УДК: 004.94:519.854.3
DOI 10.53403/9785951505071_2022_512

АДАПТИВНАЯ ГЕНЕРАЦИЯ РЕГУЛЯРНЫХ РАСЧЕТНЫХ СЕТОК НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

И. А. Шумилов, А. Е. Тимофеев, А. Л. Потехин

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, Саров

Для формирования расчетных сеток на поверхностях геометрических моделей существуют различные стратегии. Целью настоящего исследования является разработка алгоритма блочной генерации регулярных сеток с представлением граничных условий задания сетки в виде системы целочисленных линейных уравнений. Критерием задачи целочисленного программирования является минимизация числа изменений начального распределения узлов сетки на границе. Для решения составленной задачи используется алгоритм решения задач целочисленного линейного программирования, который на практике быстро находит допустимое решение, близкое к оптимальному и может быть использован для вычисления глобального оптимума. Разработанный алгоритм позволяет при необходимости изменять начальное число сеточных узлов на границе модели для применения блочной генерации регулярной сетки. Разработанный алгоритм позволил в рамках препостпроцессора пакета программ ЛОГОС получить возможность построения поверхностных сеток, обеспечивающих согласование сеточных фрагментов с разным уровнем детализации.

Ключевые слова: целочисленное линейное программирование; поверхностная сетка; расчетная сетка; блочная генерация.

Введение

Полностью автоматизированных генераторов поверхностных сеток, позволяющих построить регулярную поверхностную сетку на сложных поверхностях, не существует (рис. 1). Перспективным выглядит компромиссный подход: необходимо произвести дополнительное нарезание поверхностей на более простые, в которых возможно построение регулярных сеток. На простых поверхностях построение сетки достижимо путем их разделения на виртуальные регулярные блоки по заданной схеме с согласованной на границах блоков сеткой (рис. 2). Под регулярным блоком здесь мы будем понимать четырехугольник, у которого противоположные ребра имеют одинаковое число сеточных узлов. В развитых коммерческих продуктах под простыми поверхностями подразумеваются 3–6 угольные грани [1], а также фрагменты окружностей.

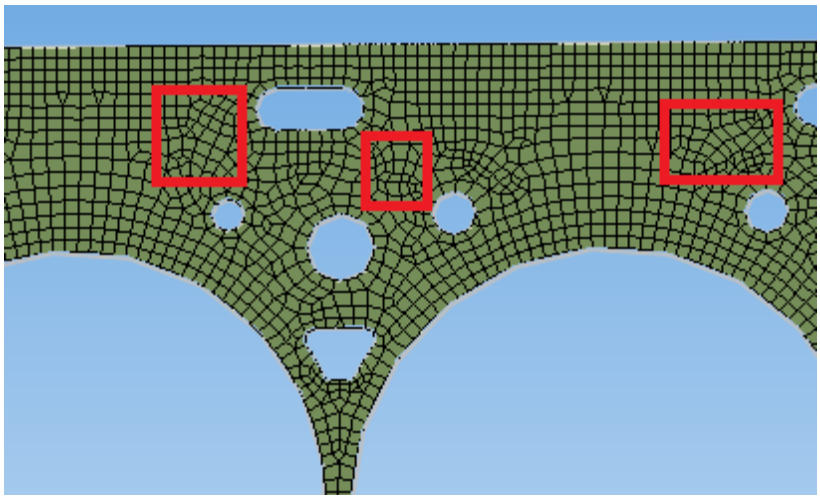

Рис. 1. Фрагмент сложной поверхности с сеткой, построенной фронтальным алгоритмом. Красным отмечены места схождения фронта генерации.

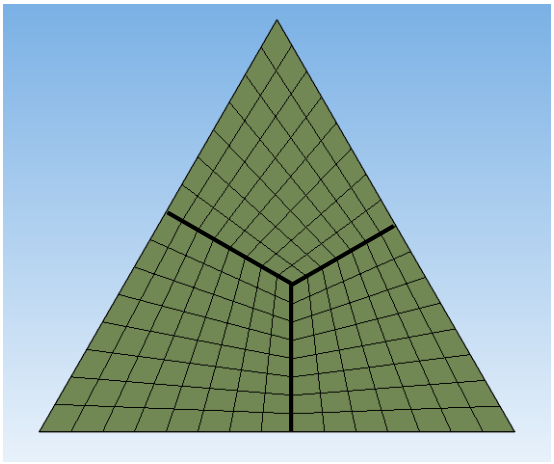

Рис. 2. Пример выделения трех виртуальных блоков на треугольной грани для построения регулярной сетки. Границы виртуальных блоков выделены черной линией.

Согласование виртуальных блоков

Очевидно, что разделение простых поверхностей на согласованные регулярные блоки возможно не всегда, а лишь при удовлетворении некоторых условий по числу узлов на их границах. В ходе построения поверхностных сеток на многогранном теле крайне желательно сохранение общего размера сеточных элементов. При этом число ячеек на границах граней может варьироваться. Для поддержания такой возможности ставится задача разбиения грани с минимальным изменением начального числа сеточных узлов контура грани.

Рассмотрим задачу выделения и согласования виртуальных блоков на грани на примере построения переходного сеточного слоя на четырехугольной грани. Под переходным слоем будем подразумевать сетку специального вида, обеспечивающую согласование сеток на соседних гранях.

Дана четырехугольная геометрическая грань вытянутой формы, на которой необходимо построить сетку, расширяющуюся с n узлов до m узлов вдоль широкой границы, где $n < m$. При этом на противоположных концах узкой границы число сеточных узлов одинаковое. Контур грани зада-

ется в виде набора ребер, представляющих собой геометрические кривые. Каждая кривая определяется однопараметрическим уравнением. Для каждого ребра заданы начальное число и координаты сеточных узлов. Также обозначим каждую вершину грани. Пример грани с исходными данными на ней изображен на рис. 3.

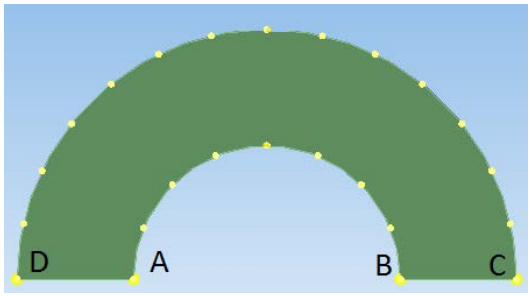

Рис. 3. Грань с заданным начальным положением
сеточных узлов на границе

На грани требуется построить поверхностную сетку, используя блоки трех видов, представленных на рис. 4. Внутренние области каждого блока представляют собой одну сеточную ячейку.

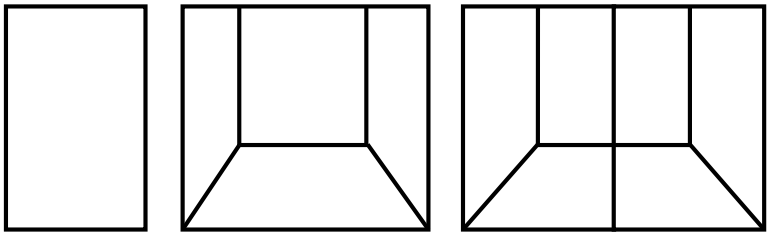

Рис. 4. Типы блоков, используемые для формирования переходного сеточного слоя

Пусть x_1, x_2, x_3 – количество блоков по каждому из трех типов, использованных для построения переходного сеточного слоя.

Для поиска набора блоков, наилучшим образом удовлетворяющего начальным ограничениям строится система уравнений:

$$\begin{cases} AB = \delta_1 + x_1 + x_2 + 2 * x_3 \\ CD = \delta_2 + x_1 + 3 * x_2 + 4 * x_3 \\ x_i \geq 0, i = \overline{1,3} \\ x_i \in \mathbb{Z}, i = \overline{1,3} \\ \delta_1 \in \mathbb{Z}, \delta_2 \in \mathbb{Z} \end{cases}, \quad (1)$$

где AB и CD – число сеточных ячеек на ребрах AB и CD соответственно (для простоты обозначения вершин выбраны таким образом, что $AB \leq CD$); δ_1 – число сеточных ячеек, добавленных к ребру AB; δ_2 – число сеточных ячеек, добавленных к ребру CD.

Для контроля за изменением числа сеточных узлов на ребрах AB и CD можно добавить дополнительные ограничения $\delta_1 = 0$ или $\delta_2 = 0$.

Критерием оптимизации является минимизация числа изменений на ребрах грани.

$$F(x) = |\delta_1| + |\delta_2| \to min \quad (2)$$

Для представления задачи (1) – (2) виде задачи целочисленного линейного программирования (ЦЛП) необходимо привести критерий к линейному виду. Для этого можно рассмотреть четыре задачи, в каждой из которых используется различная комбинация знаков функций модуля. Далее производится сравнение значений критерия по каждой задаче и выбирается наилучшее.

При решении задачи целочисленного линейного программирования получаем число каждого типа блоков, а также новое число ячеек на ребрах AB и CD, добавляя к существующим значениям δ_1 и δ_2 соответственно. После этого блоки равномерно, с учетом их ширины, распределяются по поверхности грани. Пример построенного переходного слоя изображен на рис. 5.

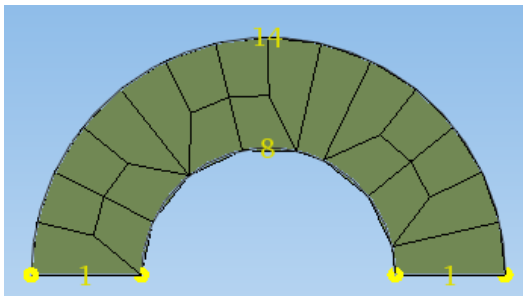

Рис. 5. Построенный переходный сеточный слой
на грани при помощи виртуальных блоков

Постановка задачи ЦЛП

Рассмотрим задачу целочисленного линейного программирования в общей постановке:

$$\begin{cases} Ax \le b \\ Dx = f \\ x \in \mathbb{Z}^n \end{cases}, \quad c^T x \to max \tag{3}$$

где c, b, f – векторы, а A и D – матрицы, все элементы которых являются целыми числами.

Ограничения на переменные, если присутствуют, включены в $Ax \le b$.

Обозначения a_i — это строки матрицы A.

Алгоритм полагается на два основных способа трансформации задачи ЦЛП:

1. Линейное преобразование переменных: мы производим замену переменных типа $x = Mx^1$, где x – старый набор переменных (система координат), x^1 – новый набор переменных, а M – унимодулярная матрица. Такое преобразование часто позволяет упрощать задачу и всегда сохраняет отношения один к одному между решениями старой и новой задачи.

2. Отсечение допустимого решения: если мы уже получили одно допустимое решение x^*, со значением критерия g, то мы можем ввести в задачу новое ограничение: $c^T x \ge g + 1$. Новая задача не содержит решения x^*, и все решения, которые она содержит, будут строго лучше x^* или любого другого решения, отсеченного новым условием. Таким образом, если глобальный оптимум новой задачи существует, то он является и глобальным оптимумом оригинальной задачи, если не существует, то x^* является глобальным оптимумом.

Алгоритм решения задачи ЦЛП

I. Если в задаче есть равенства, то матрицу равенств приведем к нормальной форме Смита (применив полученную матрицу замены переменных к неравенствам соответственно) [2, 3]. В получившейся задаче равенства содержат не более одной переменной и могут быть решены последовательно.

Возможны три случая:

1. Уравнения противоречивы (например, $x = 3$ и $x = 5$) или не имеют целочисленных решений (например, $2x = 3$ или $0x = 6$), следовательно модель не имеет решений.

2. Все переменные назначены. Если такое назначение удовлетворяет всем неравенствам, то получено единственное решение, иначе решений нет.

3. Часть переменных (равная количеству независимых уравнений) назначена, остальные переменные входят в систему неравенств, которая формулируется так:

$$\begin{cases} Ax \le b \\ x \in \mathbb{Z}^n \\ c^T x \to max \end{cases}, \tag{4}$$

где c, b – векторы, а A – матрица, все элементы которых являются целыми числами. В системе n переменных и m неравенств.

II. Проверим столбцы матрицы A на линейную зависимость. Если линейная зависимость найдена, проведем линейное преобразование, которое обратит зависимые столбцы в нули. Если у соответствующей переменной в критерии стоит ненулевой коэффициент, то, изменяя эту переменную, критерий задачи можно сделать неограниченно большим или малым при условии, что оставшаяся система совместна. Если коэффициент в критерии нулевой, то эта переменная не имеет никакого значения и может быть назначена нулем. После процедуры остается система линейных неравенств с гарантированно линейно независимыми переменными.

III. Проверим систему неравенств на неограниченность области допустимых значений. Для этого решим m систем нецелочисленных неравенств вида:

$$\begin{aligned} a_i x^j &\le -1,\ j = i,\ i \in [1..m] \\ a_i x^j &\le\ \ 0,\ j \ne i,\ i \in [1..m] \end{aligned} \tag{5}$$

для j от 1 до m симплекс-методом. Назначим $x^j = 0$, если -ая задача не имеет решений, и домножим x^j на общий знаменатель, если решение нашлось дробное – тогда эти вектора всегда существуют и являются целочисленными (умножение вектора x^j на любое число больше единицы не может нарушить условия задачи). Введем вектора $x' = \sum_{j=1}^{m} x^j$ и $b' = Ax'$.

Вектора x^j имеют смысл направления, в котором можно перемещать допустимое решение системы в n-мерном пространстве таким образом, что оно может бесконечно удаляться от гиперплоскости -ого ограничения, не нарушая остальные ограничения. Вектор x' – направление, по которому можно отдалить допустимое решение от максимального количества гиперплоскостей ограничений, не нарушив оставшиеся. Вектор b' – вектор относительных скоростей, с которыми решение будет отдаляться.

Возможны три случая:

1. Вектор b' – нулевой. В таком случае область допустимых решений ограничена.

2. Вектор b' состоит только из ненулевых отрицательных целых чисел. Найдем минимальное положительное целое число k, что $kb' \le b$. Тогда $kb' = kAx' = A(kx') \le b$ и kx' является допустимым решением исходной системы неравенств. Если при увеличении k критерий ухудшается, отсечем допустимое решение и решим получившуюся систему начиная с пункта III, в противном случае критерий не ограничен, и задача считается решенной.

3. Вектор b' имеет как нулевые, так и отрицательные значения. В таком случае составим систему неравенств, выбрав из исходной только те, которые соответствуют нулевым значениям b': $A^* x^* \le b^*$. Решим эту систему, начиная с пункта II. Если решений нет, то и исходная система несовместна. Если нашлось решение x^*, то найдем допустимое решение исходной системы в виде $x = x^* + kx'$, где k – минимальное положительное целое число. Если при увеличении k критерий ухудшается, отсечем допустимое решение и решим получившуюся систему начиная с пункта III, в противном случае критерий не ограничен, и задача считается решенной.

IV. Решим ограниченную систему линейных неравенств. В первую очередь воспользуемся алгоритмом линейной упаковки [4] для минимизации ограничивающего гиперпараллелепипеда области допустимых решений. Это уменьшит ветвление алгоритма и одновременно увеличит шанс успеха на каждом шаге.

Затем построим модифицированную версию задачи: оставим матрицу коэффициентов A и критерий c, но изменим вектор свободных членов следующим образом:

$$b_i^k = b_i - \sum_{j=1}^{n} \frac{|A_{i,j}|}{2},\ i \in [1..m] \tag{6}$$

Решим задачу $Aq \leq b^k,\ c^T q \rightarrow max,\ q \in \mathbb{R}$ симплекс-методом. Вектор q имеет смысл центра единичного гиперкуба, полностью принадлежащего области допустимых решений исходной системы неравенств. Единичный гиперкуб всегда содержит хотя бы одну целочисленную точку, и для вычисления достаточно округлить координаты центра гиперкуба [4].

Если удастся получить дробный вектор q, удовлетворяющий условиям задачи, то допустимое решение исходной целочисленной задачи найдено, его можно отсечь и продолжать решение с пункта IV**.** В противном случае выберем переменную с самым маленьким диапазоном в области допустимых значений, назначаем ей все возможные значения и рекурсивно решаем каждую получившуюся подзадачу (в которых будет на одну переменную меньше) с пункта IV**.**

Заключение

В статье рассмотрен блочный подход к генерации поверхностной расчетной сетки на примере создания переходного сеточного слоя. Приведено описание алгоритма решения задачи целочисленного линейного программирования, используемого для согласования числа сеточных узлов на границах грани и виртуальных блоков.

В качестве недостатка описанного подхода можно указать использование линейного критерия в задаче оптимизации. Возможный путь улучшения для задачи сеткогенерации – это переход к решению нелинейных оптимизационных задач, в частности включение функции модуля в критерий.

В тоже время, даже текущий вариант алгоритма позволил решить важную практическую задачу в рамках препостпроцессора пакета программ ЛОГОС. В целях экономии времени, затрачиваемого на математическое моделирование, инженеры стремятся выдержать примерно одинаковый размер сеточных элементов в разных частях конструкции, не допуская лишнего избыточного сгущения. Это приводит к появлению ситуаций с разным уровнем сеточной детализации на различных фрагментах геометрической модели, как показано на рис. 6. На основе описанного в работе алгоритма в рамках препостпроцессора реализован специальный режим сеткогенерации, решающий задачу согласования.

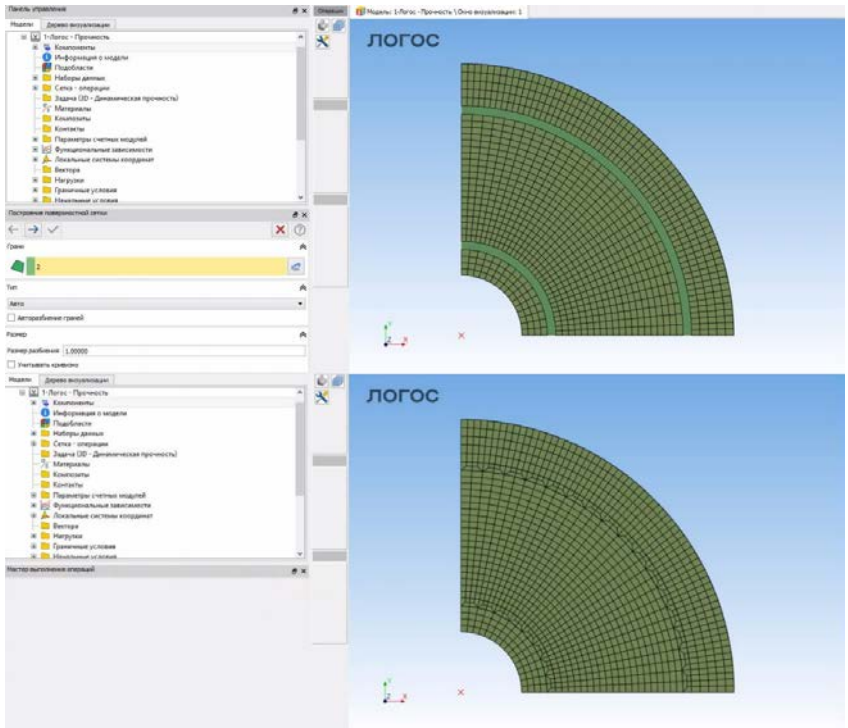

Рис. 6. Исходные данные для работы алгоритма (сверху) и результат применения (внизу)

Литература

1. Афраймович Л. Г., Прилуцкий М. Х., Потехин А. Л., Старостин Н. В., Тимофеев А. Е, Филимонов А. В. Распознавание геометрических сущностей с целью последующей генерации сеток для моделирования прочностных процессов // Супервычисления и математическое моделирование. Труды XVI Международной конференции» / Под ред. Шагалиева Р. М. – Саров: ФГУП «РФЯЦ-ВНИИЭФ». 2016. С. 50–51

2. Шумилов, И. А. Алгоритмы редукции систем целочисленных уравнений и неравенств: выпускная квалификационная работа бакалавра по напр. Прикладная информатика: 09.03.03: защищена 19.06.17. Н. Новгород, 2017. С. 56 .

3. Шумилов И. А. Алгоритмы редукции систем целочисленных уравнений и неравенств // XXIII Нижегородская сессия молодых ученых. Математические науки. Тезисы докладов. Нижний Новгород. 2018.

4. Шумилов, И. А. Алгоритмы решения задач целочисленного линейного программирования: магистерская диссертация по напр. Прикладная информатика: 09.04.03: защищена 10.06.19. Н. Новгород, 2019. С. 41 .

ADAPTIVE GENERATION OF REGULAR CALCULATION NETWORKS BASED ON THE SOLUTION OF TASKS OF AN INTEGER LINEAR PROGRAMMING

I. A. Shumilov, A. E. Timofeev, A. L. Potekhin

Russian Federal Nuclear Center –
All-Russian Scientific Research Institute of Experimental Physics, Sarov

For the formation of computational grids on the surfaces of geometric models, there are various strategies. The aim of this study is to develop an algorithm for block generation of regular grids with a representation of the boundary conditions for specifying a grid in the form of a system of integer linear equations. The criterion for the integer programming problem is to minimize the number of changes in the initial distribution of grid nodes at the boundary. To solve the problem, an algorithm is used to solve integer linear programming problems, which in practice quickly finds a feasible solution close to optimal and can be used to calculate the global optimum. The developed algorithm allows, if necessary, to change the initial number of grid nodes at the model boundary for applying block generation of a regular grid. The developed algorithm made it possible, within the framework of the prepost processor of the LOGOS software package, to obtain the possibility of constructing surface grids that ensure matching of mesh fragments with different levels of detail.

Key words: integer linear programming; surface grid; computational grid; block generation.