УДК 004.925.8
DOI: 10.25686/978-5-8158-2474-4-2025-934-941

# Моделирование структуры «звезда Паскаля» с применением генеративного метода

**С. Ю. Щур**
Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

***Аннотация.*** В работе рассмотрены вопросы применения декларативного программирования в создании геометрических структур, генеративный характер построения структуры «звезда Паскаля». Представлен алгоритм создания структуры с использованием инструментов логического программирования – предикатов. Результаты исследования могут лечь в основу создания новых алгоритмов и поиска решения фундаментальных задач геометрического моделирования. Описываемый метод является более наглядным и оптимизированным по сравнению с созданием структуры «вручную» Результаты работы могут лечь в основу решения задач в области многомерного проектирования с анализом сложных геометрических образов, а также поиском соответствия между имеющимися алгоритмами. Определена возможность генерации геометрических образов с помощью логических функций в программе «Симплекс».

***Ключевые слова:*** геометрическое моделирование, генеративное проектирование, язык логического программирования, пролог, предикат

# Modeling of the Pascal's Star structure using the generative method

**S. Yu. Shchur**
Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russia

***Abstract.*** The paper discusses the application of declarative programming in the creation of geometric structures. The generative nature of the construction of the Pascal's star structure is considered. An algorithm for creating a structure using predicate logic programming tools is presented. The results of the research can form the basis for the creation of new algorithms and the search for solutions to fundamental problems of geometric modeling. The described method is more visual and optimized than creating a structure "manually". The results of the work can form the basis for solving problems in the field of multidimensional design with the analysis of complex geometric patterns, as well as the search for a match between existing algorithms. The possibility of generating geometric images using logical functions in the Simplex program is determined.

***Keywords*:** geometric modeling, generative design, logical programming language, prolog, predicate.

## Введение

В данной работе рассматриваются методы генеративного проектирования в создании структуры «звезда Паскаля». Оценка эффективности генеративной политики представлена в работе [1]. Идея применения методов генеративного проектирования изложена ранее [2] и заключается в широких возможностях генерации вариантов проектирования на основе заданных параметров и ограничений.

Для визуального анализа геометрических структур графоаналитические алгоритмы стали наиболее развитыми и удобными за последнее время [3].

Объект исследования – генеративное проектирование.

Предмет исследования – моделирование геометрических структур на основе методов генеративного проектирования.

Связь декларативного программирования и геометрического моделирования отображается в серии работ [4, 5], в частности описывается использование языка PROLOG для доказательства теорем евклидовой геометрии. В качестве основной программы была выбрана система «Симплекс» [6].

## Постановка задачи

*Целью работы* является описание алгоритма генеративного проектирования геометрической структуры «звезда Паскаля» при помощи инструмента декларативного программирования. Актуальность исследования основывается на недостатке программных средств автоматизировать процесс построения геометрических структур при решении задач подобного рода.

Одной из задач данного исследования является демонстрация возможностей программного обеспечения «Симплекс», в частности программного модуля «Логика». Методы генеративного проектирования обсуждались ранее. Возможности модуля «Логика», основанного на принципах декларативного программирования, также были обозначены ранее [2].

Возможности программного комплекса автоматизированного проектирования «Симплекс» в решении задач генеративного проектирования определены. Это позволяет рассуждать о поиске решения геометрических задач многомерного пространства. Работа базируется на использовании языка декларативного программирования «Пролог» [7].

**Теория**

В качестве основы предлагается рассмотреть структуру «звезда Паскаля». Построение фигуры происходит следующим образом: в окружность или коническое сечение вписывается шестиугольник или определяются 6 точек A, B, C, D, E, F, принадлежащих данной конике f. Точки соединяются последовательно: AB, BC, CD, DE, EF, FA. В результате пересечения противоположных сторон получаем точки M, P, N (рис. 1).

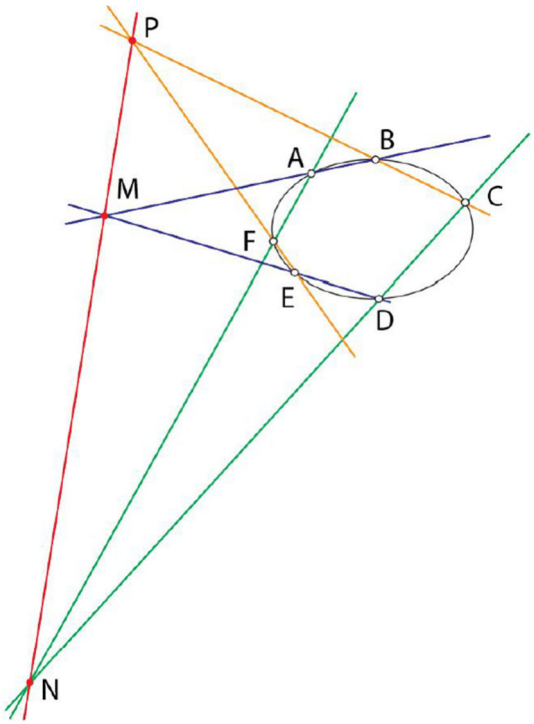


Рисунок 1. Звезда Паскаля

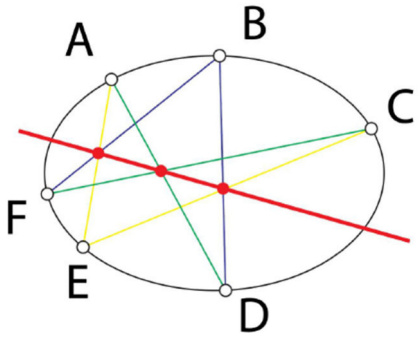


Рисунок 2. Структура «звезда Паскаля» с самопересекающимся контуром

Точки M, P и N по теореме Паскаля обязательно будут лежать на одной прямой - прямой Паскаля. Теорема остается верной даже при произвольном расположении точек на конике и самопересечении контура (рис. 2).

С точки зрения практического значения данные свойства структуры используются при решении задач проективной геометрии, в теории конических сечений, построении геометрических фигур и т.д. Свойства звезды Паскаля используются в улучшении точности классификации изображений, а также в процессе распознавания рукописного текста, а также обработки изображений с геометрическими фигурами.

Создание геометрических структур на основе декларативного программирования позволяет вести дальнейшую работу в рамках обнаружения объектов по геометрическим свойствам, классифицировать фигуры по их проективным характеристикам, а также проводить анализ симметрии и коллинеарности в изображениях.

В перспективе возможны разработка специализированных свёрточных нейронных сетей для работы с проективной геометрией, а также создание новых архитектур с учетом свойств конических сечений.

**Результаты экспериментов**

В системе «Симплекс» была создана коника с расположенными на ней точками a, b, c, d, e, f (рис. 3).

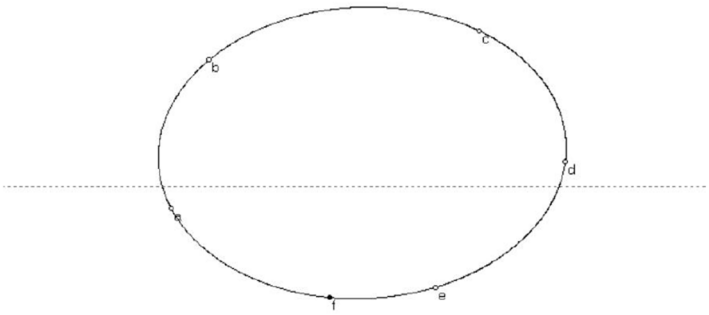


Рисунок 3. Геометрическая модель в Симплексе

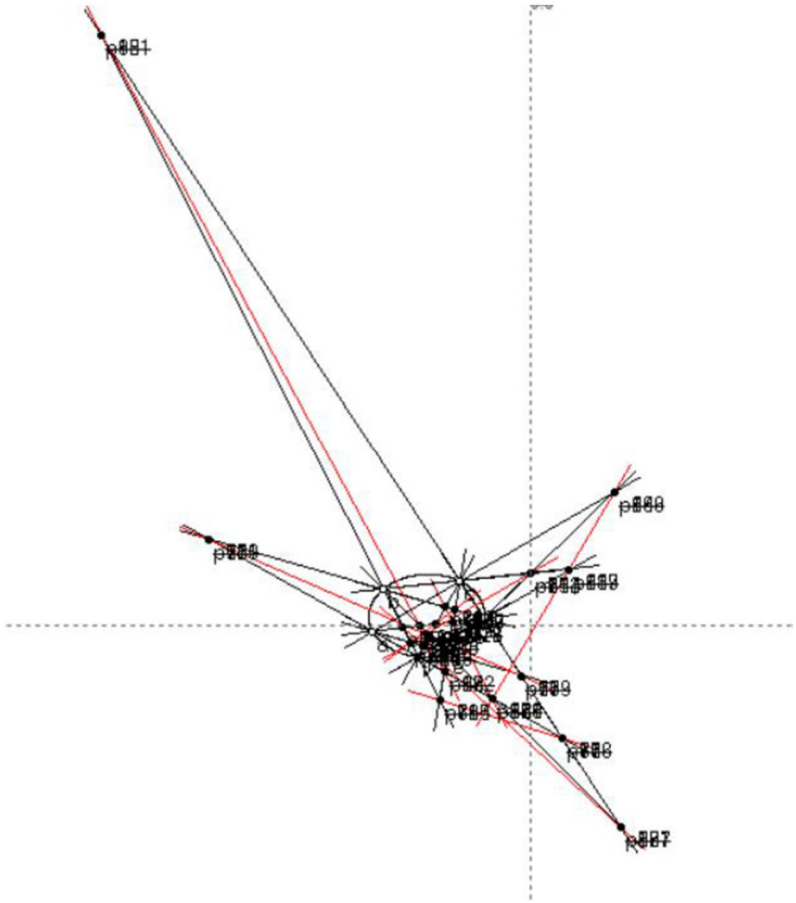

Рисунок 4. Общий вид результата генерации звезды в Симплексе

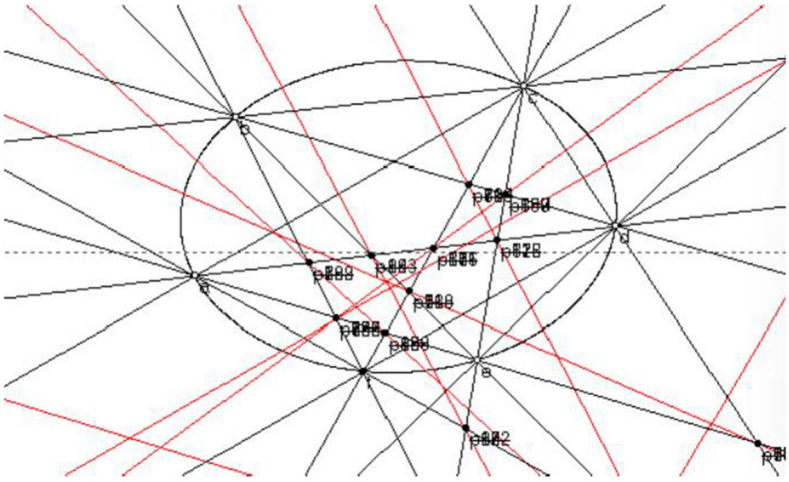

Рисунок 5. Результат генерации звезды в Симплексе

На рисунках 4, 5 представлена структура, полученная в результате генерации в Симлексе. Для изменения геометрического образа программа остается прежней. Программно задается изменение геометрии, без непосредственного влияния и достраивания вручную.

Целью программы является определение group(A,B,C,D,E,F),pascal(A,B,C,D,E,F),execute. Последний в данном случае является оператором исполнения алгоритма.

Программа составлена следующим образом:

```
group(A,B,C,D,E,F) if
member(A,[a,b,c,d,e,f]),
member(B,[a,b,c,d,e,f]),
member(C,[a,b,c,d,e,f]),
member(D,[a,b,c,d,e,f]),
member(E,[a,b,c,d,e,f]),
member(F,[a,b,c,d,e,f]),
ne(A,B),
ne(A,C),
ne(A,D),
ne(A,E),
ne(A,F),
ne(B,C),
ne(B,D),
ne(B,E),
ne(B,F),
ne(C,D),
ne(C,E),
ne(C,F),
ne(D,E),
ne(D,F),
ne(E,F).
pascal(A,B,C,D,E,F) if
lt(A,E),
lt(A,D),
lt(B,F),
lt(B,D),
lt(C,F),
lt(C,E),
nnf(O1,'o','Главный'),
```

```
nnf(O2,'o','Главный'),
nnf(O3,'o','Главный'),
nnf(O4,'o','Главный'),
nnf(O5,'o','Главный'),
nnf(O6,'o','Главный'),
nnf(O7,'o','Главный'),
nnf(P1,'p','Главный'),
nnf(P2,'p','Главный'),
nnf(P3,'p','Главный'),
addline([[E], [A]], [O1], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[D], [A]], [O2], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[F], [B]], [O3], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[D], [B]], [O4], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[F], [C]], [O5], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[E], [C]], [O6], '0', 'Прямая задана двумя точками', 'Главный'),
addline([[O1], [O3]], [P1], '0', 'Точка пересечения двух прямых', 'Главный'),
addline([[O2], [O5]], [P2], '0', 'Точка пересечения двух прямых', 'Главный'),
addline([[O4], [O6]], [P3], '0', 'Точка пересечения двух прямых', 'Главный'),
addline([[P1], [P2], [P3]], [O7], '0', 'Прямая задана триадой точек', 'Главный'),
attribute(O1,'Главный',[@alv(5)]),
attribute(O2,'Главный',[@alv(5)]),
attribute(O3,'Главный',[@alv(5)]),
attribute(O4,'Главный',[@alv(5)]),
attribute(O5,'Главный',[@alv(5)]),
attribute(O6,'Главный',[@alv(5)]),
attribute(O7,'Главный',[@alv(5)]),
attribute(O7,'Главный',[@acr(255)]).
```

В данной программе изначально задается группа group(A,B,C,D,E,F), если выполняется ряд условий: member(A,[a,b,c,d,e,f])...member(F,[a,b,c,d,e,f]). Являются предикатами проверки того, что A, B, C, D, E, F входят в состав списка, представленного в квадратных скобках.

Предикаты ne(A,B)...ne(E,F) являются предикатами проверки истинности утверждения, что аргументы, представленные в скобках, не унифицируемы.

pascal(A,B,C,D,E,F) исполняется, если:

Выполняются предикаты lt(A,E),...lt(C,E), которые служат для того, чтобы проверить, верно ли неравенство A<E...C<E.

Выполняются предикаты nnf(O1,'o','Главный'),...nnf(P3,'p','Главный'), которые определяют имя новой переменной по префиксу ‘o’ в алгоритме ‘Главный’

Выполняются предикаты addline([[E], [A]], [O1], '0', 'Прямая задана двумя точками', 'Главный')...addline([[E], [C]], [O6], '0', 'Прямая задана двумя точками', 'Главный') - добавляет отношения в алгоритм Симплекса, где [[E], [A]],...[[E], [C]] являются входными параметрами отношения;

[O1],...[O6] являются выходными параметрами отношения;

'0' - тип согласования;

'Точка задана двумя точками' - функция;

'Главный' - алгоритм.

Предикаты addline([[O1], [O3]], [P1], '0', 'Точка пересечения двух прямых', 'Главный'),...addline([[O4], [O6]], [P3], '0', 'Точка пересечения двух прямых', 'Главный') добавляют отношения в алгоритм в виде точки пересечения прямых, заданных ранее.

Последний схожий предикат addline([[P1], [P2], [P3]], [O7], '0', 'Прямая задана триадой точек', 'Главный') отвечает за отношение в виде прямой, определяемой найденными ранее тремя точками.

attribute(O1,'Главный',[@alv(5)]),...attribute(O7,'Главный',[@acr(255)]) назначает атрибуты объектам списочной переменной Симплекса.

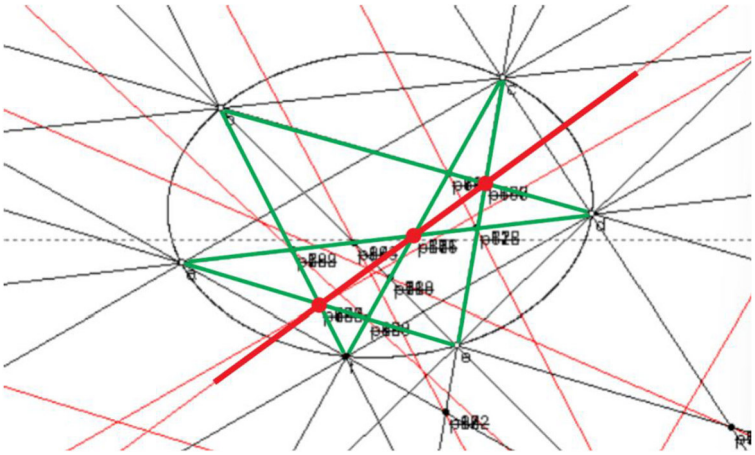

Рисунок 6. Геометрическая модель в Симплексе

В результате с помощью модуля «Логика» в Симплексе методом генеративного перебора вариантов соединения точек удалось получить различные варианты структуры «звезда Паскаля».

В таблице 1 представлены результат сгенерированных вариантов возможных структур, отвечающих запросу, реализованному в программе.

Таблица 1. **Результат генерации структур**

| # | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | a | b | c | d | e | f |
| 2 | a | b | c | d | f | e |
| 3 | a | b | c | e | d | f |
| 4 | a | b | c | e | f | d |
| 5 | a | b | c | f | d | e |
| 6 | a | b | c | f | e | d |
| 7 | a | b | d | c | e | f |
| 8 | a | b | d | c | f | e |
| 9 | a | c | b | d | e | f |
| 10 | a | c | b | d | f | e |
| 11 | a | c | b | e | d | f |
| 12 | a | c | b | e | f | d |
| 13 | a | c | b | f | d | e |
| 14 | a | c | b | f | e | d |
| 15 | a | d | b | e | c | f |
| 16 | a | d | b | f | c | e |
| 17 | b | a | c | d | e | f |
| 18 | b | a | c | d | f | e |
| 19 | b | a | c | e | d | f |
| 20 | b | a | c | e | f | d |
| 21 | b | a | c | f | d | e |
| 22 | b | a | c | f | e | d |
| 23 | b | a | d | c | e | f |
| 24 | b | a | d | c | f | e |
| 25 | b | c | a | d | e | f |
| 26 | b | c | a | d | f | e |
| 27 | b | c | a | e | d | f |
| 28 | b | c | a | e | f | d |
| 29 | b | c | a | f | d | e |
| 30 | b | c | a | f | e | d |
| 31 | b | d | a | e | c | f |
| 32 | b | d | a | f | c | e |
| 33 | c | a | b | d | e | f |
| 34 | c | a | b | d | f | e |
| 35 | c | a | b | e | d | f |

Окончание таблицы 1

| # | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 36 | c | a | b | e | f | d |
| 37 | c | a | b | f | d | e |
| 38 | c | a | b | f | e | d |
| 39 | c | b | a | d | e | f |
| 40 | c | b | a | d | f | e |
| 41 | c | b | a | e | d | f |
| 42 | c | b | a | e | f | d |
| 43 | c | b | a | f | d | e |
| 44 | c | b | a | f | e | d |
| 45 | d | a | b | e | f | c |
| 46 | d | a | b | f | e | c |
| 47 | d | b | a | e | f | c |
| 48 | d | b | a | f | e | c |

**Обсуждение**

Обоснование использования методов генеративного проектирования в вопросах геометрического моделирования лежит прежде всего в плоскости автоматизации проектирования. Однако этим оно не ограничивается. В работе представлены новые подходы в решении геометрических задач. Методы генеративного проектирования интересны прежде всего тем, что они могут подсказать новые пути решения уже устоявшихся фундаментальных задач.

Декларативный характер дает преимущества прежде всего в упрощении пользовательского взаимодействия с геометрической машиной. Работа является ярким примером того, как можно наглядно создать чертеж и динамически его редактировать, внося в него изменения без непосредственного внесения изменений в геометрию структур.

Помимо этого декларативное программирование позволяет создать на базе отлаженной системы некую экспертную систему, которая будет оценивать результаты по критериям. Это позволит сопоставлять их, находить взаимосвязи и отличия в структурах, перебирать множество вариантов для решения определенной задачи.

**Выводы**

В результате полученной структуры удалось определить, что решение геометрических задач и построение модели проходит быстрее с использованием методов генеративного проектирования. Язык «Пролог» и основанный на нем модуль «Логика» в Симплексе являются мощной основой для использования в решении подобных задач. Сама структура логического программирования базируется на переборе возможных вариантов и определении правильных, подходящих к искомой форме.

Область геометрического моделирования не исчерпывается решением одной задачи. Данный метод может быть использован в дальнейшем в решении задач более сложных, например, в области многомерного проектирования. Он также позволяет анализировать сложные геометрические образы, находить соответствия между имеющимися алгоритмами. В работе определена возможность генерации геометрических образов с помощью логических функций в программе «Симплекс».

**Благодарность**

Автор статьи благодарит научного руководителя Волошинова Дениса Вячеславовича, СПбГУТ им. Бонч-Бруевича, СПбПУ Петра Великого за помощь в подготовке работы.

**Список литературы**

1. Алексеев В. Ф., Лихачевский Д. В., Пискун Г. А. Оценка эффективности генеративной политики при оптимизации проектных решений. 2025.

2. Щур С. Ю. Динамическое редактирование геометрических образов с использованием логических инструментов в программной среде Симплекс // GraphiCon 2024 : материалы 34-й Международной конференции по компьютерной графике и машинному зрению, Омск, 17–19 сентября 2024 года. Омск: Омский

государственный технический университет, 2024. С. 837-842. DOI: 10.25206/978-5-8149-3873-2- 2024-837-842. EDN FXJSXK.

3. Волошинов Д. В. Теория автоматизации проектирования объектов и процессов на основе методов конструктивного геометрического моделирования. СПб., 2010.

4. Coelho H., Pereira L. M. Automated reasoning in geometry theorem proving with Prolog // Journal of Automated Reasoning. 1986. Vol. 2. P. 329-390.

5. Extracting isovolumes from three-dimensional torso geometry using PROLOG / Replogle J. A., et al. // IEEE transactions on information technology in biomedicine. 1998. Vol. 2, no. 1. P. 10-19.

6. Волошинов Д. В., Соловьева А. В. Разработка алгоритмического комплекса для решения задач конструктивной геометрии // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020) : сборник научных статей IX Международной научно-технической и научно-методической конференции: в 4 т., Санкт-Петербург, 26–27 февраля 2020 г. Том 2. Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2020. С. 182-186. EDN: VRYFMK.

7. Bratko I. Prolog programming for artificial intelligence. Pearson education, 2001.