

## **ОБ ОДНОМ ПОДХОДЕ К УМЕНЬШЕНИЮ ВЫЧИСЛИТЕЛЬНЫХ ЗАТРАТ ПРИ ПОСТРОЕНИИ КРИВОЛИНЕЙНЫХ СЕТОК <sup>1</sup>**

**А.А.ЧАРАХЧЬЯН**

*(117967 Москва, ГСП-1, ул. Вавилова 40, ВЦ РАН)*

Предлагается модификация известного сеточного генератора, позволившая значительно сократить вычислительные затраты на построение сеток с большим числом узлов в задачах с подвижными границами.

### **Введение**

В настоящей работе предлагается модификация известного метода [1], который занимает особое место среди методов построения регулярных (т.е. с естественной двумерной нумерацией узлов) сеток в областях с криволинейными границами. Этот метод гарантирует выпуклость на каждой итерации всех четырехугольных ячеек сетки при заданных координатах граничных узлов (за естественным исключением тех ячеек, невыпуклость которых predetermined соответствующим заданием граничных узлов), если только множество таких "выпуклых" сеток не пусто. На практике это означает возможность построения сетки из выпуклых четырехугольников при любом искривлении граничных линий. Наиболее надежные из известных до работы [1] эллиптических сеточных генераторов, представленные, например, в книгах [2,3], могли обеспечить построение невырожденной сетки только в пределе бесконечно большого числа узлов.

Естественной областью применения метода [1] являются нестационарные гидродинамические задачи с сильной деформацией границ раздела сред. В настоящее время для задач такого типа наи-

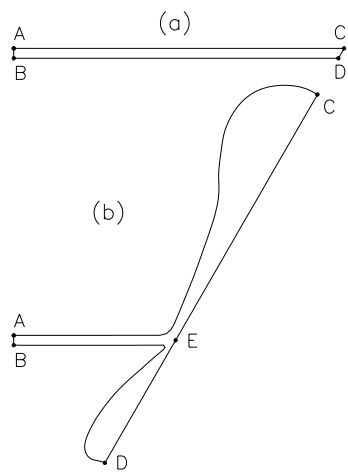
---

<sup>1</sup>Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 95-01-01161).

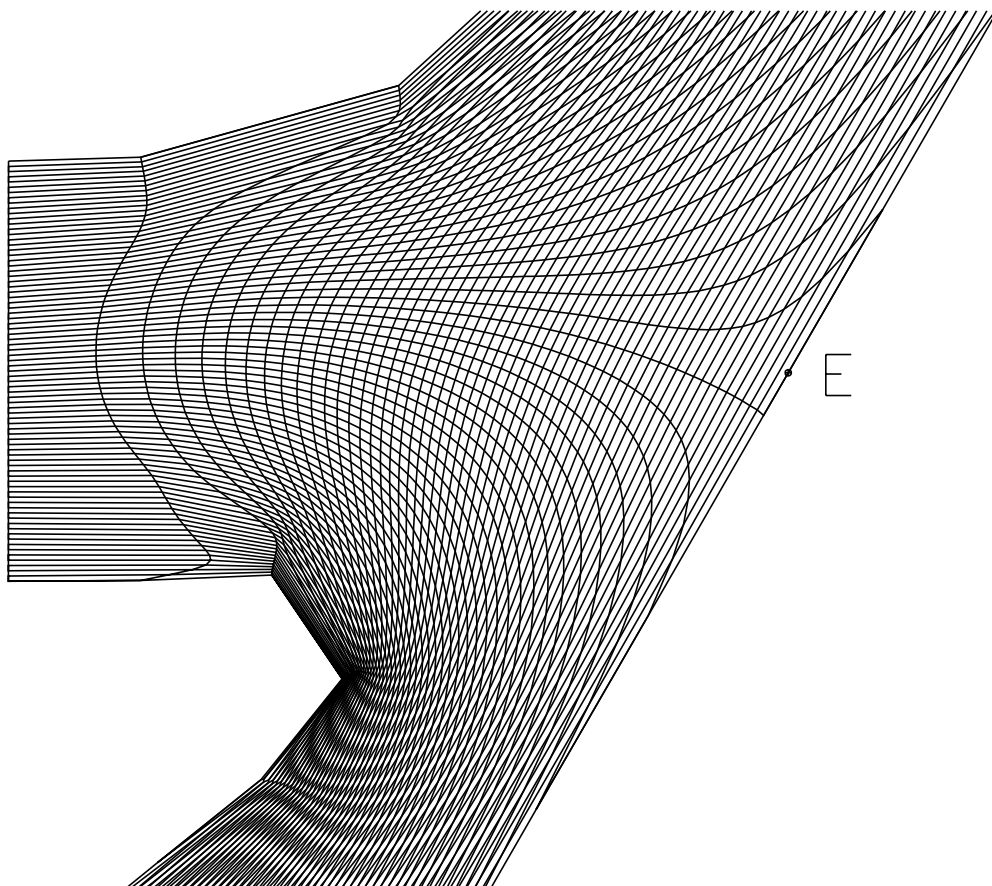
более часто используются либо методы сквозного счета через границы раздела сред типа метода "частиц в ячейке", либо свободно-лагранжевы методы, базирующиеся на нерегулярных сетках с изменяющимся шаблоном разностной схемы. Работа [1] кардинальным образом расширила и возможность использования для решения указанных задач методики [2], базирующейся на регулярных сетках с явным выделением границ раздела сред в виде некоторых линий сетки. Все эти три подхода имеют как свои достоинства, так и свои недостатки, которые здесь обсуждаться не будут.

В качестве примера использования метода [1] рассмотрим ситуацию, возникавшую при моделировании удара тонкой фольги по конической мишени. На фиг. 1 показана форма фольги в цилиндрических координатах: (a) — в начальный момент времени, (b) — через некоторое время после удара. Метод [1] позволяет построить для области на фиг. 1b сетку из выпуклых четырехугольников той же структуры, что и естественная сетка для области на фиг. 1a, т.е. с вершинами *A*, *B*, *C* и *D* в качестве угловых узлов. Фрагмент такой сетки в окрестности точки *E*, где сеточные линии наиболее сильно искривляются, показан на фиг. 2. Вопрос о точности численного решения, получаемого на такой сетке, выходит за рамки данной статьи. Укажем только, что сама возможность получения численного решения предполагает и возможность изучения его точности путем сравнения результатов расчета на сетках с разным числом узлов.

Для решения задач типа показанной на фиг. 1 нельзя полностью исключить и возможность использования блочных сеток, построенных следующим образом. Расчетная область делится несколькими априорно задаваемыми линиями на блоки, в каждом из которых строится сетка своей собственной структуры, определяемой индексацией граничных узлов. Впервые такой принцип построения сетки был описан, по-видимому, в [2], и в настоящее время широко используется (см., например, доклады по пакетам программ для построения структурированных сеток в [4]). При построении блочной сетки



Фиг. 1:



Фиг. 2:

для области на фиг. 1*b* естественно ввести три блока, отвечающие обеим "языкам" и центральной части. В этом случае сеточным линиям не нужно слишком сильно изгибаться внутри каждого блока, и поэтому для построения такой блочной сетки может использоваться не только метод [1], но и другие эллиптические сеточные генераторы, например приведенные в [2,3]. Такая блочная сетка будет, по-видимому, и более качественной, по сравнению с сеткой, фрагмент которой показан на фиг. 2. Однако реализация вычислений на такой сетке наталкивается на серьезные трудности. В самом деле, по мере увеличения "языка" должно увеличиваться и число узлов граничной линии сетки, которые попадают в этот "язык". Поэтому сеточная линия, разделяющая блоки, не может все время проходить через один и тот же узел граничной линии сетки. Это приводит к необходимости создания специального алгоритма периодического изменения структуры блоков, включающего в себя громоздкие процедуры переинтерполяции поля течения между сетками разной структуры. В то же время возможность получения на таком пути более точного численного решения представляется далеко не очевидной из-за возможного падения точности, вызванного частым использованием указанных процедур переинтерполяции. Напомним и о сильной конкуренции для рассматриваемого класса задач со стороны свободно-лагранжевых методов. Все эти соображения позволяют высказать осторожный пессимизм по поводу перспектив использования блочных сеток в такой ситуации, когда в процессе решения задачи требуется частое изменение структуры блоков. Этот пессимизм подтверждает и анализ докладов из [4]. Среди всех докладов, посвященных блочным сеткам, автору не удалось найти ни одного, где задачи с подвижными границами указывались бы в качестве возможных приложений.

Вернемся к методу [1]. За его повышенную надежность придется платить увеличением затрат машинного времени. Как показало сравнение, с результатами которого любезно ознакомил автора С.А.Иваненко, вычислительные затраты на одну итерацию для ме-

тогда [1] примерно в 7 раз выше, чем для варианта метода Winslow из [2]. Хотя метод [1] генерирует сетку из выпуклых четырехугольников после каждой итерации, проблема уменьшения вычислительных затрат метода для сеток с большим числом узлов стоит достаточно остро, по крайней мере в применении к задачам с подвижными границами. Сильная "недоитерированность" сетки часто приводит к тому, что внутренние узлы перестают успевать за движением границ. Вблизи одних границ могут возникнуть очень большие ячейки, что приводит к снижению точности расчета. Вблизи других границ, наоборот, могут возникать очень маленькие ячейки, что приводит к неоправданно маленькому шагу по времени. Возникают трудности и при построении начальной сетки для метода [1] на очередном шаге по времени.

Один путь уменьшить вычислительные затраты заключается в использовании компьютеров, допускающих параллельные вычисления. Настоящая работа развивает другое направление, использующее неравномерную сходимость итерационных методов по узлам сетки. Впервые рекомендация фиксировать в процессе итераций положения отдельных узлов была дана, по-видимому, в книге [2]. С.А.Иваненко [5] эффективно использовал этот прием в применении к методу [1] в интерактивном режиме. Было создано программное обеспечение (2D-CONVEX, разработчики С.А.Иваненко и С.А.Чернивецкий), позволяющее, в частности, выделить на экране дисплея часть сетки и выполнить какое-то количество итераций по методу [1] только для этой части, полагая остальные узлы сетки неподвижными. Во многих случаях удовлетворительную сетку можно было получить, выполнив основное число итераций в маленьких подобластях, где линии сильно искривлялись. Это наблюдение послужило первоначальным толчком к разработке предлагаемого здесь алгоритма.

### **§1. Сетки из выпуклых четырехугольников**

Напомним, в чем состоит метод [1]. На плоскости  $(x, y)$  нуж-

но построить сетку  $G = \{(x, y)_{ij}, i = 1, \dots, N; j = 1, \dots, M\}$  при заданных координатах граничных узлов  $(x, y)_{i1}, (x, y)_{iM}, (x, y)_{1j}, (x, y)_{Nj}$ . В каждой ячейке сетки  $i + 1/2, j + 1/2$  с узлами  $(x, y)_{i,j}, (x, y)_{i,j+1}, (x, y)_{i+1,j+1}, (x, y)_{i+1,j}$  рассматриваются четыре треугольника, которые можно построить по этим узлам. Узлы ячейки для краткости записи занумеруем от 1 до 4. Пусть  $J_k$  - удвоенная площадь треугольника с центральной вершиной  $k$ , например,  $J_1 = (x_4 - x_1)(y_2 - y_1) - (y_4 - y_1)(x_2 - x_1)$ . Система неравенств

$$[J_k]_{i+1/2, j+1/2} > 0, \quad k = 1, 2, 3, 4; \quad i = 1, \dots, N - 1; \quad j = 1, \dots, M - 1,$$

определяет сетку только из выпуклых четырехугольников, которую для краткости будем называть выпуклой сеткой. Рассматривается конечно-разностная функция

$$(1a) \quad I^h = \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} \left[ \frac{\varphi_{412} + \varphi_{123} + \varphi_{234} + \varphi_{341}}{4} \right]_{i+1/2, j+1/2},$$

$$(1b) \quad \varphi_{lmn} = \frac{(x_l - x_m)^2 + (y_l - y_m)^2 + (x_n - x_m)^2 + (y_n - y_m)^2}{J_m},$$

аппроксимирующая известный функционал в некотором вспомогательном пространстве, минимум которого достигается на гармоническом отображении заданной области на прямоугольник в этом пространстве. Далее ищется минимум функции (1), достигаемый на решении системы алгебраических уравнений

$$(2) \quad R_x = \frac{\partial I^h}{\partial x_{ij}} = 0, \quad R_y = \frac{\partial I^h}{\partial y_{ij}} = 0, \quad i = 2, \dots, n - 1; \quad j = 2, \dots, m - 1.$$

Функция  $I^h$  обладает следующим свойством. Если хотя бы одна из величин  $J_k$  для какой-то ячейки стремится к нулю, оставаясь при этом положительной, то  $I^h(G) \rightarrow +\infty$ . Наличие такого бесконечного барьера на границе множества выпуклых сеток и обеспечивает надежность метода [1], гарантируя выпуклость получаемой сетки, если только начальная сетка итерационного процесса выпукла. При

решении задач с подвижными границами получение начальной выпуклой сетки не является серьезной проблемой, поскольку в распоряжении вычислителя имеется заведомо выпуклая сетка с предыдущего временного слоя. Внутренние узлы начальной сетки берутся либо непосредственно из сетки предыдущего временного слоя, либо вычисляются с помощью какой-либо интерполяционной процедуры по смещениям граничных узлов на очередном шаге по времени. В результате начальная сетка в худшем случае оказывается близкой к выпуклой, что легко исправляется с помощью минимизации функции

$$(3) \quad I_D = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \sum_{k=1}^4 \left( [\epsilon - J_k]_{i+1/2, j+1/2} \right)_+^2, \quad (f)_+ = \max(0, f),$$

с разумным выбором  $\epsilon > 0$ .

Итерационный процесс последовательного уточнения решения уравнений (2) является, как и в [6], квазиньютоновской процедурой

$$(4a) \quad \tau_s R_x + \frac{\partial R_x}{\partial x_{i,j}} (x_{i,j}^{s+1} - x_{i,j}^s) + \frac{\partial R_x}{\partial y_{i,j}} (y_{i,j}^{s+1} - y_{i,j}^s) = 0,$$

$$(4b) \quad \tau_s R_y + \frac{\partial R_y}{\partial x_{i,j}} (x_{i,j}^{s+1} - x_{i,j}^s) + \frac{\partial R_y}{\partial y_{i,j}} (y_{i,j}^{s+1} - y_{i,j}^s) = 0,$$

где  $s$  — номер итерации. Итерационный параметр  $\tau_s$  выбирается так, чтобы, с одной стороны, сетка на каждой итерации оставалась выпуклой, и, с другой стороны, суммарная квадратичная невязка уравнений (2)

$$(5) \quad W = \sum_{i,j} (R_x^2 + R_y^2)_{i,j}$$

была бы по возможности минимальной. Подробности алгоритма выбора итерационного параметра приведены в [1].

По сравнению с методом Ньютона, учитывающем все вторые производные функции (1), итерационная процедура (4) обладает

значительно более низкой сходимостью. В то же время здесь нет проблемы решения системы линейных алгебраических уравнений большой размерности.

Из функций (1), (3) естественно исключить те слагаемые, которые зависят только от координат граничных узлов. Это позволяет использовать метод и тогда, когда граничные линии сетки пересекаются под тупым углом, предопределяя тем самым невыпуклость соответствующей угловой ячейки сетки.

Производные, входящие в (4), вычисляются следующим образом. Массивам, в которых хранятся эти производные, предварительно присваиваются нулевые значения, а затем проводится перебор всех треугольников сетки и к соответствующим элементам массивов добавляются соответствующие производные от функции  $\varphi_{lmn}$  из (1б).

Для управления расположением узлов сетки описанный выше алгоритм применяется к функции

$$L^h(G) = I^h(G) + \sum_{i,j} f_{ij} S_{ij},$$

где  $f_{ij}$  — элементы управляющего массива,  $S_{ij}$  — площади ячеек сетки. Идея такого способа управления заимствована из [6].

## §2. Итерации на подмножествах узлов

В качестве характеристики степени сходимости метода в каждом узле естественно взять относительную невязку уравнений (2)

$$\delta_{ij} = \sqrt{(R_x)_{ij}^2 + (R_y)_{ij}^2} / \sigma_{ij},$$

где  $\sigma_{ij}$  имеет тот же порядок величины, что и отдельные слагаемые, входящие в  $(R_x)_{ij}$  и  $(R_y)_{ij}$ . Напомним, что

$$(R_x)_{ij} = \sum_k (a_k + b_k), \quad (R_y)_{ij} = \sum_k (c_k + d_k),$$

где суммирование ведется по тем двенадцати треугольникам, которые дают ненулевой вклад для узла  $(i, j)$ , а через  $a_k$ ,  $b_k$ ,  $c_k$  и  $d_k$  обозначены слагаемые, которые получаются при соответствующем



дифференцировании функции  $\varphi_{lmn}$  из (16). В качестве  $\sigma_{ij}$  можно взять

$$\sigma_{ij} = \sum_k \sqrt{(|a_k| + |b_k|)^2 + (|c_k| + |d_k|)^2}.$$

Обозначим через  $\Omega_0$  множество внутренних узлов сетки. Уменьшение вычислительных затрат будет достигаться за счет выделения на каждой итерации  $s$  множества  $\Omega^s$ , для узлов которого проводятся вычисления, в то время как остальные узлы  $(i, j) \in \Omega_0 \setminus \Omega^s$  полагаются неподвижными. Первоначальный вариант метода был достаточно прост. Задаются максимально допустимое значение относительной невязки  $\delta$  и целое число  $N \sim 5$ . После каждой  $s$ -той итерации, начиная с  $N$ -той, объявляются неподвижными те узлы множества  $\Omega^s$ , для которых  $\delta_{ij} \leq \delta$ . Если все четыре узла ячейки стали неподвижными, то такая ячейка выбрасывается из расчета. Итерационный процесс заканчивается, как только множество  $\Omega^s$  на очередной итерации оказывается пустым. Процесс сходил достаточно быстро, и в ряде практических расчетов позволил значительно сократить затраты на построение сетки без видимого ухудшения ее качества. Однако более тщательный анализ получаемых сеток показал, что такой процесс выбрасывания узлов может приводить к значительному росту как максимальной локальной невязки

$$D = \max_{ij \in \Omega_0} \delta_{ij},$$

так и суммарной квадратичной невязки  $W$ , определяемой формулой (5).

Для того, чтобы исправить дефект описанного выше процесса, надо учесть, что после  $s$ -той итерации параметр  $\delta_{ij}$  меняет свое значение не только в узлах множества  $\Omega^s$ . Обозначим через  $\gamma_{ij}$  множество 9-ти узлов в окрестности узла  $(i, j)$ :  $(i', j') \in \gamma_{ij}$ , если  $|i - i'| \leq 1, |j - j'| \leq 1$ . Пусть задано множество  $\Omega^s$  такое, что  $\delta_{ij} \leq \delta$  для  $(i, j) \in \Omega_0 \setminus \Omega^s$ . Получим множество  $\Omega^{s+1}$ , обладающее тем же свойством после  $s$ -той итерации, в которой подвижными полагались только узлы множества  $\Omega^s$ . Поскольку в уравнения (2)

для каждого узла  $(i, j)$  входят только координаты узлов  $(i', j') \in \gamma_{ij}$ , после выполнения  $s$ -той итерации параметр  $\delta_{ij}$  может изменить свое значение только в узлах множества

$$\bar{\Omega}^s = \bigcup_{ij \in \Omega^s} (\gamma_{ij} \cap \Omega_0).$$

Далее для получения множества  $\Omega^{s+1}$  надо вычислять  $\delta_{ij}$  при  $(i, j) \in \bar{\Omega}^s$  по значениям координат, полученных после  $s$ -той итерации. Хотя множество  $\bar{\Omega}^s$  может значительно превосходить множество  $\Omega^s$ , для сеток с большим числом узлов множество  $\bar{\Omega}^s$  может оставаться небольшим по сравнению с  $\Omega_0$ .

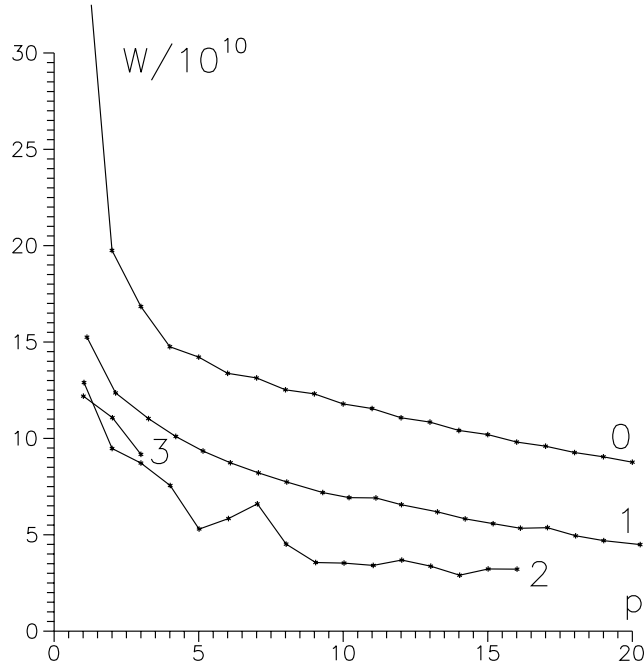
Множество  $\Omega^{s+1}$  можно получать, выбрасывая из  $\bar{\Omega}^s$  все те узлы, для которых  $\delta_{ij} \leq \delta$ . Для того примера, на котором тестировался алгоритм, более выгодным оказалось оставлять в множестве  $\Omega^{s+1}$  часть узлов с  $\delta_{ij} \leq \delta$ , если  $(i, j) \in \Omega^s$  и, хотя бы для одного узла  $(i', j') \in \gamma_{ij}$ ,  $\delta_{i'j'} > \delta$ .

Тест для сравнения описанного выше алгоритма с исходным вариантом метода взят из расчета задачи о высокоскоростном ударе тонкой фольги по конической мишени. Форма области показана на фиг. 1b. Сетка в левой недеформированной части области легко строится с помощью прямых линий. Поэтому настоящий метод применялся для расчета внутренних узлов только в правой части области, состоящей из примерно 4000 узлов. Внутренние узлы начальной сетки вычислялись с использованием некоторой интерполяционной процедуры по значениям смещений граничных узлов на очередном шаге по времени.

В качестве характеристики вычислительных затрат выбрана величина

$$p = \sum_s N(\bar{\Omega}^s) / N(\Omega_0),$$

где  $N(\Omega)$  — число узлов множества  $\Omega$ . Для исходного варианта метода  $p$  совпадает с номером итерации  $s$ . На фиг. 3 показаны зависимости квадратичной невязки  $W$  от вычислительных затрат  $p$  для трех значений  $\delta = 0.01, 0.02$  и  $0.03$  в описанном выше алгорит-



Фиг. 3: Зависимость квадратичной невязки  $W$  от вычислительных затрат  $p$ ; исходный вариант (0),  $\delta = 0.01$  (1), 0.02 (2) и 0.03 (3).

ме и для исходного варианта метода. Точками отмечены данные для значений  $p$  вблизи целых чисел. Видно, что по сравнению с исходным вариантом метода все три расчета с разными значениями  $\delta$  при одинаковых затратах приводят к значительно более "доитерированным" сеткам. Итерационный процесс для  $\delta = 0.03$  сошелся очень быстро и оказался небольшим по затратам. Расчет с  $\delta = 0.02$ , хотя и занимает большее время, уменьшает степень "недоитерированности" сетки. Расчет с  $\delta = 0.01$  оказался менее выгодным, но и он, как и указывалось выше, значительно выгоднее исходного варианта метода.

$\delta$	0.01	0.02	0.03
$W$	4.5	3.2	9.2
$p$	20	16	3
$p_0$	50	71	18

В таблице приведены затраты исходного варианта  $p_0$  в сравнении с затратами модифицированного варианта  $p$  для одних и тех же

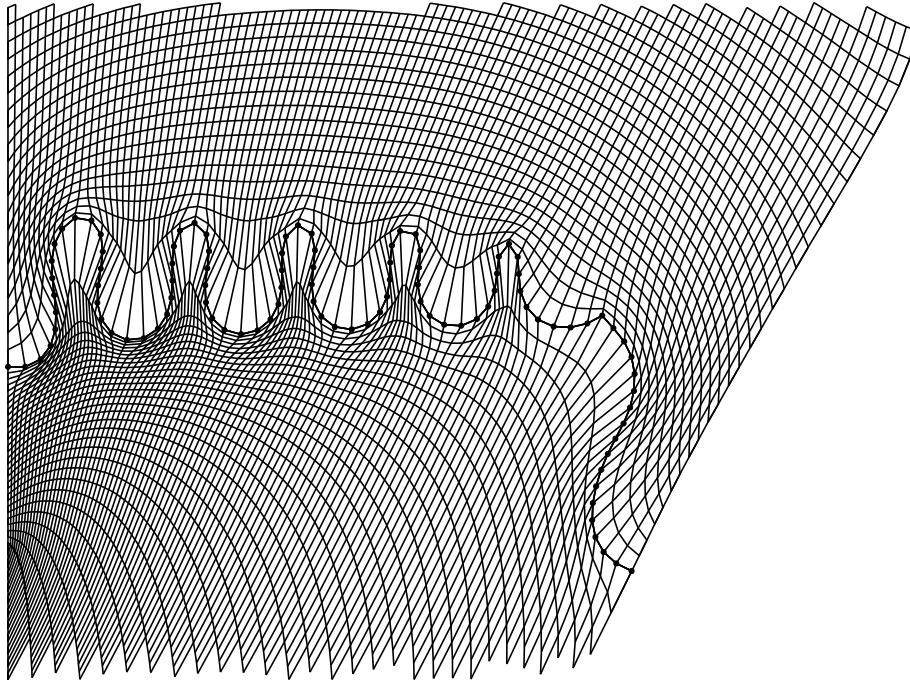
значений  $W$ . Уменьшение затрат в модифицированном методе колеблется от 2.5 раз для  $\delta = 0.01$  до 6 раз для  $\delta = 0.03$ . Что касается затрат исходного варианта метода, необходимых для достижения тех же значений локальной невязки  $D$ , что и в случае модифицированного варианта, то их вообще не удалось определить, поскольку даже для  $p_0 = 100$  невязка  $D$  в исходном варианте не опускалась ниже 0.04.

Была сделана попытка еще больше сократить вычислительные затраты за счет перехода к чисто ньютоновским итерациям с учетом всех вторых производных функции (1). Как и следовало ожидать, это привело к значительному падению невязки в области  $\Omega^s$  на каждой итерации. Однако скорость падения невязки в области  $\bar{\Omega}^s$  изменилась мало. В результате вычислительные затраты даже несколько возросли из-за их увеличения на этапе решения системы линейных уравнений.

Описанный выше метод применялся и при численном исследовании устойчивости границы раздела сред в задаче об импульсном воздействии на конические мишени. На фиг. 4 приведен типичный пример фрагмента сетки, где точками отмечены узлы сеточной линии, являющейся границей раздела. В некоторых вариантах расчета число узлов сетки выше и ниже этой линии достигало соответственно 16000 и 12000.

Сетка на фиг. 4 отражает известный недостаток метода [1]. Узлы сеточной линии, ближайшей к границе раздела, слишком сильно отходят от границы. Как было показано в [1], подходящим выбором управляющего массива  $f_{ij}$  этот недостаток можно устранить. В данном случае в этом не было необходимости, поскольку сравнение результатов, полученных на сетках с разным числом узлов, указывало на вполне удовлетворительную точность расчета.

В заключение укажем, что развитый здесь подход может оказаться полезным и в применении к методу построения адаптивных сеток [7,8].



Фиг. 4: Фрагмент сетки из задачи об устойчивости границы раздела сред.

#### ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. *Иваненко С.А., Чарахчьян А.А.* Алгоритм построения криволинейных сеток из выпуклых четырехугольников // Докл. АН СССР. 1987. Т. 295. №2. С. 280—283; Криволинейные сетки из выпуклых четырехугольников // Ж. вычисл. матем. и матем. физ. 1988. Т. 28. №4. С. 503—514.
2. *Годунов С.К., Забродин А.В., Иванов М.Я., Крайко А.Н., Прокопов Г.П.* Численное решение многомерных задач газовой динамики. М.:Наука, 1976.
3. *Thompson J.F., Warsi Z.U.A., Mastin C.W.* Numerical grid generation. N.Y. etc: North-Holland, 1985.
4. Numerical Grid Generation in Computational Field Simulations / Edited by B.K.Sony, J.F.Thompson, J. Hauser, and P.R.Eiseman. Proceedings of the 5th International Conference. Mississippi State University, 1996.

5. *Иваненко С.А.* Построение невырожденных сеток // Ж. вычисл. матем. и матем. физ. 1988. Т. 28. №10. С. 1498—1506.
6. *Brasbill J.U., Saltzman J.S.* Adaptive zoning for singular problems in two dimensions // J. Comput. Phys. 1982. V. 46. N3, P. 342—368.
7. *Иваненко С.А.* Адаптивные сетки и сетки на поверхностях // Ж. вычисл. матем. и матем. физ. 1993. Т. 33. №9. С. 1333—1351.
8. *Иваненко С.А.* Применение адаптивно-гармонических сеток для численного решения задач с пограничными и внутренними слоями // Ж. вычисл. матем. и матем. физ. 1995. Т. 35. №10. С. 1494—1517.