется *нормальным изображением линии*. Вместо множества нормальных векторов, отложенных от одной точки, можно рассмотреть концевые точки этих векторов, т.е. множество точек на единичной окружности. Для замкнутой линии нормальным изображением является окружность единичного радиуса.

Рассмотрим пример аналитического определения нормального изображения для параболы $y = ax^2$, где $a > 0$.

Составим параметрические уравнения параболы: $x = t, y = at^2$.

Получаем векторную функцию параболы $\vec{r}(t; at^2)$. Вектор $\vec{r}'(1; 2at)$ является направляющим для касательной. Рассмотрим вектор $\vec{N}(2at; -1)$, перпендикулярный вектору $\vec{r}'(1; 2at)$. Если векторы \vec{N} отложить от начала координат, то концевые точки этих векторов расположены на прямой $y = -1$ (рис. 4).

Вектор $\vec{n} = \frac{\vec{N}}{|\vec{N}|} = \left(\frac{2at}{\sqrt{1+4a^2t^2}}; -\frac{1}{\sqrt{1+4a^2t^2}} \right)$ имеет единичную

длину.

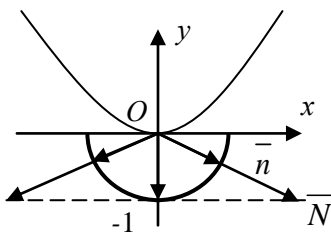


Рис. 4

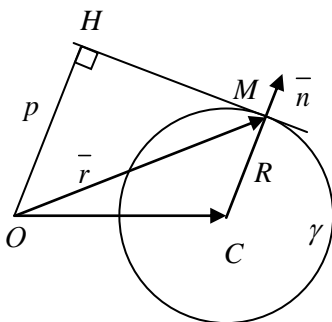


Рис. 5

Множество концевых точек таких векторов, отложенных от начала координат, образует полуокружность без концевых точек этой полуокружности. Нормальным изображением параболы является полуокружность без концевых точек.

Рассмотрим снова произвольную линию γ (рис. 5). Для каждого нормального единичного вектора \vec{n} линии определим вектор $\vec{r} = \overline{OM}$, выходящий из начала координат O и приходящий в точку касания M . Составим скалярную функцию $p = \vec{r} \cdot \vec{n}$, которая называется *опорной функцией* линии γ .

Пусть H — основание перпендикуляра, опущенного из начала координат на касательную с нормальным вектором \vec{n} .

Для опорной функции получаем

$$p = |\vec{r}| \cdot |\vec{n}| \cos(\vec{r}, \vec{n}) = \pm OM \cos(\overline{OM}, \overline{OH}) = \pm OH.$$

Если векторы \vec{r} и \vec{n} образуют острый угол, то опорная $p(\vec{n})$ функция имеет простой геометрический смысл — это расстояние от начала координат до касательной с нормальным вектором \vec{n} .

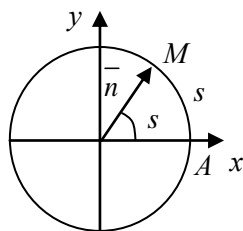


Рис. 6

Приведем примеры составления опорных функций для некоторых линий, но вначале рассмотрим параметризацию единичных нормальных векторов или равносильно этому высказыванию — параметризацию точек на единичной окружности.

Пусть вектор \vec{n} образует угол s с положительным направлением оси Ox (рис. 6). Для единичной окружности длина дуги AM также равна s . Такой параметр для единичной окружности называется естественным параметром. Для естественной параметризации окружности получаем $\vec{n}(\cos s; \sin s)$.

Пусть γ — окружность с центром $C(a; b)$ и радиусом R (рис. 5), тогда

$$\vec{r} = \overline{OM} = \overline{OC} + \overline{CM} = \overline{OC} + R\vec{n} = (a, b) + R(\cos s, \sin s).$$

$$p = \vec{r} \cdot \vec{n} = R + a \cos s + b \sin s.$$

Если центр окружности находится в начале координат, то опорная функция окружности равна $p = R$. Кстати, это становится очевидным равенством, если вспомнить, что опорная функция в этом случае равна расстоянию от центра до семейства касательных к окружности, т.е. равна радиусу окружности.

Слагаемое $a \cos s + b \sin s$ в опорной функции характеризует перенос фигуры в точку $C(a; b)$.

Для параболы $x = t, y = at^2$ опорная функция равна

$$p = \vec{r} \cdot \vec{n} = \frac{2at^2}{\sqrt{1+4a^2t^2}} - \frac{at^2}{\sqrt{1+4a^2t^2}} = \frac{at^2}{\sqrt{1+4a^2t^2}}.$$

Опорная функция вычислена как функция от параметра t линии. В дальнейшем будет показано, что для некоторых задач геометрии наиболее естественной является определение опорной функции как функции естественного параметра s на нормальном изображении, т.е. от параметра s на единичной окружности.

Сравнивая координаты единичного вектора

$$\bar{n}(\cos s, \sin s) \text{ и } \bar{n} = \left(\frac{2at}{\sqrt{1+4a^2t^2}}; -\frac{1}{\sqrt{1+4a^2t^2}} \right), \text{ получаем}$$

$$\frac{2at}{\sqrt{1+4a^2t^2}} = \cos s, -\frac{1}{\sqrt{1+4a^2t^2}} = \sin s \text{ или } tgs = -\frac{1}{2at}, t = -\frac{1}{2atgs}.$$

Для естественной параметризации получаем опорную функцию

$$p = -\frac{\cos^2 s}{4a \sin s}, \text{ где } s \in (-\pi; 0).$$

Найдем опорную функцию для эллипса, заданного векторной функцией $\bar{r} = (a \cos t; b \sin t)$. Направляющий вектор касательной $\bar{r}' = (-a \sin t, b \cos t)$. Нормальный вектор $\bar{N}(b \cos t; a \sin t)$.

Единичный нормальный вектор

$$\bar{n} \left(\frac{b \cos t}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}}; \frac{a \sin t}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}} \right).$$

Опорная функция, зависящая от параметра t :

$$p = \bar{r} \cdot \bar{n} = \frac{ab}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}}.$$

Если перейти к параметру s нормального изображения:

$$\frac{b \cos t}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}} = \cos s, \frac{a \sin t}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}} = \sin s,$$

то получаем

$$p = \sqrt{a^2 \cos^2 s + b^2 \sin^2 s}.$$

Если линия γ имеет в каждой точке касательную, то всегда можно определить опорную функцию линии $p(\bar{n})$, которую можно также обозначить через $p(s)$.

А как по опорной функции $p(s)$ восстановить линию, т.е. определить координаты той точки на линии, в которой касательная

имеет единичный нормальный вектор \bar{n} и касательная удалена от начала координат на расстояние $p(\bar{n})$?

Перепишем определение опорной функции $p = \bar{r} \cdot \bar{n}$ в виде

$$x \cos s + y \sin s - p = 0, \quad (1)$$

где $\bar{r}(x, y)$, $\bar{n}(\cos s; \sin s)$.

Уравнение (1) определяет на плоскости семейство касательных к линии. Огибающей линей к этому семейству является линия γ , которую нужно найти.

Дифференцируя уравнение (1) по параметру s , получим

$$-x \sin s + y \cos s - p' = 0. \quad (2)$$

Решая систему

$$\begin{cases} \cos s \cdot x + \sin s \cdot y = p, \\ -\sin s \cdot x + \cos s \cdot y = p', \end{cases}$$

найдем координаты точки касания, т.е. точки искомой линии

$$\begin{aligned} x &= p \cos s - p' \sin s, \\ y &= p \sin s + p' \cos s. \end{aligned} \quad (3)$$

Если для вектора $\bar{n}(\cos s; \sin s)$ использовать значение $\bar{n}'(-\sin s; \cos s)$, то векторную функцию \bar{r} линии, выраженную через опорную функцию $p = p(s)$, можно компактно записать в виде

$$\bar{r} = p\bar{n} + p'\bar{n}',$$

где дифференцирование выполняется по естественному параметру s .

3.13. Проверьте, что для опорной функции:

а) $p = R + a \cos s + b \sin s$ получаем параметрические уравнения окружности $x = R \cos s + a$, $y = R \sin s + b$ с центром в точке $(a; b)$;

б) $p = \sqrt{a^2 \cos^2 s + b^2 \sin^2 s}$ получаем параметрические уравнения эллипса $x = a \cos t$, $y = b \sin t$,

в) $p = -\frac{\cos^2 s}{4a \sin s}$ получаем параметрические уравнения параболы $x = t$, $y = at^2$.

Рассмотрим построение семейства касательных с данной опорной функцией, а также линии γ .

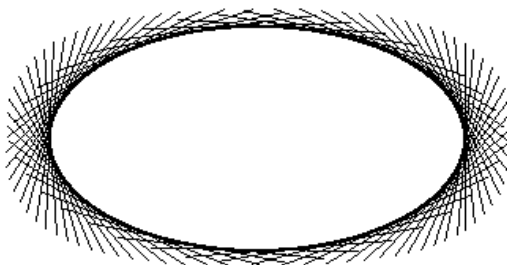


Рис. 7

Программа «Опорная функция» строит точки линии, а затем отрезки касательной длиной $2c$ с центром в точке касания (рис. 7).

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 2000: b = 1200: c = 1000: x0 = 4000: y0 = 4000: pi = 4 * Atn(1)
DrawWidth = 2
For s = 0 To 2 * pi Step 0.001
p = Sqr(a ^ 2 * (Cos(s)) ^ 2 + b ^ 2 * (Sin(s)) ^ 2)
x = a ^ 2 * Cos(s) / p: y = b ^ 2 * Sin(s) / p
PSet (x + x0, y + y0)
Next s
DrawWidth = 1
For s = 0 To 2 * pi Step 0.1
p = Sqr(a ^ 2 * (Cos(s)) ^ 2 + b ^ 2 * (Sin(s)) ^ 2)
x = a ^ 2 * Cos(s) / p: y = b ^ 2 * Sin(s) / p
Line (x - c * Sin(s) + x0, y + c * Cos(s) + y0)-(x + c * Sin(s) + x0, y - c * Cos(s) + y0)
Next s
End Sub
```

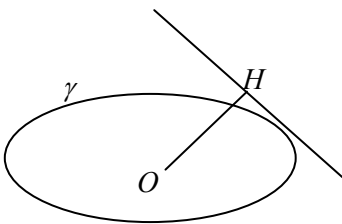


Рис. 8

Пусть H — основание перпендикуляра, опущенного из начала координат на касательную к линии γ . Семейство касательных к линии можно построить, не используя точку касания, а откладывая отрезки длиной c в одну и другую сторону от точки $H(p \cos s; p \sin s)$ (рис. 8).

В этом случае вид семейства касательных получается лучше (рис. 9), т.к. меньше происходит наложений одних касательных на другие, но при малых значениях константы c отрезок касательной может не коснуться самой линии.

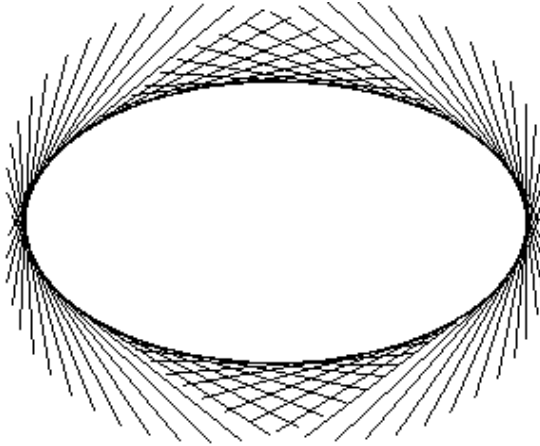


Рис. 9

Программа построения отрезков на касательных с центром в точке H имеет вид:

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 2000: b = 1200: c = 1000: x0 = 4000: y0 = 4000: pi = 4 * Atn(1)
For s = 0 To 2 * pi Step 0.1
p = Sqr(a ^ 2 * Cos(s) ^ 2 + b ^ 2 * Sin(s) ^ 2)
Line (p * Cos(s) - c * Sin(s) + x0, p * Sin(s) + c * Cos(s) + y0)-(p * Cos(s) + c *
Sin(s) + x0, p * Sin(s) - c * Cos(s) + y0)
Next s
DrawWidth = 2
For s = 0 To 2 * pi Step 0.001
p = Sqr(a ^ 2 * (Cos(s)) ^ 2 + b ^ 2 * (Sin(s)) ^ 2)
x = a ^ 2 * Cos(s) / p: y = b ^ 2 * Sin(s) / p
PSet (x + x0, y + y0)
Next s
End Sub
```

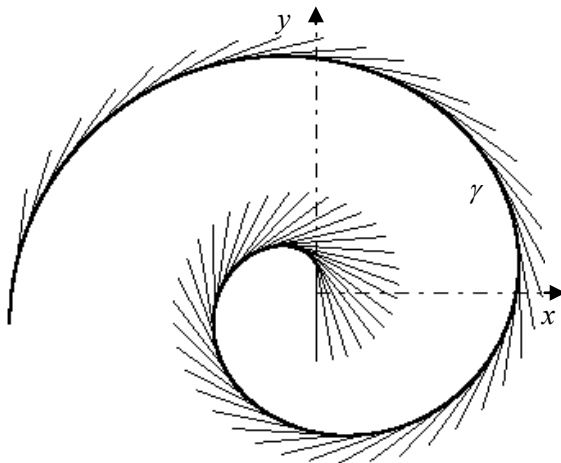


Рис. 10

Для опорной функции $p = as$ на рис. 10 построена спираль γ и семейство отрезков касательных длиной c от точек касания.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
a = 300: c = 1000: x0 = 4000: y0 = 4000: pi = 4 * Atn(1)
DrawWidth = 2
For s = 0 To 3 * pi Step 0.001
p = a * s: p1 = a
x = p * Cos(s) - p1 * Sin(s): y = p * Sin(s) + p1 * Cos(s)
PSet (x + x0, y + y0)
Next s
DrawWidth = 1
For s = 0 To 3 * pi Step 0.15
p = a * s: p1 = a
x = p * Cos(s) - p1 * Sin(s): y = p * Sin(s) + p1 * Cos(s)
Line (x + x0, y + y0)-(x + c * Sin(s) + x0, y - c * Cos(s) + y0)
DrawStyle = 3
Line (x0, y0)-(x0 + 3000, y0): Line (x0, y0)-(x0, y0 + 3000)
End Sub
```

3.14. Выберите определенные значения параметров a , b , k в опорной функции $p = a + b \sin(ks)$, чтобы построить аналоги линий, изображенных на рис. 11, 12.

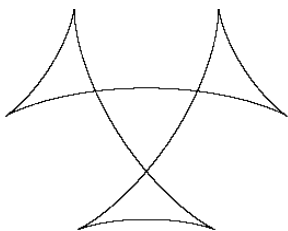


Рис. 11

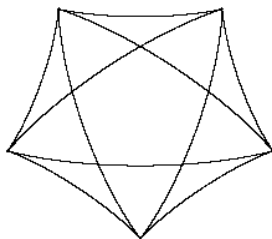


Рис. 12

На рис. 13 показан крутой поворот автомобильной дороги, а на рис. 14 поворот не является крутым. Величина крутизны зависит от радиуса окружности.

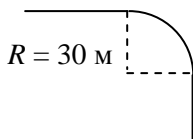


Рис. 13

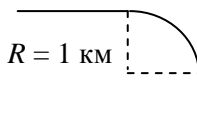


Рис. 14

На рис. 15 представлена крутая горка, а на рис. 16 — пологая горка.



Рис. 15

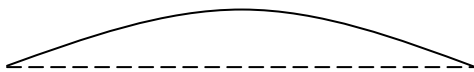


Рис. 16

Форма линии снежной горки может иметь различную крутизну в своих точках (рис. 17).

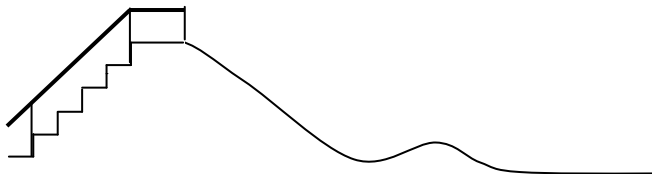


Рис. 17

Для характеристики точек на линии γ существуют две взаимно обратные характеристики: кривизна k и радиус кривизны R , причем $R = 1/k$.

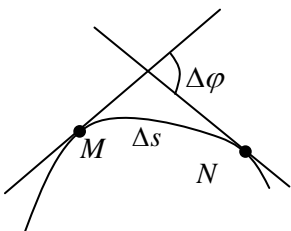


Рис. 18

Определим кривизну линии в точке M (рис. 18). Рассмотрим произвольную точку N на линии. Проведем касательные в этих точках, и пусть $\Delta\varphi$ — угол между касательными, а Δs — длина дуги MN линии. Кривизной линии в точке M называется число, к которому стремится разностное отношение $\Delta\varphi/\Delta s$ при Δs , стремящемся к нулю.

Если использовать определение предела, то можно записать:

$$k = \lim_{\Delta s \rightarrow 0} \frac{\Delta\varphi}{\Delta s}.$$

Используя определение, можно найти кривизну для простейших линий.

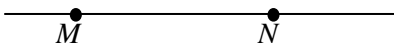


Рис. 19

Для прямой $\Delta\varphi = 0$ (рис. 19), поэтому кривизна прямой в любой точке равна нулю. Радиус кривизны в каждой точке считают бесконечным.

Для окружности угол между касательными равен углу $\angle MON$ между радиусами (рис. 20), а длина дуги окружности равна произведению радиуса на величину центрального угла, опирающегося на эту дугу, т.е. $\Delta s = R \cdot \Delta\varphi$. Для разностного отношения получаем:

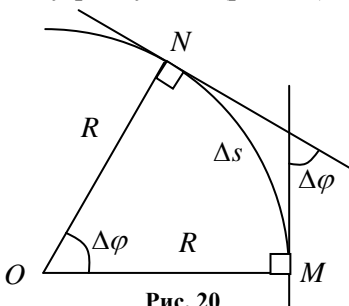


Рис. 20

$$\frac{\Delta\varphi}{\Delta s} = \frac{\Delta\varphi}{R \cdot \Delta\varphi} = \frac{1}{R}.$$

Кривизна окружности в любой точке постоянна и равна величине, обратной радиусу окружности.

Радиус кривизны окружности в любой точке равен радиусу окружности.

Для вычисления кривизны и радиуса кривизны других линий используют следующие формулы:

а) Для линии, заданной уравнением $y = f(x)$, кривизна равна

$$k = \frac{|f''(x)|}{\sqrt{1 + (f'(x))^2}}. \quad (4)$$

б) Для линии, заданной уравнениями $x = x(t)$, $y = y(t)$,

$$k = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{\sqrt{(x'(t))^2 + (y'(t))^2}^3}. \quad (5)$$

в) Для линии, заданной опорной функцией $p = p(s)$, где s — длина дуги единичной окружности, радиус кривизны равен

$$R = |p(s) + p''(s)|.$$

Если начало координат находится внутри замкнутой линии, то $p(s) > 0$ и

$$R = p(s) + p''(s). \quad (6)$$

3.8. Для прямой, заданной уравнением $y = kx + b$, найдите кривизну по формуле (4).

3.9. Для прямой, заданной уравнениями $x = x_0 + mt$, $y = y_0 + nt$, найдите кривизну по формуле (6).

3.10. Для окружности, заданной уравнениями $x = R \cos t$, $y = R \sin t$ или уравнением $p = R$, найдите кривизну и радиус кривизны.

Одной из важнейших задач геометрии является восстановление линии по ее кривизне или радиусу кривизны.

Пусть дана функция $R(s)$, причем $R(s) \geq 0$. Требуется найти линию, радиус кривизны которой равен данной функции.

Если использовать формулы (4), (5), то интегрирование полученного дифференциального уравнения невозможно или является сложной задачей. В этом легко убедиться для самого простого случая $R(s) = R$, т.е. постоянного значения радиуса кривизны. Одну из линий, удовлетворяющих этому условию, мы знаем. Это окружность радиуса R .

Но задача в общем случае быстро разрешима, если использовать опорную функцию.

Для опорной функции получаем дифференциальное уравнение

$$p + p'' = R(s). \quad (7)$$

Для соответствующего однородного дифференциального уравнения $p + p'' = 0$ легко предъявить два частных решения $p = \cos s, p = \sin s$.

Общее решение уравнения будем искать в виде $p = C_1(s) \cos s + C_2(s) \sin s$, где $C_1(s), C_2(s)$ — неизвестные пока функции.

Дифференцируя это равенство, получим

$$p' = C_1'(s) \cos s - C_1(s) \sin s + C_2'(s) \sin s + C_2(s) \cos s.$$

На две функции $C_1(s), C_2(s)$ наложим первое условие

$$C_1'(s) \cos s + C_2'(s) \sin s = 0, \quad (8)$$

тогда получаем $p' = -C_1(s) \sin s + C_2(s) \cos s$.

Дифференцируя снова, получаем

$$p'' = -C_1'(s) \sin s - C_1(s) \cos s + C_2'(s) \cos s - C_2(s) \sin s.$$

Подставим функции p, p'' в уравнение (7) и найдем второе условие на функции $C_1(s), C_2(s)$

$$-C_1'(s) \sin s + C_2'(s) \cos s = R(s). \quad (9)$$

Решая систему для уравнений (8) и (9)

$$\begin{cases} C_1'(s) \cos s + C_2'(s) \sin s = 0, \\ -C_1'(s) \sin s + C_2'(s) \cos s = R(s). \end{cases},$$

найдем производные неизвестных функций

$$C_1'(s) = -R(s) \sin s, C_2'(s) = R(s) \cos s.$$

Для функции $C_1(s), C_2(s)$ получаем равенства

$$C_1(s) = -\int R(s) \sin s \, ds, C_2(s) = \int R(s) \cos s \, ds. \quad (10)$$

Итак, общее решение уравнения (7) имеет вид:

$$p = -\cos s \int R(s) \sin s \, ds + \sin s \int R(s) \cos s \, ds. \quad (11)$$

Если $R(s) = \text{const} = R$, то получаем общее решение:

$$p = R + C_1 \cos s + C_2 \sin s.$$

Эта опорная функция задает семейство окружностей радиуса R , зависящее от двух параметров, т.е. окружности с точностью до параллельного переноса на любой вектор плоскости.

Применим формулу интегрирования по частям

$$\int u \, dv = uv - \int v \, du$$

для равенства (11), тогда

$$\begin{aligned} \int R(s) \sin s \, ds &= \left(u = R(s), du = R'(s) \, ds \right. \\ &\quad \left. dv = \sin s \, ds, v = -\cos s \right) = \\ &= -R(s) \cos s + \int R'(s) \cos s \, ds, \\ \int R(s) \cos s \, ds &= \left(u = R(s), du = R'(s) \, ds \right. \\ &\quad \left. dv = \cos s \, ds, v = \sin s \right) = \\ &= R(s) \sin s - \int R'(s) \sin s \, ds, \\ p &= R(s) - \cos s \int R'(s) \cos s \, ds - \sin s \int R'(s) \sin s \, ds. \end{aligned} \quad (12)$$

Интегрируя снова по частям, получим

$$p = R(s) + \cos s \int R''(s) \sin s \, ds - \sin s \int R''(s) \cos s \, ds. \quad (13)$$

3.15. Для функции $R(s) = as$ найдите опорную функцию и сравните с линией, изображенной на рис. 10.

3.16. Кривизной вершины многоугольника называется величина внешнего угла многоугольника в этой вершине.

Сумма кривизн всех вершин замкнутого выпуклого плоского многоугольника равна 2π .

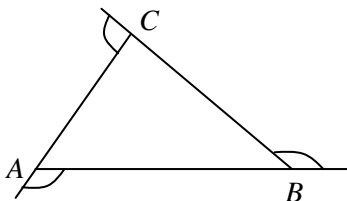


Рис. 21

Действительно, рассмотрим развернутые углы в вершинах замкнутого выпуклого n -угольника, которые содержат этот многоугольник. Сумма величин всех развернутых углов равна сумме величин углов данного многоугольника и сумме кривизн всех вершин, т.е.

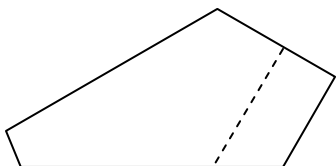


Рис. 22

$$n\pi = (n-2)\pi + \sum_{i=1}^n k_i .$$

Следовательно, $\sum_{i=1}^n k_i = 2\pi$.

Если решать задачу о восстановлении n -угольника с данными кривизнами вершин k_1, k_2, \dots, k_n , где кривизны удовлетворяют условию $\sum_{i=1}^n k_i = 2\pi$, то задача имеет бесконечно много различных решений. Поясните это высказывание, используя рис. 22.

Если дополнительно задать длины некоторых сторон многоугольника, то задача может иметь единственное решение с точностью до движений плоскости.

Задание для проекта. 1. Корректно сформулируйте задачу о восстановлении замкнутого выпуклого плоского многоугольника с данными кривизнами вершин и данными длинами сторон. Напишите компьютерную программу построения многоугольника с этими параметрами. В качестве практического приложения данного проекта установите связь этой задачи с проблемой поиска «клада» при спортивном ориентировании.

2. Изучите кручение ломаной линии в пространстве [19, с. 122], постройте бильярдную траекторию на поверхности куба и определите кривизну и кручение этой ломаной в вершинах.

3. Изучите определение кривизны вершины многогранника [1] и познакомьтесь в научной литературе с решением проблемы о восстановлении многогранника с данными кривизнами вершин.

§ 25. Матрица с доминирующей диагональю

При изучении линейной алгебры студенты увлечены операциями с матрицами и определителями и затрудняются в приведении примеров по использованию матричной алгебры. В методике преподавания математики и в психологии встречаются матрицы, основное направление использования которых ограничивается удобной формой представления информации в таблице. Изучение свойств специальных матриц подобного типа можно осуществить в курсе по выбору. Назрела необходимость создания банка типов специальных матриц, которые используются в педагогических и психологических исследованиях. Изучение таких матриц будет способствовать как формированию интереса к линейной алгебре, так и научному обоснованию психолого-педагогических исследований. Тематика учебных исследований по линейной алгебре актуальна тем, что исследования можно выстраивать с первого курса обучения в вузе.

В линейной алгебре элементы матрицы являются математическими объектами (числа, математические выражения). В психологии большое значение придается интерпретации элементов матрицы для количественной оценки индивидуальности. Для вопросов управления коллективом и направления его на решение определенных задач большое значение имеют интегративные (кумулятивные) характеристики. Оперирование интегративными характеристиками значительно упрощает переработку информации и позволяет представить объекты в целом, опуская некоторые детали.

В образовании интегративные характеристики можно использовать для укрупнения дидактических единиц.

Если перед коллективом поставлена новая задача, то для ее решения необходимо сформировать группу по некоторым критериям для решения задачи. Среди намеченных кандидатур полезно провести тестирование по оценке мнения каждого планируемого участника относительно потенциальных возможностей не только остальных участников, но и самого себя.

Пусть k -ый участник в группе из n человек представил оценки $(a_{k1}, a_{k2}, \dots, a_{kk}, \dots, a_{kn})$. Объединив все оценки в одну таблицу, получим матрицу

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{pmatrix} \begin{matrix} \text{вектор высказываний } 1^{\text{го}} \text{ участника} \\ \text{вектор высказываний } 2^{\text{го}} \text{ участника} \\ \\ \text{вектор высказываний } i^{\text{го}} \text{ участника} \\ \\ \text{вектор высказываний } n^{\text{го}} \text{ участника} \end{matrix} \quad (1)$$

Рассматривая совокупность чисел в j -м столбце, получим вектор-оценку конкретного участника остальными участниками.

Таким образом, для строк устанавливается свойство, имеющее интегративный смысл.

В дальнейшем такую матрицу будем называть матрицей отношений в группе с завышенной самооценкой. Аналогичную матрицу можно получить, если в матрице представить оценку личностного отношения члена коллектива к остальным членам коллектива.

В настоящей статье выделяется класс квадратных матриц, обладающих свойством: в каждой строке элемент на главной диагонали матрицы больше суммы модулей остальных элементов этой строки. Их можно интерпретировать как высказывание мнений членами группы, в которой каждый член группы оценивает свои способности выше, чем способности всех остальных членов группы. Для матрицы такого типа доказано, что ее определитель больше нуля.

Рассмотрим другие примеры аналогичных матриц.

Имеется сеть банков. Для страхования риска каждый банк размещает часть своих активов в других банках. Естественное ограничение состоит в том, что активы любого банка, размещенные в собственном банке, больше, чем активы, размещенные в других банках.

Имеется ряд государств, которые поставляют продукцию как на внутренний рынок, так и на внешний рынок. Объем продукции, поставляемый на внутренний рынок, в основном больше объема продукции, поставляемой на экспорт. Матрицу такого типа можно назвать матрицей производственных отношений, ориентированных в основном на внутренний рынок.

В линейной алгебре такие матрицы называются *матрицами с доминирующей диагональю*. Рассмотрим опыт изучения свойства определителя матрицы с доминирующей диагональю.

Для каждого вектора высказываний и каждого вектора оценки можно ввести норму, тогда получим по одной числовой характеристике на каждого члена коллектива: норму высказываний и норму оценки. Существует ряд видов деятельности, где одна ошибка может существенно повлиять на оценку компетентности специалиста. Например, управление полетами самолетов, врачебная практика, судопроизводство и т.д. В этом случае норма должна существенно учитывать любое отклонение от нормативов деятельности. В этом случае лучше использовать для задания нормы максимальное значение или сумму модулей отклонений. В других видах деятельности некоторые ошибки можно сгладить и в этом случае можно использовать обычную евклидову норму.

Для задания элементов матрицы можно использовать различные шкалы оценивания. Достаточно полная и часто употребляемая в практике тестирования шкала предполагает ответы: нет; скорее нет, чем да; в равной мере, как нет, так и да; скорее да, чем нет; да.

Выбирая единицу измерения, равную числу 1 для ответа «да» и исходя из равномерного распределения величины оценивания, получаем числовые характеристики дискретной величины X для ответа: нет — «0»; скорее нет, чем да — «0,25»; в равной мере, как нет, так и да — «0,5»; скорее да, чем нет — «0,75»; да — «1».

Пример матрицы отношений при такой системе оценивания:

$$\begin{pmatrix} 1 & 0,5 & 0,25 \\ 0 & 0,5 & 0,25 \\ 0,25 & 0,75 & 1 \end{pmatrix} \quad (2)$$

Если при оценивании следует наглядно представить положительное и отрицательное мнение, то система оценивания использует величины: нет: «-2»; скорее нет, чем да: «-1»; в равной мере, как нет, так и да: «0»; скорее да, чем нет: «1»; да: «2».

Те же отношения, оцененные ранее матрицей (2), в этой системе оценивания представлены следующей матрицей:

$$\begin{pmatrix} 2 & 0 & -1 \\ -2 & 0 & -1 \\ -1 & 1 & 2 \end{pmatrix}. \quad (3)$$

Если предложить учащимся пятого класса дать оценку способности членов класса решить уравнение $\ln(e + x^2) = \sin x$, то очевидно, что все элементы матрицы будут равны (-2).

Если предложить преподавателям кафедры математики дать оценку способности членов коллектива решить уравнение $2x = 6$, то очевидно, что все элементы будут равны 2.

Таким образом, величины элементов матрицы отношений зависят от уровня образованности, от уровня сложности поставленной задачи для тестирования и многих других характеристик.

Между элементами матриц (2), (3) можно установить взаимно однозначное соответствие

$$0 \rightarrow -2; 0,25 \rightarrow -1; 0,5 \rightarrow 0; 0,75 \rightarrow 1; 1 \rightarrow 2.$$

Аналитически изоморфизм задается формулами:

$$u = 4x - 2 \text{ или } x = \frac{u + 2}{4}.$$

Интересно изучить некоторые типы матриц отношений.

Иногда группа может сформироваться таким образом, что каждый член коллектива имеет завышенную оценку мнения о себе и не замечает достижений других членов коллектива.

Матрицей отношений типа D_{pr} назовем квадратную матрицу, если в каждой строке элемент главной диагонали больше суммы модулей всех остальных элементов в этой строке. Матрица (1) удовлетворяет условию:

$$a_{ii} > |a_{i1}| + |a_{i2}| + \dots + |a_{ii-1}| + |a_{ii+1}| + \dots + |a_{in}|, \quad \forall i.$$

Нижние индексы в этой и других матрицах являются первыми буквами в словах: principal — главный, row — строка, secondary — второстепенный, column — столбец.

Из определения матрицы следует, что все элементы на главной диагонали являются положительными числами.

Следом матрицы называется сумма элементов матрицы, расположенных на главной диагонали. След матрицы A обозначается trA . Для матриц типа D_{pr} выполняется неравенство:

$$trA > \sum_{\substack{i,j=1 \\ i \neq j}}^n |a_{ij}|.$$

Неравенство можно интерпретировать следующим образом — сумма сомнений членов коллектива больше суммы мнений членов коллектива о других члена этого коллектива.

Определитель диагональной матрицы равен произведению элементов на главной диагонали. Матрица типа D_{pr} «близка» в некотором смысле к диагональной матрице с положительными диагональными элементами, поэтому естественно ожидать некоторую аналогию в определителе матрицы типа D_{pr} по отношению к определителю диагональной матрицы.

В дальнейшем будем использовать следующую очевидную лемму.

Лемма 1. Пусть числа a_1, a_2, \dots, a_k удовлетворяют условию

$$|a_1| + |a_2| + \dots + |a_k| < 1 \quad (4)$$

и пусть каждому числу $a_i, i = 1, 2, \dots, k$ поставлено в соответствие некоторое число β_i , равное 1 или (-1), тогда выполняется неравенство

$$|\beta_1 a_1 + \beta_2 a_2 + \dots + \beta_k a_k| < 1. \quad (5)$$

Замечание. Лемма означает, что если выполняется неравенство (1) для суммы модулей величин, то будет выполняться аналогичное неравенство для алгебраической суммы этих величин при произвольной расстановке знаков перед слагаемыми.

Лемма доказывается на основе неравенств $a + b \leq |a| + |b|$, $a - b \leq |a| + |b|$.

Лемма 2. Матрица второго порядка типа D_{pr} имеет положительный определитель.

Доказательство. Пусть диагональные элементы матрицы

$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ являются положительными числами и удовлетворяют условию $a > |b|$, $d > |c|$.

Знак определителя матрицы не изменится, если элементы каждой строки разделить на диагональный элемент этой строки, тогда

$$\Delta_2 = \begin{vmatrix} 1 & p \\ q & 1 \end{vmatrix} = 1 - pq, \Delta_2 > 0, \text{ т.к. } 1 > |p|, 1 > |q|.$$

Замечание. Знак определителя можно легко установить непосредственным подсчетом, но мы использовали переход к вспомогательному определителю, чтобы упростить в дальнейшем вычисления для определителей более высокого порядка.

Лемма 3. Матрица третьего порядка типа D_{pr} имеет положительный определитель.

Доказательство леммы достаточно провести для матрицы

$$\begin{pmatrix} 1 & a & b \\ c & 1 & d \\ e & f & 1 \end{pmatrix}, \text{ где } a, b, c, d, e, f \text{ — дробные числа и выполняются условия } 1 > |a| + |b|, 1 > |c| + |d|, 1 > |e| + |f|.$$

Используя элементарные преобразования, получим

$$\begin{vmatrix} 1 & a & b \\ c & 1 & d \\ e & f & 1 \end{vmatrix} \xrightarrow{\substack{(-c) \\ (-e)}} \begin{vmatrix} 1 & a & b \\ 0 & 1 - ac & d - bc \\ 0 & f - ae & 1 - be \end{vmatrix} = \begin{vmatrix} 1 - ac & d - bc \\ f - ae & 1 - be \end{vmatrix}.$$

Матрица полученного определителя второго порядка является матрицей типа D_{pr} . Проверим, например, для первой строки $1 - ac - |d - bc| = 1 - ac + \gamma(d - bc) = 1 + \gamma d - c(a + \gamma b)$, где γ — знак, полученный при снятии модуля.

Элементы первой строки матрицы третьего порядка типа D_{pr} удовлетворяют условию $|a| + |b| < 1$. По лемме 1 получаем $|a + \gamma b| < 1$, поэтому $|c(a + \gamma b)| < |c|$.

Элементы второй строки матрицы третьего порядка типа D_{pr} удовлетворяют условию $|c|+|d|<1$, поэтому $|c(a+\gamma b)|+|d|<1$. По лемме 1 получаем $|\gamma d - c(a+\gamma b)|<1$.

Следовательно, $1 - ac - |d - bc| > 0$.

Лемма доказана.

Лемму 3 можно геометрически интерпретировать следующим образом. Значение определителя можно представить как значение смешанного произведения для векторов $\bar{a}(1;a;b)$, $\bar{b}(c;1;d)$, $\bar{c}(e;f;1)$. Отложим векторы из начала декартовой системы координат, тогда концевые точки расположены, соответственно, в плоскостях $x=1$, $y=1$, $z=1$.

Неравенства $|a|+|b|<1$, $|c|+|d|<1$, $|e|+|f|<1$ определяют в этих плоскостях, соответственно, множества внутренних точек многоугольников $KEFG$, $MNPF$, $PQRG$. Векторы

$\bar{a}(1;a;b)$, $\bar{b}(c;1;d)$, $\bar{c}(e;f;1)$, концевые точки которых расположены внутри этих квадратов, образуют правую тройку и смешанное произведение этих векторов положительно. Для проверки утверждения о правой тройке векторов можно рассмотреть случаи, когда векторы оказываются

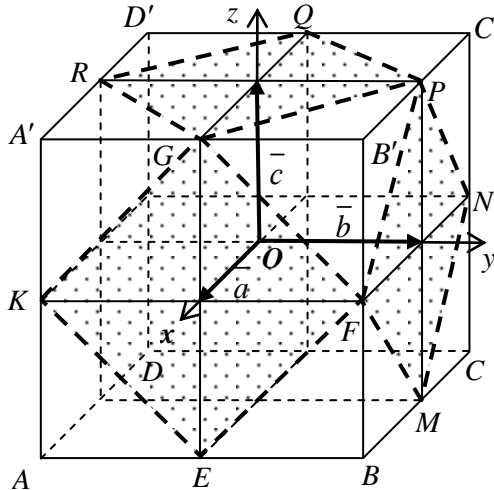


Рис. 1

в вершинах квадратов, а потом, используя непрерывность значения смешанного произведения в зависимости от непрерывного изменения координат, получить общий случай.

Леммы 2 и 3 дают основания для выдвижения гипотезы о том, что определитель матрицы любого порядка является положительным числом.

Теорема 1. Матрица типа D_{pr} порядка k , где $k \geq 2$ имеет положительный определитель.

Докажем утверждение методом математической индукции.

Для матриц второго и третьего порядка утверждение справедливо. Предположим, что оно справедливо для матрицы $k-1$ порядка. Докажем утверждение для матрицы порядка k . Поделив элементы каждой строки на соответствующий диагональный элемент этой строки, получим определитель матрицы типа D_{pr} с диагональными элементами, равными 1.

$$\Delta_k = \begin{vmatrix} 1 & p_{12} & \dots & p_{1k} \\ p_{21} & 1 & \dots & p_{2k} \\ \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{kk} \end{vmatrix}, \text{ где } 1 - \sum_{\substack{i=1 \\ i \neq j}}^k |p_{ij}| > 0.$$

Применяя элементарные преобразования к строкам, получим

$$\Delta_k = \begin{vmatrix} 1 & p_{12} & \dots & p_{1k} \\ 0 & 1 - p_{12}p_{21} & \dots & p_{2k} - p_{1k}p_{21} \\ \dots & \dots & \dots & \dots \\ 0 & p_{k2} - p_{12}p_{k1} & \dots & 1 - p_{1k}p_{k1} \end{vmatrix}.$$

Докажем, что следующая матрица

$$\begin{pmatrix} 1 - p_{12}p_{21} & p_{23} - p_{13}p_{21} & \dots & p_{2k} - p_{1k}p_{21} \\ \dots & \dots & \dots & \dots \\ p_{k2} - p_{1k}p_{k1} & p_{k3} - p_{1k}p_{k1} & \dots & 1 - p_{1k}p_{k1} \end{pmatrix}$$

принадлежит типу D_{pr} .

Например, рассмотрим разность для элементов первой строки

$$1 - p_{12}p_{21} - |p_{23} - p_{13}p_{21}| - \dots - |p_{2k} - p_{1k}p_{21}|.$$

Пусть знак выражения $p_{2i} - p_{1i}p_{21}$ равен γ_i , тогда разность равна

$$1 - p_{12}p_{21} - \gamma_3(p_{23} - p_{13}p_{21}) - \gamma_4(p_{24} - p_{14}p_{21}) - \dots - \gamma_k(p_{2k} - p_{1k}p_{21}) =$$

$$= 1 - \gamma_3 p_{23} - \gamma_4 p_{24} - \dots - \gamma_k p_{2k} + p_{21}(-p_{12} + \gamma_3 p_{13} + \gamma_4 p_{14} + \dots + \gamma_k p_{1k}).$$

Элементы первой строки матрицы k -го порядка типа D_{pr} удовлетворяют условию

$$|p_{12}| + |p_{13}| + \dots + |p_{1k}| < 1.$$

По лемме 1 получаем $|-p_{12} + \gamma_3 p_{13} + \gamma_4 p_{14} + \dots + \gamma_k p_{1k}| < 1$, следовательно $|p_{21}(-p_{12} + \gamma_3 p_{13} + \gamma_4 p_{14} + \dots + \gamma_k p_{1k})| < |p_{21}|$.

Используя оценку $|p_{21}| + |p_{23}| + |p_{24}| + \dots + |p_{2k}| < 1$ для элементов второй строки, получаем

$$|p_{21}(-p_{12} + \gamma_3 p_{13} + \gamma_4 p_{14} + \dots + \gamma_k p_{1k})| + |p_{23}| + |p_{24}| + |p_{2k}| < 1.$$

По лемме 1 следует оценка

$$\gamma_3 p_{23} + \gamma_4 p_{24} + \dots + \gamma_k p_{2k} - p_{21}(-p_{12} + \gamma_3 p_{13} + \gamma_4 p_{14} + \dots + \gamma_k p_{1k}) < 1$$

и необходимое свойство матрицы $k - 1$ порядка доказано, а следовательно, и доказана теорема.

Каждой симметричной квадратной матрице можно поставить в соответствие квадратичную форму.

Следствие. Квадратичная форма с матрицей типа D_{pr} является положительно определенной.

Доказательство следует из критерия Сильвестра, т.к. все угловые миноры для матрицы типа D_{pr} являются положительными.

Следствие показывает, что значение квадратичной формы на любом ненулевом векторе высказываний участников группы принимает положительное значение.

Матрицей отношений типа D_{pc} назовем квадратную матрицу, если в каждом столбце элемент главной диагонали больше суммы модулей всех остальных элементов в этом столбце.

Теорема 2. Матрица типа D_{pc} порядка k , где $k \geq 2$ имеет положительный определитель.

Доказательство. Транспонированная матрица D'_{pc} к матрице D_{pc} является матрицей типа D_{pr} . При транспонировании матрицы определитель является инвариантом, поэтому по теореме 1 матрица D_{pc} имеет положительный определитель.

Матрицей отношений типа D_{tr} назовем квадратную матрицу, если в каждой строке элемент второстепенной диагонали больше суммы модулей всех остальных элементов в этой строке.

Матрицей отношений типа D_{ic} назовем квадратную матрицу, если в каждом столбце элемент второстепенной диагонали больше суммы модулей всех остальных элементов в этом столбце.

Теорема 3. Матрицы типа D_{tr} , D_{tc} порядка k , где $k \geq 2$, имеют положительный определитель, если целая часть отношения $k/2$, т.е. $\lfloor k/2 \rfloor$ является четным числом и имеет отрицательный определитель, если $\lfloor k/2 \rfloor$ является нечетным числом.

Доказательство. Рассмотрим матрицу типа D_{tr} . Применяя $k/2$ раз перестановку столбцов, симметрично расположенных относительно серединного столбца в данной матрице, получим матрицу типа D_{pr} . Определители этих матриц отличаются множителем $(-1)^{\lfloor k/2 \rfloor}$. Аналогично доказывается теорема для матриц типа D_{tc} .

Замечание. Определитель диагональной матрицы равен произведению диагональных элементов. Если все диагональные элементы отличны от нуля, то определитель такой матрицы отличен от нуля. Можно ли все элементы, расположенные вне главной диагонали, незначительно изменить, так чтобы знак определителя новой матрицы совпал со знаком определителя диагональной матрицы?

Рассмотрим значение определителя как функцию элементов соответствующей матрицы. Эта функция является непрерывной функцией от n^2 переменных.

Используя непрерывность функции для диагональной матрицы с диагональными элементами, отличными от нуля, можно так изменить элементы, расположенные вне главной диагонали, на величину, бесконечно малую по сравнению с диагональными элементами, что знак определителя этой матрицы совпадет со знаком произведения элементов на главной диагонали.

Выделим интересную проблему: как можно изменить нулевые элементы диагональной матрицы, расположенные вне главной диагонали, чтобы знак диагональной матрицы сохранился.

Теоремы 1—3 в некотором смысле близки к очевидному утверждению о сохранении знака при бесконечно малом изменении элементов, расположенных вне главной диагонали.

Справочный материал

I. Операторы в Visual Basic 6

Способы задания цветов:

QBColor (0) или vbBlack — черный;
QBColor (1) — темно-синий;
QBColor (2) — темно-зеленый;
QBColor (3) — темно-голубой;
QBColor (4) — темно-красный;
QBColor (5) — темно-сиреневый;
QBColor (6) — коричневый;
QBColor (7) — светло-серый;
QBColor (8) — темно-серый;
QBColor (9) или vbBlue — синий;
QBColor (10) или vbGreen — зеленый;
QBColor (11) или vbCyan — голубой;
QBColor (12) или vbRed — красный;
QBColor (13) или vbMagenta — сиреневый;
QBColor (14) или vbYellow — желтый;
QBColor (15) или vbWhite — белый.

Значения свойств DrawStyle и DrawWidth:

DrawStyle = 0 — сплошная заливка;
DrawStyle = 1 — штриховая линия;
DrawStyle = 2 — пунктирная линия;
DrawStyle = 3 — штрихпунктирная линия;
DrawStyle = 4 — штрих двойной пунктир;
DrawStyle = 5 — прозрачная заливка линии;
DrawWidth = i — толщина линии.

По умолчанию DrawWidth = 1.

CLS — оператор, очищающий экран.

Form1.Scale (0, Form1.Height)-(Form1.Width, 0) — оператор перехода на экране компьютера к декартовой системе координат.

pi =4*ATN (1) — вычисление числа π в программе.

LINE $(x_1, y_1) - (x_2, y_2)$, \dots , B | BF , $\&H000$ — оператор
коорд. противоположных вершин цвет прямоугольник закрашка прямоугольника штрих – линия

рисования отрезка (прямоугольника при указании B)

CIRCLE (x , y) , \dots , \dots , \dots , \dots , \dots — оператор
координаты центра радиус цвет дуги начало дуги конец дуги коэффициент сжатия

рисования окружности (дуги эллипса при указании начала и конца дуги).

Символ	Значение	Символ	Значение
+	сложение	COS (x)	cos x
—	вычитание	SIN (x)	sin x
*	умножение	TAN (x)	tg x
/	деление	ATN (x)	arc tg x
\	целочисленное деление	EXP (x)	e ^x
^	возведение в степень	LOG (x)	ln x
MOD	нахождение остатка при делении	SQR (x)	\sqrt{x}
		ABS (x)	x

PSET (x,y) — оператор, рисующий точку (x, y) на экране.

PSet (x, y), QBColor(15): Print "b="; Int(b) — оператор, устанавливающий точку белым цветом с координатами x, y на экране компьютера и оператор, печатающий рядом целую часть значения переменной b.

Print s — оператор печати значения переменной s.

FOR i =1 **TO** 10 **STEP** 0.1 — начинается цикл

.....
NEXT i — закрывается цикл

— оператор цикла по переменной i, принимающей значение от 0 до 10 с шагом изменения 0,1.

IF x>0 **THEN** a = \sqrt{x}

ELSE

PRINT *значение квадратного корня не существует*

— условный оператор, который при выполнении условия x>0 вычисляет значение квадратного корня из неотрицательного числа или сообщает о невозможности вычисления квадратного корня из отрицательного числа.

Do While <Условие> — начало цикла

<Операторы>

Loop — окончание цикла

в) **Do Until** <Условие>

<Операторы>

Loop

г) **Do**

<Операторы>

Loop While <Условие>

д) **Do**

<Операторы>

Loop Until <Условие>

Print Screen (клавиша на клавиатуре) — копирование содержания экранного окна в буфер.

Byte — короткое неотрицательное число, занимает 1 байт памяти,
Integer — целое число, занимает 2 байта памяти,
Long — длинное целое число, занимает 4 байта памяти,
Single — десятичное число обычной точности, занимает 4 байта памяти,
Double — десятичное число двойной точности, занимает 8 байт памяти,
Currency — десятичное протяженное число, занимает 8 байт памяти,
String — строка, память зависит от числа символов в строке,
Variant — произвольное значение, занимает 16 байт памяти,
Boolean — логическая величина, занимает 2 байта памяти.

End Sub

II. Процедуры в Pascal

SetLineStyle(i,1,1) — выбор типа линии.

Более подробно об операторах языка Pascal можно прочесть в пособии [13].

III. Процедуры в исполнителе Чертежник

Rectangle (, , ,) — построение прямоугольника.

Дуга (, , , ,)

Arc (, , , ,) — построение дуги, для которой указываются координаты двух углов прямоугольника, внутри которого расположена дуга, а также начальное и конечное значения углов конечных точек.

Для формирования цвета используется функция

RGB (red, green, blue)

Команда для установки свойств кисти имеет английский и русский варианты

Brush (Color, Style);

Кисть (Color, Style);

Эта процедура принимает три параметра:

Color Числовое значение цвета, полученного с помощью функции **RGB**, например, **RGB(10, 20, 40)**.

Style Стиль кисти, который выбирается из следующих значений:

- | | |
|----------------------------|----------------------------|
| 0 сплошная заливка | 4,5 диагональная штриховка |
| 1 нет заливки | 6 сетчатая штриховка |
| 2 горизонтальная штриховка | 7 диагональная сетка |
| 3 вертикальная штриховка | |

Пример

Brush (**RGB**(255,0,0), 2); // красная кисть

Кисть (**RGB** (0,0 , 0) ,2); // горизонтальная

Заливка (x+x0 ,y+y0); } — указание координат точки, из которой начинается заливка.

Более подробно об операторах исполнителя можно прочесть в пособии [4].

IV. Процедуры и операторы в Pascal ABC

Line (X1,Y1,X2,Y2) — построение отрезка между точками (x₁,y₁) и (x₂,y₂).

Circle (X,Y,R) — построение окружности с центром (x, y) и радиусом R.

Ellipse (X1,Y1,X2,Y2) — эллипс, заданный описанным прямоугольником с противоположными вершинами (x₁,y₁) и (x₂,y₂).

SetPixel(x,y,color) — заливает цветом color точку (x,y).

SetPenStyle(ps...) — устанавливает стиль пера, задаваемый параметром ps:

psSolid — сплошная линия,

psDash — штриховая линия,

psDot — пунктирная линия,

psDashDot — штрихпунктирная линия,

psDashDotDot — штрих, два пунктира,

psClear — отсутствие линии.

TexOut(x,y,text) — выводит text на экран. Начало текста в точке (x,y).

Arc(x,y,r,a1,a2) — рисует дугу окружности с центром (x,y), радиусом r, заключенную между лучами, образующими углы a1 и a2 с осью Oх (a1 и a2 — вещественные, задаются в градусах и отсчитываются против вращения часовой стрелки).

Pie(x,y,r,a1,a2) — рисует сектор окружности, ограниченный дугой (параметры те же, что и в процедуре Arc).

Chord(x,y,r,a1,a2) — рисует сегмент между дугой окружности и отрезком, соединяющим ее концы (параметры те же, что и в процедуре Arc).

SetPenWidth(n) — устанавливает толщину линии, n — натуральное.

SetPenColor(color) — задает цвет линии.

FloodFillColor(color) — заливает область цветом color, начиная с точки (x, y).

SetBrushColor(color) — устанавливает цвет кисти, задаваемый параметром color.

bsSolid — сплошная заливка области,

bsClear — очищает область,

bsHorizontal — покрытие области горизонтальными линиями,

bsVertical — покрытие области вертикальными линиями,

bsCross — сеть из горизонтальных и вертикальных линий,

bsDiagCross — сеть из двух диагональных направлений,

bsBDiagonal — диагонали из правого верхнего угла в левый нижний,

bsFDiagonal — диагонали из левого верхнего угла в правый нижний,

SetWindowSize(w,h) — устанавливает ширину и высоту графического окна.

Стандартные цвета:

clBlack — черный,

clPurple — фиолетовый,

clWhite — белый,

clMaroon — темно-красный,

clGreen — зеленый,

clBrown — коричневый,

clBlue — синий,

clSkyBlue — голубой,

clYellow — желтый,

clCream — кремовый.

clAqua — бирюзовый,

clOlive — оливковый,

clFuchsia — сиреневый,

clTeal — сине-зеленый,

clMoneyGreen — цвет зеленых денег,

clLtGray — светло-серый,

clDkGray — темно-серый,

clMedGray — серый,

clSilver — серебристый,

Имя	Тип	Значение
Abs (x)	x- integer, real	$ x $
Sqr (x)	x- integer, real	x^2
Sqrt (x)	x- real	\sqrt{x}
Power (x,y)	x- real	x^y
Round (x)	x- real	округляет x до ближайшего целого числа
Int (x)	x- real	целая часть числа x
Frac (x)	x- real	дробная часть числа x

Ответы

```
1.10. Private Sub Command1_Click()  
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)  
x0 = 4000: y0 = 3000: c = 800: a2 = c - 100  
For i = 1 To 6  
For x = a2 To 2000 Step 0.2  
y = Sqr((c ^ 2 - a2 ^ 2) * (x ^ 2 / a2 ^ 2 - 1))  
PSet (x + x0, y + y0): PSet (x + x0, -y + y0)  
PSet (-x + x0, y + y0): PSet (-x + x0, -y + y0)  
Line (x0 - c, y0)-(x0 + c, y0): Next x  
a2 = a2 - 100  
Next i  
For R = 2 To 24 Step 2  
Circle (x0 + c, y0), R: Circle (x0 - c, y0), R  
Next R: End Sub
```

1.11. «Эллипсно-гиперболическая система координат»

```
Private Sub Command1_Click()  
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)  
c = 1000: x0 = 5000: y0 = 3500: pi = 4 * Atn(1)  
Line (-3000 + x0, y0)-(3000 + x0, y0): Line (x0, -3000 + y0)-(x0, 3000 + y0)  
For a1 = 1100 To 2400 Step 400  
For t = 0 To 2 * pi Step 0.001  
x = a1 * Cos(t): y = Sqr(a1 ^ 2 - c ^ 2) * Sin(t): PSet (x + x0, y + y0)  
Next t, a1  
For a2 = 100 To 950 Step 200  
For y = 0 To 3000  
x = a2 * Sqr(1 + y ^ 2 / (c ^ 2 - a2 ^ 2))  
PSet (x + x0, y + y0): PSet (x + x0, -y + y0)  
PSet (-x + x0, y + y0): PSet (-x + x0, -y + y0)  
Next y, a2  
For r = 10 To 20  
Circle (c + x0, y0), r: Circle (-c + x0, y0), r  
Next r  
End Sub
```

```
1.12. Private Sub Command1_Click()  
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)  
pi = 4 * Atn(1): xc = 500: yc = 1000  
For u = 0 To 1 Step 0.1  
For t = 0 To 2000  
PSet (xc -(1 - u) * 400 * Sin(2 * pi * t / 2000) + u * (400 * Sin(2 * pi * t / 2000) +  
4000) xc, t + yc)  
Next t, u  
DrawWidth = 2
```

```

For u = 0 To 1
For t = 0 To 2000
PSet (-(1 - u) * 400 * Sin(2 * pi * t / 2000) + u * (400 * Sin(2 * pi * t / 2000) + 4000)
+ xc, t + yc)
Next t, u
End Sub

```

```

1.22. Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
xc = 4000: yc = 2000: a = 1 / 500
For h = -800 To 800 Step 200
For t = -1000 To 1000
x = t + 2 * a * h * t / Sqr(1 + (2 * a * t) ^ 2)
y = a * t ^ 2 - h / Sqr(1 + (2 * a * t) ^ 2)
PSet (x + xc, y + yc): Next t, h
DrawWidth = 2
For t = -1000 To 1000
x = t: y = a * t ^ 2: PSet (x + xc, y + yc): Next t
End Sub

```

```

1.23.
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
xc = 4000: yc = 3000: a = 1 / 100000
DrawWidth = 2
For t = -500 To 500
x = t: y = a * t ^ 3: PSet (x + xc, y + yc): Next t
DrawWidth = 1
For t = -500 To 500 Step 40
x = t: y = a * t ^ 3: Circle (x + xc, y + yc), 200: Next t
For h = -200 To 200 Step 200
For t = -500 To 500
x = t + 3 * a * h * t ^ 2 / Sqr(1 + (3 * a * t ^ 2) ^ 2)
y = a * t ^ 3 - h / Sqr(1 + (3 * a * t ^ 2) ^ 2)
PSet (x + xc, y + yc): Next t, h
DrawWidth = 2
For t = -500 To 500
x = t: y = a * t ^ 3: PSet (x + xc, y + yc): Next t
End Sub

```

1.24. Для семейства окружностей $(x - \alpha)^2 + y^2 = R^2$ огибающей линией являются две прямые $y = R$ и $y = -R$. Для построения семейства окружностей можно использовать уравнения $x = R \cos t + \alpha, y = R \sin t$.

1.25. Семейство окружностей $(x - R \cos \alpha)^2 + (y - R \sin \alpha)^2 = R_1^2$ имеет огибающей линией две окружности $x^2 + y^2 = (R + R_1)^2$ и $x^2 + y^2 = (R - R_1)^2$

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): xc = 4000: yc = 3000: r = 1200: r1 = 400
For i = 0 To 2 * pi Step pi / 18
For t = 0 To 2 * pi Step 0.001
PSet (r * Cos(i) + r1 * Cos(t) + xc, r * Sin(i) + r1 * Sin(t) + yc)
Next t, i
DrawWidth = 2
For t = 0 To 2 * pi Step 0.001
PSet (r * Cos(t) + xc, r * Sin(t) + yc)
PSet ((r + r1) * Cos(t) + xc, (r + r1) * Sin(t) + yc)
PSet ((r - r1) * Cos(t) + xc, (r - r1) * Sin(t) + yc)
Next t: End Sub
```

```
1.26. Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): xc = 4000: yc = 3000: R = 1000
For al = 0 To 2 * pi Step pi / 25
xa = R * Cos(al) - R * Sin(al): ya = R * Sin(al) + R * Cos(al)
xb = R * Cos(al) + R * Sin(al): yb = R * Sin(al) - R * Cos(al)
Line (xa + xc, ya + yc)-(xb + xc, yb + yc)
Next al
End Sub
```

1.27. Пусть оси координат направлены вдоль сторон данного прямого угла. Рассмотрим отрезки с концевыми точками $A(\alpha; 0), B(0; 2S/\alpha)$, где $\alpha \neq 0$. Уравнение прямой AB : $y = \frac{2S}{\alpha} - \frac{2S}{\alpha^2}x$. Дифференцируя по α , получаем

$0 = -\frac{2S}{\alpha^2} + \frac{4Sx}{\alpha^3}$. Следовательно, $\alpha = 2x$. Уравнение огибающей $2xy = S$. Оги-

бающей линией является ветвь гиперболы, расположенная в первой четверти.

```
Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
xc = 4000: yc = 3000
For i = 1 To 4 Step 0.4
x = 200 * 2 ^ i: y = 1000000 / x: Line (x + xc, yc)-(xc, y + yc)
Next i
DrawWidth = 2
Line (xc, yc)-(xc, 4000 + yc): Line (xc, yc)-(xc + 4000, yc)
End Sub
```

1.29. Программа «Домики»
Private Sub Домик(x1, y1, a, b, hs, hk)


```

Line (x1, y1)-(x1 + a, y1 + hs), wblack, B
Line (x1 + a, y1)-(x1 + a + b, y1 + hs), wblack, B
Line (x1, y1 + hs)-(x1 + a + b, y1 + hs) '
Line (x1 + a / 2, y1 + hs + hk)-(x1, y1 + hs) 'EG
Line (x1 + a / 2, y1 + hs + hk)-(x1 + a, y1 + hs)
Line (x1 + a / 2, y1 + hs + hk)-(x1 + b + a / 2, y1 + hs + hk)
Line (x1 + b + a / 2, y1 + hs + hk)-(x1 + a + b, y1 + hs)
End Sub
Private Sub Command1_Click()
Scale (-3, 16)-(30, -2)
Домик 0, 0, 1, 2, 1, 1: Домик 4, 0, 3, 3, 2, 2: Домик 11, 0, 4, 3, 2, 2
Домик 0, 3, 1, 1, 3, 2: Домик 4, 5, 3, 8, 2, 1: End Sub

```

```

1.30. Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): x0 = 5000: y0 = 3500: r = 200
Circle (x0, y0), r
For k = 0 To 7
t0 = k * pi / 4
For t = t0 To 4 * pi Step 0.001
xn = r * Cos(t) + r * (t - t0) * Sin(t): yn = r * Sin(t) - r * (t - t0) * Cos(t)
PSet (xn + x0, yn + y0)
Next t, k
End Sub

```

```

1.31. Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): x0 = 5000: y0 = 3500: r = 200
Circle (x0, y0), r
For k = 0 To 12
t0 = k * pi / 6
For t = t0 To 4 * pi + t0 Step 0.001
xn = r * Cos(t) + r * (t - t0) * Sin(t): yn = r * Sin(t) - r * (t - t0) * Cos(t)
PSet (xn + x0, yn + y0): Next t, k
For k = 0 To 12
t0 = k * pi / 6
For t = t0 To -4 * pi - t0 Step -0.001
xn = r * Cos(t) + r * (t + t0) * Sin(t): yn = r * Sin(t) - r * (t + t0) * Cos(t)
PSet (xn + x0, yn + y0)
Next t, k: End Sub

```

1.32. Программа «Вращающееся колесо».

```

Private Sub Command1_Click()
Form1.Scale (0, Form1.Height)-(Form1.Width, 0)
pi = 4 * Atn(1): n = 3: xc = 4000: yc = 4000
r = 2000: al = pi / (12 * n): bet = al
For k = 0 To 200
For t = 0 To 4 * pi Step 2 * al
xl = r * Cos(t + bet): yl = r * Sin(t + bet)

```

```

    Line (xc, yc)-(xl + xc, yl + yc)
  Next t
For s = 0 To 1000000
  Next s
For u = 0 To 2 * pi Step 4 * al
  xl = r * Cos(u + bet): yl = r * Sin(u + bet)
  Line (xc, yc)-(xl + xc, yl + yc), QBColor(15)
Next u: bet = bet + al: Next k
End Sub

```

1.38. Программа «Полосы и окружности».

1.40. Программа «Закраска между прямыми в прямоугольнике».

Литература

1. *Александров А.Д., Вернер А.Л., Рыжик В.И.* Геометрия 10, 11. М.: Просвещение, 1992.
2. *Бескин Л.Н.* Деление отрезка в данном отношении. М.: Наука, 1973.
3. *Болтянский В.Г.* Огибающая М.: ГИФМЛ, 1961.
4. *Босова Л.Л.* Информатика: учебник для 7 класса. М.: БИНОМ. Лаборатория знаний, 2008.
5. *Вернер А.Л., Кантор Б.Е., Франгулов С.А.* Геометрия. Ч. 1: Учеб. пособие. СПб.: Специальная литература, 1997.
6. *Волчёнков Н.Г.* Программирование на Visual Basic 6: В 3 ч. М.: ИНФРА, 2000. Ч. 1.
7. *Гальперин Г.А., Степин А.М.* О треугольном бильярде // Квант. 1989. № 11. С. 30—32.
8. *Гальперин Г.А., Земляков А.Н.* Математические бильярды М.: Наука, 1990.
9. *Гарднер М.* Путешествие во времени. М.: Мир, 1990.
10. *Деца Е.И.* О содержании элективного курса «Фигурные числа» // Математика в школе. 2007. № 4 С. 55—59; № 5. С. 31—43.
11. *Коксетер Г.С.М., Грейцер С.Л.* Новые встречи с геометрией М.: Наука, 1974.
12. Математический цветник / Сост. и ред. Д.А.Кларнер; Пер с англ. М.: Мир, 1983.
13. *Меженный О.А.* Turbo Pascal: учитесь программировать. М.: Издательский дом «Вильямс», 2009.
14. *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение. М.: Мир, 1989.
15. *Савелов А.А.* Плоские кривые. М.: ГИФМЛ, 1969.
16. *Сворцов А.В.* Триангуляция Делоне и ее применение. Томск: Изд-во Том. ун-та, 2002.
17. *Смирнова И.М., Смирнов В.А.* Геометрия. Учебник для 7—9 кл., М.: Просвещение, 2001.
18. *Смогоржевский А.С., Столова Е.С.* Справочник по теории плоских кривых третьего порядка. М.: ГИФМЛ, 1961.
19. *Совертков П.И.* Занимательное компьютерное моделирование в элементарной математике: Учеб. пособие. М.: Гелиос АРВ, 2004.
20. *Совертков П.И., Назин А.Г.* Моделирование в интегративном проекте по математике и информатике. Элективный курс: Практикум. М.: БИНОМ. Лаборатория знаний. 2012.
21. *Совертков П.И., Назин А.Г.* Моделирование в интегративном проекте по математике и информатике. Элективный курс: Учеб. пособие. М.: БИНОМ. Лаборатория знаний. 2012.
22. *Угринович Н.Д.* Исследование информационных моделей. Элективный курс: Учеб. пособие. М.: БИНОМ. Лаборатория знаний, 2006.
23. *Яглом И.М.* Геометрические преобразования. М.: ГИТТЛ. 1956. Ч. II.

Учебное издание

Совертков Петр Игнатьевич

**ИССЛЕДОВАТЕЛЬСКИЕ
ПРОЕКТЫ
ПО МАТЕМАТИКЕ
И ИНФОРМАТИКЕ**

Практикум

Редактор *Т.А.Фридман*

Компьютерная верстка *А.З.Насибуллиной*

Изд. лиц. ЛР № 020742. Подписано в печать 10.09.2013. Формат 60×84/16
Бумага для множительных аппаратов. Гарнитура Times
Усл. печ. листов 20,75. Тираж 300 экз. Заказ 1407

Отпечатано в Издательстве

*Нижевартовского государственного университета
628615, Тюменская область, г.Нижевартовск, ул.Дзержинского, 11
Тел./факс: (3466) 43-75-73, E-mail: izdatelstvo@nggu.ru*