

Алгоритм расчета параметризованных геометрических объектов

Копорушкин П.А. (k_p_a@rambler.ru), Партин А.С.

Уральский государственный технический университет-УПИ

1. Введение

Первые системы геометрического моделирования являлись лишь специализированными графическими редакторами, их возможности были ограничены производительностью компьютеров. Такие системы автоматизировали создание чертежа с помощью плоских геометрических примитивов – отрезков прямых, дуг окружностей, текстовых надписей и т.д. Одним из важных изменений при развитии систем геометрического моделирования стало то, что первичной стала модель объекта, по которой уже в дальнейшем строился чертеж. Конструктор стал работать с трехмерной твердотельной моделью объекта и в любой момент мог ее видеть и модифицировать [1].

Одним из методов проектирования, позволивших совершить качественный прорыв в развитии САД-систем стала так называемая **параметризация** [2,3].

Современные параметрические системы принципиально изменили представления конструктора о проектируемом объекте и позволили создавать сложные сборочные 3D модели (сборки [4,5]), состоящие из тысяч деталей (сборочных единиц). Сборки позволяют максимально точно моделировать реальные механизмы, состоящие из большого количества деталей, посмотреть на механизм в действии и др.

Одним из первых и наиболее простых и естественных подходов к решению проблемы удобного и эффективного конструирования сложных объектов явился метод, опирающийся на то, что сложные механизмы строятся, как правило, из однотипных деталей и узлов. Первоначально, в большинстве САД-систем этот подход был реализован с помощью некоторого формального языка, который позволял описывать последовательность действий, необходимых для создания чертежей и моделей типовых деталей по их заданным **параметрам** [3] (например, тип, диаметр и число зубьев шестерни). Из этих деталей в дальнейшем можно составлять стандартные узлы (с помощью того же языка). При конструировании сложного механизма можно использовать созданные ранее стандартные детали и узлы, вводя их параметры и вставляя полученный объект в нужное место чертежа или модели.

Фактически, это один из методов параметризации геометрических объектов. Недостатком описанного подхода является то, что при использовании языка для описания

геометрических объектов конструктору необходимо иметь навыки программирования. К тому же, процесс задания объекта при этом не является наглядным. Другим недостатком является невозможность легкой модификации созданного ранее геометрического объекта, так как после его формирования он хранится в виде набора простых геометрических примитивов. В результате теряется любая информация, как о самом объекте, так и о его связях с другими объектами. Кроме того, любая сборочная единица может постоянно меняться в результате исправления ошибок, изменения условий рабочей среды, производства и многих других факторов. Все эти недостатки были вызваны ограниченными возможностями компьютеров на разных этапах развития параметризации. Однако подобные недостатки встречаются и в современных САД-системах, и это, скорее, проблемы плохой разработки соответствующих систем.

Для решения описанных выше проблем необходимо хранить не только геометрическую информацию, но и связи между элементами детали, отдельными деталями. По своей сути такой метод позволяет в любой момент синтезировать отдельную деталь механизма на основе ее параметров, описать взаимодействие этой детали с остальными деталями механизма, определить работу всего механизма в целом. Чаще всего, в качестве параметров выступают размеры модели объекта.

Встречаются и "компромиссные" варианты реализации параметрических возможностей. Например, в системе SprutCAD параметрическая информация хранится в виде программы, которая создается в процессе моделирования и зависит от последовательности действий [6]. На наш взгляд, такой подход не является достаточно гибким, и не отвечает всем требованиям современных пользователей.

Термин "параметризация" получил довольно большое распространение. Он может означать и процесс создания модели и результат проектирования, который характеризует параметрические свойства модели. В широком смысле под параметризацией понимается процесс проектирования, результатом которого является модель с небольшим набором простых и понятных параметров. Меняя их, конструктор получает нужную ему модель объекта. Например, параметризованная модель шестерни полностью задает число зубьев и несколько размеров. В процессе проектирования конструктор постепенно абстрагируется от исходных примитивов и получает модель, максимально соответствующую его представлениям. Можно говорить и о качестве параметризации [7]. Хорошо параметризованная модель обладает минимальным набором связей и "работает" в широком диапазоне варьируемых параметров.

Процесс получения хорошо параметризованной модели является нетривиальной творческой работой, однако позволяет создавать сложные сборки, моделирующие реальные объекты.

В данной статье мы не будем обсуждать качество параметризации и подойдем к параметризации формально. Под **параметризацией** или **параметрическим моделированием** будем далее подразумевать процесс создания **параметрической модели объекта**, состоящей из набора геометрических примитивов и параметрических связей. Говорят, что геометрическая модель вторична по отношению к параметрической модели, т.к. может быть по ней однозначно синтезирована.

Термин "параметр" также имеет двойственную природу и употребляется в двух смыслах:

1. Параметр объекта. Модель любого геометрического объекта имеет параметры, например, параметрами прямоугольника являются его размеры. В дальнейшем параметры объекта будем называть **макропараметрами**.
2. Любой геометрический элемент чертежа имеет свои параметры. Например, прямоугольник состоит из четырех отрезков, каждый отрезок имеет 4 параметра – координаты точек.

Несмотря на то, что конструктор выделяет только макропараметры, любая модель должна задаваться множеством обычных параметров – координат. Таким образом, можно сказать, что параметризация – это процесс связывания параметров и макропараметров модели проектируемого объекта.

Параметрическое конструирование обладает рядом практических преимуществ: интуитивный процесс создания, возможность легкой модификации и повторного использования объекта. Параметризация позволяет существенно сократить сроки проектирования объектов, особенно сложных, так как большую часть времени занимает не само проектирование, а корректировка объекта. Форму параметризованного объекта можно легко модифицировать простым изменением параметров и соотношений. Параметрическая модель создается интерактивно, без программирования.

Параметризация также удобна для создания библиотек типовых деталей и повторного использования созданных ранее объектов.

Задача параметризации моделей геометрических объектов была сформулирована достаточно давно [8,9,10]. Предложено большое количество разнообразных методов решения этой задачи, отличающихся по сложности реализации, аппаратным требованиям и объему хранимой информации. К сожалению, отсутствует подробный теоретический анализ методов параметризации, применяемых в популярных САД-системах. По-видимому, это связано с

коммерческих характером таких систем. Был проведен сравнительный тест функциональных возможностей следующих систем: AutoCAD 2002, Compas 5.10, Cimatron 11, Inventor 5, SolidWorks 2001 Plus, Solid Edge 11 [11]. Большинство САД-систем не обеспечивают полной параметризации проектируемого объекта. Из всех рассматриваемых систем только SolidWorks работает в режиме полностью параметризованной модели. По глубине и гибкости средств параметризации модели лидируют SolidWorks и Solid Edge.

Закрытость теоретических основ параметрического конструирования привело к тому, что разработчики вынуждены либо лицензировать ядро параметризации сторонних разработчиков, либо разрабатывать такое ядро самостоятельно [3].

В данной статье мы постарались описать основные идеи и алгоритмы, использованные при разработке собственного модуля параметрического конструирования. Наш подход к параметризации позволяет наделить параметрическими возможностями широкий класс САПР.

2. Основные термины и определения

Эскизироващик – это модуль, позволяющий наглядно вводить и отображать параметрические связи. В нашем понимании связь – это некоторое ограничение, накладываемое на проектируемую модель объекта, которое может быть описано математическими уравнениями. На любом этапе проектирования должны соблюдаться все ограничения (связи), наложенные на модель.

Параметрические связи можно разбить на следующие категории:

- топологические;
- геометрические (параллельность, перпендикулярность и т.д.);
- математические (соотношения между параметрами различных элементов контура или параметрами различных объектов, задаваемые произвольными математическими уравнениями);
- размеры (линейные, радиальные, угловые и т.д.).

Размеры попали в список, т.к. часто также являются связями. Такие размеры мы будем называть размерными связями. Размеры, которые не накладывают никаких ограничений на модель, назовем иллюстративными.

Аналогично связи выделяется понятие привязки. **Привязка** отражает отношение между объектами, на которые наложена связь. Например, привязка одного отрезка к другому по параллельности означает, что оба отрезка наложены связь "параллельность", т.е. отрезки параллельны друг другу.

Каждому геометрическому объекту соответствует фиксированный **набор параметров**. Например, параметрами точки являются ее координаты, окружности – координаты центра и радиус и т.д.

Связи, задающие размер, обладают особыми свойствами, т.к. они порождают макропараметр, который характеризует весь проектируемый объект в целом.

Каждая накладываемая на объекты и их параметры связь может породить одно или несколько связывающих эти параметры математических **уравнений**.

Параметризованный контур (в общем случае – **трехмерная твердотельная модель**) - это совокупность геометрических примитивов и наложенных на них параметрических связей. В любой момент по параметрической модели контура может быть рассчитана геометрическая модель.

Для каждой связи назначается **вес** или **приоритет**, который может изменяться пользователем. Связи с большим весом являются более предпочтительными, так как обычно обеспечивают лучшую схему параметризации. Введение приоритетов – это попытка формализации представлений пользователя о параметрических связях. Например, параллельность обеспечивает более гибкую параметрическую модель, чем вертикальность. Вес используется в двух случаях:

1. При работе эскизироващика, для выбора одной из списка найденных возможных связей.
2. При расчете параметров для анализа конфликтных ситуаций и автоматического добавления связей.

Эскизироващик присутствует в большинстве систем автоматизированного проектирования. Отличительной чертой параметрических систем является наличие встроенного механизма расчета параметров и анализа конфликтных ситуаций – **параметризатора**.

При попытке расчета нового набора параметров геометрии, соответствующей связям, параметризатор проверяет возможность такого расчета. Если связей недостаточно, то говорят, что модель **недоопределена** [12]. Если же связей слишком много, то контур **переопределена**.

Будем различать следующие типы параметров: **свободные**, **полусвободные** и **связанные**. Свободные параметры вообще не участвуют в процессе расчета, т.к. не входят ни в одну связь. В полностью определенном контуре нет свободных параметров. Полусвободные параметры задействованы в связях, но не участвуют в процессе расчета, так как, например, их значение должно остаться неизменным. Как правило, макропараметры являются полусвободными.

Существует два вида параметризации: жесткая и мягкая (с полным и неполным набором связей) [7]. **Жесткая параметризация** требует от пользователя задания всех связей, необходимых для однозначного расчета геометрии объекта. **Мягкая параметризация** дает пользователю возможность задавать только некоторые связи, определяющие необходимые свойства контура. Оставшиеся связи должны быть добавлены автоматически.

3. Задача расчета параметров геометрических объектов

Основную задачу параметризации можно сформулировать следующим образом: дана параметрическая модель объекта, необходимо рассчитать его геометрическую модель или объяснить, почему это невозможно [8]. Желательно, чтобы объем необходимых вычислений был минимален. В данной статье предлагаются алгоритмы для решения этой задачи. Рассмотрим их работу на следующем примере. Предположим, что задан такой контур:

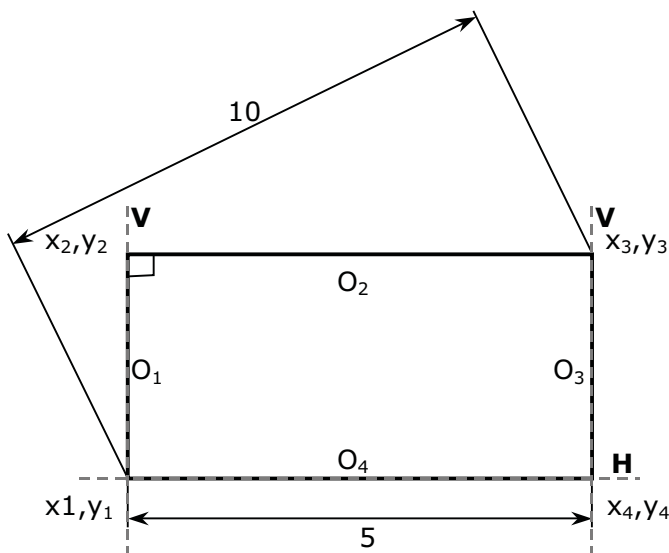


Рис. 1

Отрезки O_1, O_3 привязаны друг к другу по вертикали. На отрезок O_4 наложена связь "горизонтальность". Отрезки O_1 и O_2 привязаны друг к другу по перпендикулярности.

Все связи можно представить в виде уравнений:

1	Вертикальность O_3	$x_3 - x_4 = 0$
2	Вертикальность O_4	$y_1 - y_4 = 0$
3	Вертикальность O_1	$x_1 - x_2 = 0$
4	Горизонтальный размер O_4	$(x_1 - x_4)^2 - 25 = 0$
5	Диагональный размер	$(x_1 - x_3)^2 + (y_1 - y_3)^2 - 100 = 0$
6	Перпендикулярность O_1 и O_2	$(x_1 - x_2)(x_2 - x_3) + (y_1 - y_2)(y_2 - y_3) = 0$

Таблица 1

Всего введено 6 связей, которые порождают 6 уравнений. Представим эту систему уравнений в виде графа. Вершинами в этом графе будут параметры, входящие в уравнения и сами уравнения.

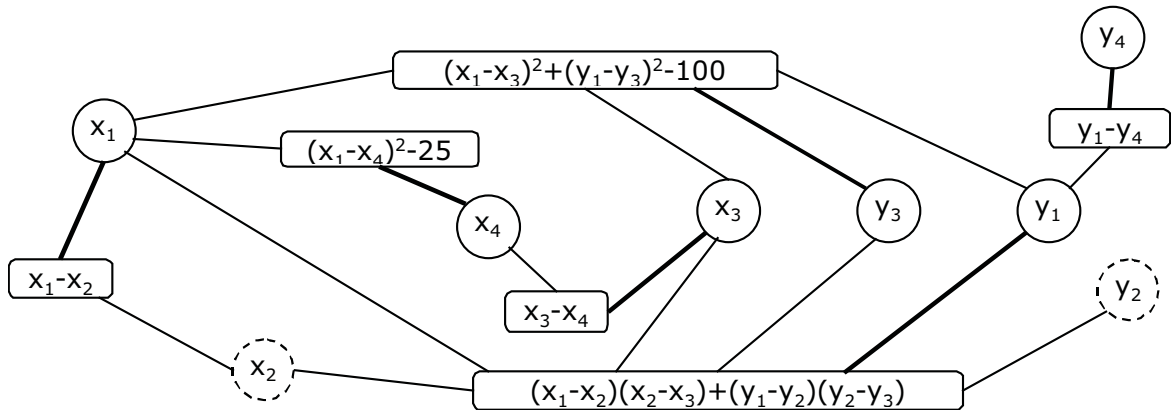


Рис.2

Граф, приведенный на рис. 2 позволяет наглядно представить зависимости между параметрами. Первый этап состоит в том, чтобы решить, какой параметр нужно выразить из каждого уравнения. На рисунке такое соответствие обозначено жирными линиями. Параметры, которые не должны меняться в процессе расчета, обозначены пунктиром. Параметрам x_2, y_2 соответствует левая верхняя точка прямоугольника, ее координаты в процессе расчета не поменяются.

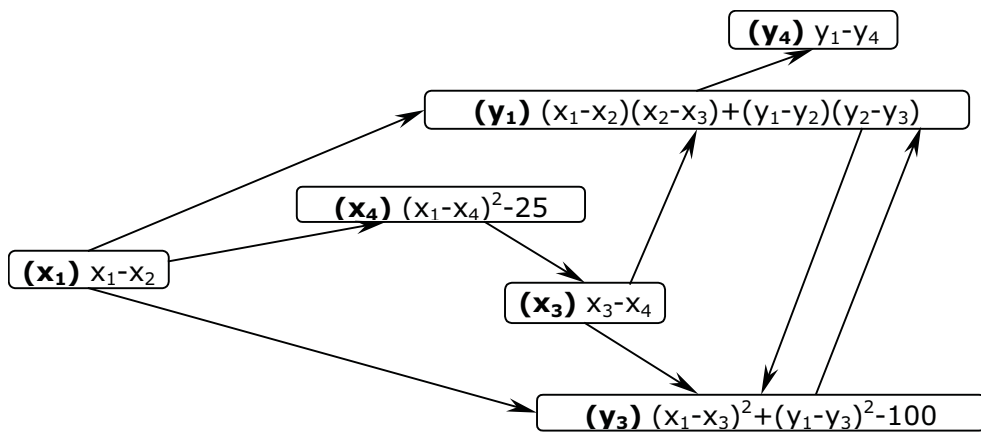


Рис. 3

Граф пересчета, изображенный на рис. 3, получается из предыдущего "стягиванием" пар вершин, соединенных жирной линией. Оставшиеся ребра заменяются ориентированными ребрами, идущими от параметра к связи. Этот граф отображает последовательность, в которой нужно подставлять параметры. Такой метод вычисления систем уравнений называется методом последовательного исключения.

В данном графе возможно появление циклов. Это значит, что подстановки параметров должны выполняться циклически. Следовательно, все уравнения, входящие в цикл, должны решаться совместно в одной системе уравнений.

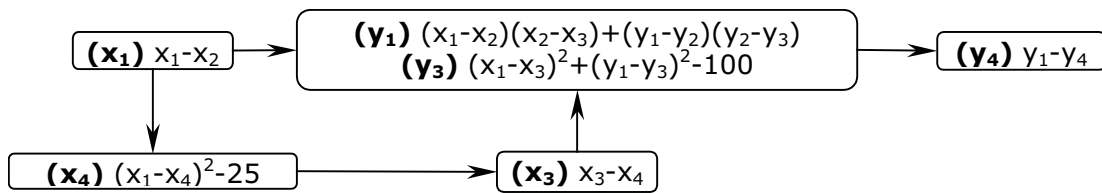


рис. 4

На рис. 4 показан граф пересчета, вершинам которого соответствуют системы уравнений, ребра графа показывают правильную последовательность расчета этих систем. Последовательность расчета считается правильной, если на каждом шаге в решаемой системе число неизвестных параметров равно числу уравнений. Расчет начинается с тех систем, для которых на графе пересчета нет ни одной входящего ребра, и заканчивается после решения всех систем уравнений.

Исходному эскизному контуру соответствовали следующие координаты:

$$x_1=10, y_1=10, x_2=10, y_2=40, x_3=20, y_3=40, x_4=20, y_4=20$$

В результате расчета получаем следующий результат:

$$x_1=10, y_1=31.34, x_2=10, y_2=40, x_3=15, y_3=40, x_4=15, y_4=31.34$$

В данном разделе мы рассмотрели один из реальных примеров. Далее мы обсудим алгоритм расчета параметров более подробно. Для простоты и большей наглядности получаемых графов были использованы упрощенные уравнения для размеров. Значение размеров в них было представлены в виде числовой константы. В общем случае им соответствует макропараметр. Например, в нашем примере диагональному размеру соответствует такое уравнение:

$$(x_1-x_3)^2+(y_1-y_3)^2-100=0$$

В общем случае оно должно быть записано так:

$$(x_1-x_3)^2+(y_1-y_3)^2-L_1^2=0$$

В уравнение входят параметры объекта (координаты связываемых точек) и макропараметр, который также присутствует на графе параметров и обычно является полусвободным.

3.1. Построение графа параметров

Граф параметров (пример на рис.2) может состоять из нескольких компонент связности[13]. Т.к. компоненты не зависят друг от друга, то обрабатываются отдельно. Дальнейшие рассуждения будут приводить для отдельной компоненты связности.

Первый этап расчета заключается в том, чтобы выразить из каждой связи по параметру, т.е. выделить в графе пары "параметр-связь", а также проанализировать конфликтные ситуации.

Каждая вершина может входить только в одну пару. Набор таких пар называется **паросочетанием**.

Нетрудно заметить, что граф является двудольным[13], т.е. содержит два множества взаимно несмежных вершин.

Решением задачи расчета параметров будет паросочетание, в которое входили бы все заданные пользователем связи. В результате, задача расчета параметров на первом этапе сводится к поиску максимального паросочетания[13]. Требование максимальной паросочетанию означает, что должно быть использовано максимальное количество связей.

После построения максимального паросочетания необходимо провести анализ возможности успешного расчета параметров. Если найденное паросочетание включает все пользовательские связи (совершенное паросочетание), то расчет параметров можно продолжить. Однако часть связанных параметров, возможно, не была задействована в процессе расчета. Параметр исключается из процесса расчета, если он является полусвободным. Например, временно свободным объявляется параметр, если его значение было изменено пользователем. Дальнейшие действия зависят от того, какая модель параметризации была выбрана.

В случае жесткой параметризации в контуре допускается два свободных параметра. Только в этом случае контур будет считаться полностью образмеренным. На нашем примере свободной является точка с координатами x_2, y_2 . При выборе модели мягкой параметризации контур считается образмеренным, если в максимальном паросочетании участвуют все пользовательские связи. На наш взгляд мягкая параметризация более предпочтительна, т.к. избавляет от необходимости задавать все связи, делает контур более простым и наглядным.

3.2. Автоматическое добавление связей

Независимо от вида параметризации системе приходится решать задачу автоматического добавления неявных связей. Эти связи добавляются каждый раз при расчете контура. Поведение недообразмеренного контура зависит от набора автоматических связей. Каждая такая связь также имеет приоритет, который может быть изменен пользователем.

Без добавления автоматических связей нормальная работа с недообразмеренным контуром невозможна. В этом случае контур ведет себя абсолютно непредсказуемо. Это неприемлемо, т.к. первоначально контур конструируется эскизно и должен оставаться узнаваемым. Здесь должен действовать принцип локальности модификации. Этот принцип заключается в том, что в процессе расчета должно изменяться как можно меньше параметров контура, а любые изменения геометрии должны быть по возможности локальными.

Автоматические связи могут быть двух видов:

1. привязки геометрических примитивов друг к другу (параллельность, перпендикулярность);
2. связи, фиксирующие положение их на плоскости, ориентацию и размеры.

Все связи (пользовательские и автоматические) можно объединить в один двудольный граф пересчета. Обобщенную проблему расчета параметров можно сформулировать так: необходимо найти максимальное паросочетание с максимальным суммарным весом (приоритетом) связей, входящих в него. В теории графов существует точный алгоритм решения этой задачи со сложностью $O(n^3)$ [13,14,15].

Можно отделить поиск решения для пользовательских связей от процедуры выбора автоматических связей и получить более быстрый алгоритм со сложностью $O(n^{5/2})$, пользуясь следующими соображениями:

1. При поиске максимального паросочетания для пользовательских связей можно не учитывать приоритет, т.к. все связи должны быть использованы. Приоритеты пользовательских связей пригодятся только при анализе конфликтных ситуаций.
2. Выбирать автоматические связи можно аналогичным образом, используя приоритет автоматических связей только для ограничения числа связей, входящих в набор.
3. Для построения максимального паросочетания в случае с автоматическими связями нужно использовать параметры, оставшиеся неиспользованными после обработки пользовательских связей.

3.3. Анализ конфликтных ситуаций

В процессе расчета параметров могут возникать различные конфликтные ситуации. Чаще всего появляются следующие проблемы:

1. Локальное переопределение контура;
2. Недоопределенность контура;
3. Ошибки в процессе решения систем математических уравнений (реализованный численный метод не позволяет решить систему уравнений);
4. Вырожденные решения.

Каждая ситуация отслеживается на соответствующем этапе решения. Например, если в найденном паросочетании не задействованы некоторые пользовательские связи, то контур переопределен. Если контур недообразмерен, то некоторые параметры не участвуют в расчете. Следовательно, контур необходимо дополнить автоматическими связями.

3.4. Построение сети пересчета

Следующим этапом расчета параметров является построение графа пересчета (рис.3). Этот граф получается из графа, изображенного на рис.2 путем "стягивания" пар максимального паросочетания.

Этот граф содержит "правильную" последовательность расчета параметров. Если на графе есть цикл, то это обозначает, что все уравнения, которым соответствуют вершины, входящие в этот цикл, должны решаться с помощью одной системы уравнений. Строго говоря, граф необходимо разбить на компоненты сильной связности (КСС). КСС называется максимальный сильно связанный подграф графа пересчета[15]. В сильно связанном графе произвольная вершина x_j достижима из любой другой вершины x_i , в ориентированном графе существует одна и только одна КСС, содержащая данную вершину x_j . Каждая КСС формирует свою систему уравнений. На рис. 5 показан результат процесса выделения компонент сильной связности.

Получение компонент сильной связности гарантирует минимизацию объема вычислений. При уменьшении числа уравнений в системе порождает циклы на графе пересчета, что противоречит теории.

Стягиванием компонент сильной связности получаем конденсацию графа пересчета или сеть пересчета[16]. Каждая вершина такого графа представляет множество вершин некоторой КСС. Очевидно, что конденсация графа пересчета не содержит циклов и не является ориентированным деревом.

Сеть пересчета на рис. 4 показывает правильную последовательность расчета компонент сильной связности.

На данном этапе могут возникнуть проблемы расчета связанные с невозможностью решения системы уравнений. Например, проблема может возникнуть, если в решаемую систему попала группа линейно-зависимых уравнений. Процедура выявления линейно-зависимых уравнений очень сложна. Авторам не известен ни один надежный алгоритм исключения групп линейно-зависимых уравнений. Однако такой ситуацию просто избежать, если соблюдать ряд правил. Например, параметрическая модель не может содержать двух одинаковых связей.

3.5. Пересчет модифицированного контура и поддержка актуальности контура

Основным достоинством параметризованного контура является возможность легкой модификации контура путем изменения его параметров.

Задача пересчета параметров контура возникает при изменении параметров контура пользователем и является частным случаем задачи расчета. Например, пользователь может изменить размер или положение точки.

Так как параметр влияет на связь, а связь влияет на параметр, то нужно пересчитать все компоненты исходного графа, куда входят модифицированные параметры. Эти параметры объявляются полусвободными и не участвуют в пересчете. Очевидно, что максимальное количество параметров, изменяемых одновременно, ограничено.

4. Вместо заключения

Кратко сформулируем основные этапы расчета параметров:

1. Построение максимального паросочетания для пользовательских связей;
2. Анализ возможных конфликтных ситуаций;
3. Добавление автоматических связей;
4. Построение максимального паросочетания с максимальным суммарным приоритетом для автоматических связей;
5. Построение графа пересчета;
6. Выделение компонент сильной связности;
7. Построение сети пересчета;
8. Последовательный расчет всех параметров.

На заключительном этапе решаются системы нелинейных уравнений с произвольным количеством уравнений. Это один из самых критичных этапов работы алгоритма. К решателю систем уравнений (от англ. solver) предъявляются высокие требования, т.к. он должен устойчиво работать для любых систем уравнений.

В худшем случае, при параметризации плоских контуров может появиться система уравнений, в которой число уравнений равно числу связей в контуре.

Требования, предъявляемые к решателю, зависят от сложности используемых в параметрических моделях связей. Например, для решения систем рациональных уравнений достаточно решателя, использующего метод Ньютона.

Данный подход к расчету параметров может быть одинаково успешно применен к двумерным и трехмерным геометрическим объектам.

В заключение хочется сказать о полученных практических результатах. Описанные выше алгоритмы реализованы авторами статьи и успешно применяется для получения параметризованных плоских контуров. Для того, чтобы оценить эффективность методов параметризации необходимо сравнить практические примеры и теоретические основы подходов, пользуясь тем малым, что написано в руководствах пользователя, обзорных статьях и рекламных брошюрах.

Список использованных источников

1. Рутковский И.В. CAD/CAM/CAE – 20 лет развития
<http://www.nocikt.kgtu.runnet.ru/bulletin/b1s2.html>
2. Параметрическое моделирование. Словарь терминов. <http://www.niac.ru/graphinfo.nsf/>
3. Параметризация - анализ по прессе в Интернете.
<http://vm.msun.ru/Vectorwi/Vbsdoc/Parametr.htm>
4. Кононов В., Миронова М., Левкин Д. Autodesk Inventor - уникальный инструмент для инженеров в новом тысячелетии. <http://www.cad.ru/publ/p11.shtml>
5. Аскон. КОМПАС-3D http://www.cps.ru/vendors_ru/ascon/kompas3d.shtml
6. Б.Кузьмин, В.Хараджиев SprutCAD: особенности национальной параметризации САПР и Графика № 09 2001 год
7. В. Жигулин О том, как твердое тело может быть слишком твердым, или Взгляд на параметризацию сбоку "САПР и графика" №1 2000 год
8. Enrico Motta, Zdenek Zdrahal Parametric Design Problem Solving
<http://ksi.cpsc.ucalgary.ca/KAW/KAW96/motta/pardes-banff.html>
9. Javier Monedero Parametric design. A review and some experiences
<http://info.tuwien.ac.at/ecaade/proc/moneder/moneder.htm>
10. David Rosen Parametric modeling
<http://www.srl.gatech.edu/education/ME6175/notes/ParamModel/ParamModeling.html>
11. М.П.Головин CAD-системы глазами пользователя
<http://www.nocikt.kgtu.runnet.ru/bulletin/b1s4.html>
12. Л.В. Баранов, Д.В. Сёмин. "Современные тенденции развития инструментальных средств разработки программного обеспечения для САПР"
http://magazine.stankin.ru/arch/n_17/29/index.htm
13. М.О.Асанов, В.А.Баранский, В.В.Расин. Дискретная математика: Графы, матроиды, алгоритмы – Ижевск: НИЦ "РХД", 2001. – 288 с.:ил.
14. А.Ахо, Дж.Хопкрофт, Дж.Ульман. Построение и анализ вычислительных алгоритмов – М.: Мир, 1979. – 536 стр.:ил.
15. Н.Кристофидес. Теория графов. Алгоритмический подход – М.: Мир, 1978. – 432 стр. :ил.
16. Ф.Харари. Теория графов – М.: Мир, 1973. – 300 стр.