

СЕКЦИЯ 1.
***MATLAB — среда разработки инженерных
и научных приложений***

Часть 1.

Председатель:
к. т. н., доцент В. Г. Потемкин

Оглавление

Александров А. Г., Михайлова Л. С., Орлов Ю. Ф. АДАПЛАБ-М: ДИРЕКТИВА ДЛЯ САМОНАСТРОЙКИ ИСПЫТАТЕЛЬНОГО СИГНАЛА	6
Алексеев К. А., Исянов Р. Н. ВИРТУАЛЬНЫЕ ИСПЫТАНИЯ ДВУХКООРДИНАТНОГО СЕЙСМОПРЕОБРАЗОВАТЕЛЯ В СРЕДЕ MATLAB	18
Алексеев Е. Р., Федоров М. М., Чеснокова О. В., Литвинова Т. МОДЕЛИРОВАНИЕ ТЕПЛОВОГО СОСТОЯНИЯ АСИНХРОННЫХ ДВИГАТЕЛЕЙ ПРИ ПОНИЖЕННОМ НАПРЯЖЕНИИ В РЕЖИМАХ S1, S2, S3, S4.....	26
Андрижиевский А. А., Веремеева О. Н., Трифонов А. Г. ОРГАНИЗАЦИЯ ИНТЕРАКТИВНОГО ПРОЦЕССА В ЗАДАЧЕ ОПТИМИЗАЦИИ ТЕПЛООБМЕННЫХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ MATLAB.....	38
Асмыкович О. И. О ВЫЧИСЛЕНИИ ОБРАТНОЙ ДРАЙЗИНА В СРЕДЕ MATLAB.....	47
Бабаян П. В. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЮСТИРОВКИ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ИЗОБРАЖЕНИЙ В СРЕДЕ MATLAB.....	53
Бардушкин В. В. ИСПОЛЬЗОВАНИЕ ПАКЕТА MATLAB ДЛЯ ВЫЧИСЛЕНИЯ УПРУГИХ СВОЙСТВ ДИСПЕРСНОУПРОЧНЕННЫХ МАТРИЧНЫХ КОМПОЗИТОВ С НЕИЗОМЕТРИЧНЫМИ ВКЛЮЧЕНИЯМИ	55
Бардушкина И. В., Бардушкин В. В., Никитин А. Н., Сычев А. П., Яковлев В. Б. ИСПОЛЬЗОВАНИЕ СИСТЕМЫ MATLAB ПРИ МОДЕЛИРОВАНИИ НАПРЯЖЕННО-ДЕФОРМИРОВАННОГО СОСТОЯНИЯ ОЛИВИНОВ .	60
Бутковский О. Я., Логунов М. Ю. РАЗРАБОТКА ПАКЕТА ПРОГРАММ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ	71
Будак В. П., Козельский А. В., Савицкий Е. Н. РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ ТЕОРИИ ПЕРЕНОСА ИЗЛУЧЕНИЯ В СРЕДЕ MATLAB.....	76
Вержбицкая И. С. ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ РЕЖИМОВ КАТАЛИТИЧЕСКОГО ОКИСЛЕНИЯ НА ОСНОВЕ СИСТЕМЫ MATLAB	84
Волков К. Н., Емельянов В. Н. ПРИМЕНЕНИЕ ПАКЕТА MATLAB ДЛЯ РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ МЕХАНИКИ ЖИДКОСТИ И ГАЗА	92

Диаб Т. О. ОЦЕНИВАНИЕ ПАРАМЕТРОВ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ МЕТОДОМ МАКСИМАЛЬНОГО ПРАВДОПОДОБИЯ	116
Дубанов А. А. МЕТОДЫ ИНТЕРАКТИВНОГО КОНСТРУИРОВАНИЯ ПАРАМЕТРИЧЕСКИХ СОСТАВНЫХ ПОВЕРХНОСТЕЙ	121
Дьяченко Г. Г. ПРИМЕНЕНИЕ МАТЛАВ В СПЕКТРОСКОПИИ	131
Дьячук А. К., Медынский М. М. К ПРОБЛЕМЕ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ В СРЕДЕ МАТЛАВ	141
Ершова Т. А., Малафеев О. А. ТЕОРЕТИКО–ИГРОВАЯ МОДЕЛЬ ВХОЖДЕНИЯ В РЫНОК КУРНО N ФИРМ	159
Иглин С. П. GRTHEORY TOOLBOX — ИНСТРУМЕНТАРИЙ МАТЛАВ ДЛЯ РЕШЕНИЯ ЗАДАЧ НА ГРАФАХ	165
Кальнеус Е. В., Стась Д. В. РАЗРАБОТКА ИНТЕРАКТИВНОЙ СРЕДЫ РАСЧЕТА МАГНИТНОЙ СИСТЕМЫ МАРИ СПЕКТРОМЕТРА НА БАЗЕ ПАКЕТА МАТЛАВ	185
Кацюба О. А., Гущин А. В., Угнич К. А. ПАКЕТ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ РАСПРЕДЕЛЕНИЯ БИОПОТЕНЦИАЛОВ ЭЛЕКТРИЧЕСКОЙ АКТИВНОСТИ СЕРДЦА НА ОСНОВЕ РЯДА ЛАПЛАСА	198
Корчагин П. Н. АНАЛИЗ ДАННЫХ МНОГОФАКТОРНЫХ ЭКСПЕРИМЕНТОВ С ИСПОЛЬЗОВАНИЕМ КОДОВ ГРЕЯ	207

УДК 519.6

АДАПЛАБ-М: ДИРЕКТИВА ДЛЯ САМОНАСТРОЙКИ ИСПЫТАТЕЛЬНОГО СИГНАЛА

Александров А. Г., Михайлова Л. С.,

Институт проблем управления им. В.А. Трапезникова РАН, Москва,

E-mail: alex7@ipu.rssi.ru

Орлов Ю. Ф.

Московский государственный университет им. М. В. Ломоносова, Москва

1. Введение

В теории идентификации и адаптивного управления можно выделить два направления, зависящие от предположений о внешних возмущениях и помехах измерений. В первом из них возмущения и помехи являются случайными процессами типа «белый шум». Это направление имеет большую историю и во многом связано с методами наименьших квадратов [1]. Во втором направлении возмущения и помехи — неизвестные, ограниченные функции. На объект управления в этом случае подаются испытательные воздействия (испытательные сигналы), и применяются, в частности, методы инструментальных переменных [1], конечно-частотной идентификации [2], рандомизированной идентификации [3].

Естественное ограничение на выбор испытательного сигнала состоит в том, что он должен быть не связан (не коррелирован) с возмущениями и помехами. Во многих случаях это трудно выполнить, так как последние неизвестны. Кроме того, испытательный сигнал не должен заметно влиять на режим нормальной эксплуатации объекта управления и заметно снижать качество управления. Для выполнения этих условий необходима самонастройка испытательного сигнала в реальных условиях. В системе MATLAB программное обеспечение первого направления содержится [4] в пакетах расширения «System Identification Toolbox» и «Frequency Domain Identification». Кроме того, в первом из этих пакетов имеется программное обеспечение метода инструментальных переменных, где предполагается, что испытательный сигнал задается пользователем пакета априори. Такое задание возможно лишь при большом объеме информации о возмущениях, помехах и свойствах объекта.

В работе [5] описано MATLAB-приложение ADAPLAB-M, являющееся программным обеспечением конечно-частотной идентификации и

частотного адаптивного управления. Оно состоит из двух директив¹: «Конечно-частотная идентификация» и «Частотное адаптивное управление».

В методе конечно-частотной идентификации испытательный сигнал имеет определенную структуру: это сумма гармоник, число которых не превышает размерность вектора пространства состояний объекта. Некоррелированность гармоник этого сигнала (с возмущениями и помехами) проверяется в процессе идентификации, когда амплитуды и частоты испытательных гармоник известны. Интуитивно ясно, что испытательные частоты нужно выбирать из диапазона собственных частот объекта, в котором логарифмическая амплитудно-частотная характеристика объекта имеет изломы.

Предлагаемая ниже директива M011 «Самонастройка испытательного сигнала» служит для самонастройки амплитуд и частот этого сигнала в реальных условиях. Функции этой директивы могут быть использованы для планирования эксперимента по применению самонастройки. Для этой цели используется технологическая модель объекта (системы, управляемого процесса, описываемая дифференциальными уравнениями). Она строится на основе знаний специалиста (технолога) по объекту. Это некоторая предполагаемая модель истинного объекта и она может существенно отличаться от истинной его модели. Параметры алгоритма самонастройки, полученные с помощью технологической модели, являются первыми их значениями в реальном эксперименте.

2. Область применения

Рассмотрим асимптотически устойчивый линейный объект, описываемый уравнениями

$$\sum_{i=0}^{\gamma_1} Q^{(i)} s^i q = \sum_{j=0}^{\gamma_2} M^{(j)} s^j u + \sum_{k=0}^{\gamma_3} L^{(k)} s^k f, \quad y = \bar{D}q \quad (1)$$

где $q(t) \in \mathbb{R}^{n_1}$ — вектор переменных объекта, $y(t) \in \mathbb{R}^r$ — измеряемый выход, $u(t) \in \mathbb{R}^m$ — испытательный сигнал, $f(t) \in \mathbb{R}^u$ — внешнее возмущение. Эти уравнения в форме Коши имеют вид

$$\dot{x} = Ax + Bu + Mf, \quad y = Cx + Du, \quad t \geq t_0, \quad (2)$$

¹ Директива — это программа, формирующая интерфейс (ввода исходных данных, вывода промежуточных данных и результатов вычислений), производящая необходимые вычисления (содержащая расчетную часть, позволяющую решать точно описанный класс задач автоматического управления либо идентификации), и формирующая протокол по завершении своей работы.

где $x(t) \in \mathbb{R}^n$ — вектор состояния. Матрицы чисел A, B, M, C, D и N неизвестны. Компоненты возмущения — неизвестные, ограниченные функции:

$$|f_i(t)| \leq f_i^* \quad (i = \overline{1, \mu}), \quad (3)$$

где $f_i^* (i = \overline{1, \mu})$ — заданные числа. Выходы и входы объекта не должны превышать заданных границ:

$$|y_i(t)| \leq y_i^*, \quad |u_j(t)| \leq u_j^*, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (4)$$

где $y_i^*, u_j^* (i = \overline{1, r}, j = \overline{1, m})$ — заданные числа. Числа y_i^* таковы, что выполняется условие

$$|\bar{y}_i(t)| \leq y_i^* \quad i = \overline{1, r}, \quad (5)$$

где $\bar{y}_i(t) (i = \overline{1, r})$ — «естественные» выходы объекта (выходы в режиме его нормальной эксплуатации) когда испытательный сигнал отсутствует ($u=0$).

Элементы передаточной матрицы $W(s) = C(sE - A)^{-1}B + D$ объекта (1) по управлению имеют вид

$$w_{ij}(s) = k_{ij} s^q \frac{\prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)} s + 1) \prod_{k=1}^{\bar{p}_{ij}} (\bar{\tilde{T}}_{ij}^{(k)2} s^2 + 2\bar{\tilde{T}}_{ij}^{(k)} \bar{\xi}_{ij}^{(k)} s + 1)}{\prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)} s + 1) \prod_{k=1}^{\bar{p}_{ij}} (\bar{\tilde{T}}_{ij}^{(k)2} s^2 + 2\bar{\tilde{T}}_{ij}^{(k)} \bar{\xi}_{ij}^{(k)} s + 1)}, \quad i = \overline{1, r}, \quad j = \overline{1, m}. \quad (6)$$

Задача идентификации объекта (либо системы «объект-регулятор») состоит в нахождении оценок постоянных времени передаточных функций (6).

Для идентификации, объект (1) возбуждается испытательными сигналами

$$u_j(t) = \sum_{k=1}^{\rho} \rho_{jk} \sin \omega_{jk} (t - t_o), \quad j = \overline{1, m}, \quad (7)$$

чьи амплитуды ρ_{jk} и частоты $\omega_{jk} (j = \overline{1, m}, k = \overline{1, n})$ определяются с помощью предлагаемой директивы M011.

Директива может использоваться для планирования эксперимента по самонастройке испытательного сигнала. В этом случае неизвестные матрицы уравнения (1), возмущение формируется на основе знаний специалиста (технолога) по объекту (либо системе), и тогда модель (1) называется технологической моделью. Ее использование позволяет определить параметры алгоритма самонастройки, которые затем используются при самонастройке в реальных условиях, когда технологическая модель заменяется поступающим в директиву в реальном времени вектором выхода $y(t)$ объекта, возбуждаемого испытательным сигналом (7) и возмущениями.

3. Цели самонастройки

Цель самонастройки испытательных частот состоит в определении их нижней и верхней границ. Это означает, что частоты сигнала (7) должны удовлетворять условиям

$$\omega_{ij}^H \leq \omega_{ij} \leq \omega_{ij}^E, \quad (j = \overline{1, m}, k = \overline{1, n}), \quad (8)$$

в которых ω_{ij}^H и ω_{ij}^E — нижние и верхние границы собственных частот передаточных функций $\omega_{ij}(s)$ ($i = \overline{1, r}, j = \overline{1, m}$). Эти границы связаны постоянными времени соответствующих передаточных функций как

$$\omega_{ij}^H = \min \left\{ \frac{1}{\overline{T}_{ij}^{(k1)}}, \frac{1}{\widehat{T}_{ij}^{(k2)}}, \frac{1}{\widetilde{T}_{ij}^{(k3)}}, \frac{1}{\check{T}_{ij}^{(k4)}} \right\}, \quad \omega_{ij}^E = \max \left\{ \frac{1}{\overline{T}_{ij}^{(k1)}}, \frac{1}{\widehat{T}_{ij}^{(k2)}}, \frac{1}{\widetilde{T}_{ij}^{(k3)}}, \frac{1}{\check{T}_{ij}^{(k4)}} \right\}, \quad (9)$$

$$i = \overline{1, r}, \quad j = \overline{1, m}, \quad k_1 = \overline{1, \overline{p}_{ij}}, \quad k_2 = \overline{1, \widehat{p}_{ij}}, \quad k_3 = \overline{1, \widetilde{p}_{ij}}, \quad k_4 = \overline{1, \check{p}_{ij}}.$$

Для экспериментального определения этих границ, на вход объекта (1) подаются сигналы

$$u_j(t) = \sum_{k=1}^p \rho_{jk} \sin \omega_{jk}(t - t_0), \quad t_0 + (j-1)\tau \leq t < t_0 + j\tau, \quad j = \overline{1, m}, \quad (10)$$

где τ — заданная длительность j -го эксперимента (время фильтрации в j -ом эксперименте).

Оценки нижних границ находятся по формулам:

$$\omega_{ij}^H(\tau) = \frac{\alpha_{ij}(\tau)\omega}{\beta_{ij}(\tau)}, \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (11)$$

где

$$\alpha_{ij}(\tau) = \frac{2}{\rho_j \tau} \int_{t_0 + (j-1)\tau}^{t_0 + j\tau} y_{ji}(t) \sin \omega_k(t - t_0) dt, \quad (12)$$

$$\beta_{ij}(\tau) = \frac{2}{\rho_j \tau} \int_{t_0 + (j-1)\tau}^{t_0 + j\tau} y_{ji}(t) \cos \omega_k(t - t_0) dt, \quad i = \overline{1, r}, \quad j = \overline{1, m}$$

— выходы фильтров Фурье, а ω — заданное достаточно малое число. Оценки верхних границ находятся с помощью похожих формул, где ω — заданное достаточно большое число.

Самонастройка амплитуд $\rho_j (j = \overline{1, m})$ направлена на выполнение требований (4) к границам входов и выходов объекта.

4. Функции самонастройки

Опишем функции самонастройки в рамках ее алгоритма, предложенного в [6]. Пусть $\omega = \omega_o$ — начальная частота — заданное достаточно малое число.

С помощью MATLAB-функции TunRho (определения амплитуд испытательных гармоник) при некотором начальном $\tau = \tau_0$ находятся амплитуды $\rho_j (j = \overline{1, m})$ сигнала (10). Их поиск осуществляется последовательным увеличением либо уменьшением начального значения $\rho_j = u_j^* (j = \overline{1, m})$ с последующим его уточнением половинным делением, так, чтобы выполнялось требование (4) к границам выходов системы.

Эта функция (равно как и приведенные ниже) обращается к MATLAB-функции Analysis (решения уравнений (1) объекта при произвольных входных воздействиях), для которой MATLAB-функциями Test и Dist готовятся векторы значений $u(t)$ — испытательного сигнала и $f(t)$ — внешнего возмущения (с $\eta(t)$ — помехой измерения) на временной сетке $t = t_0 + (j-1)\tau : h : t_0 + j\tau$, где h — шаг интегрирования дифференциального уравнения (1). Последние три функции отсутствуют, когда директива используется для самонастройки в реальном времени.

Наряду с искомыми амплитудами $\rho_j (j = \overline{1, m})$, MATLAB-функция TunRho вычисляет показатели интенсивности испытательных сигналов

$$\chi_{ij} = \frac{|y_{ij}^{\max} - \bar{y}_{ij}^{\max}|}{y_{ij}^{\max}}, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (13)$$

где

$$y_{ij}^{\max} = \max_{t_0 + (j-1)\tau + \tau/2 \leq t \leq t_0 + j\tau} |y_{ij}(t)| \quad i = \overline{1, r}, \quad j = \overline{1, m}$$

$$\bar{y}_{ij}^{\max} = \max_{t_0 + (j-1)\tau + \tau/2 \leq t \leq t_0 + j\tau} |\bar{y}_{ij}(t)| \quad (14)$$

На следующем за $\tau = \tau_0$ этапе используется MATLAB-функция **FilTimFF** (определения времени фильтрации из условия ФФ-фильтруемости возмущения). Время фильтрации $\tau = \tau^*$ определяется ей из неравенств

$$\left| \frac{\bar{\alpha}_{ij}(\tau)}{\alpha_{ij}(\tau)} \right| \leq \varepsilon_{ij}^{\alpha}, \quad \left| \frac{\bar{\beta}_{ij}(\tau)}{\beta_{ij}(\tau)} \right| \leq \varepsilon_{ij}^{\beta}, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (15)$$

где $\bar{\alpha}_{ij}(\tau)$ и $\bar{\beta}_{ij}(\tau)$ — выходы фильтров Фурье (12), в которых $y_{ij}(t)$ заменено на $\bar{y}_{ij}(t)$, а $\varepsilon_{ij}^{\alpha}$ и ε_{ij}^{β} ($i = \overline{1, r}, j = \overline{1, m}$) — заданные достаточно малые числа.

После выполнения в момент времени $\tau = \tau^*$ неравенств (15), по формуле (11) вычисляются оценки нижних границ собственных частот объекта (1). Процесс фильтрации продолжается теперь уже с использованием MATLAB-функции **FilTime** (определения времени фильтрации из необходимых условий сходимости границ собственных частот по времени) до тех пор пока не выполняются эти условия:

$$\left| 1 - \frac{\omega_{ij}^H(\tau)}{\omega_{ij}^H(t - \Delta\tau)} \right| \leq \varepsilon_{ij}^\tau, \quad \tau > \tau^*, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (16)$$

в которых ε_{ij}^τ ($i = \overline{1, r}, \quad j = \overline{1, m}$) — заданные достаточно малые числа.

Затем изложенное повторяется для $\omega = \omega_0 / 2$, где находятся значения $\omega_{ij}^H(\tau^{**})$ при $\tau^{***} > \tau^{**}$, и проверяются условия

$$\left| 1 - \frac{\omega_{ij}^H(\tau^{***})}{\omega_{ij}^H(\tau^{**})} \right| \leq \varepsilon_{ij}^\omega, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (17)$$

в которых ε_{ij}^ω ($i = \overline{1, r}, \quad j = \overline{1, m}$) — заданные достаточно малые числа. Если условия (17) выполняются, то искомые оценки

$$\omega_{ij}^H = \omega_{ij}(\tau^{***}), \quad i = \overline{1, r}, \quad j = \overline{1, m}. \quad (18)$$

В противном случае, в (10) полагается $\omega = \omega_0 / 2^2$, и т. д., до тех пор, пока не выполняются неравенства (17).

Процесс нахождения оценок верхних границ собственных частот аналогичен описанному. При этом $\omega_0 = M \cdot \min_{\substack{1 \leq i \leq r \\ 1 \leq j \leq m}} \{\omega_{ij}^H\}$, где M — достаточно

большое число ($M \approx 10^3$).

Искомые испытательные частоты сигнала (7) находятся с помощью MATLAB-функции TestOm (вычисления испытательных частот). Эти частоты определяются как

$$\log \omega_{jk} = \log \omega_{ij}^H + (k - 1) \log \Delta_{ij}, \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad k = \overline{2, n}, \quad (19)$$

где $\log \Delta_{ij} = \frac{\log \omega_{ij}^6 - \log \omega_{ij}^H}{n}$ ($i = \overline{1, r}, \quad j = \overline{1, m}$).

Параметрами алгоритма самонастройки являются положительные числа: ω_0 , ε_{ij}^α , ε_{ij}^β , ε_{ij}^τ , ε_{ij}^ω ($i = \overline{1, r}, \quad j = \overline{1, m}$) и M .

5. Интерфейс

Директива M011 имеет удобный графический интерфейс, написанный с использованием MATLAB-функций, предназначенных для создания пользовательского интерфейса. Интерфейс запускается с командной строки системы MATLAB командой **DM011**. В результате, на экране появляется диалоговое окно директивы, показанное на рис. 1.

Уравнение объекта в форме Лагранжа

1q1(2)+400q1(1)+0.342q3(1)+0.94q4(1)-940q3(0)+342q4(0)=0

1q2(2)+400q2(1)+0.866q3(1)+0.5q4(1)-500q3(0)+866q4(0)=0

803q1(1)+154q2(1)+100q3(1)+754q3(0)+1130q4(0)=1u1(0)+1w1(0)

-718q1(1)-1070q2(1)+200q4(1)-867q3(0)-754q4(0)=1u2(0)+1w2(0)

Уравнения измеряемых переменных

y1=q1; y2=q2

Порядок уравнений объекта: 6 Интервал дискретности: 0.0007

Параметры внешнего возмущения

[3 2 150 0; 0 0 0 0]

Ограничения на управление: [300 300] Ограничения на выходы: [0.1 0.1]

Сохранить Загрузить Выполнить

Рис. 1. Интерфейс директивы M011.

Пользователь может ввести исходные данные с клавиатуры, сохранить их в файле для дальнейшего использования, загрузить из файла. При выборе пункта меню «Run» директива запускается на исполнение. В ходе выполнения директивы, на экран выводятся результаты промежуточных вычислений и графики переходных процессов. Результаты работы директивы сохраняются в файле.

6. Пример

Полностью управляемый и полностью наблюдаемый объект - гироскоп-латформа [7], описывается системой уравнений вида

$$P \ddot{\beta} + PS\dot{\alpha} + HC\alpha + N\beta = 0,$$

$$J\dot{\alpha} - C^T HS\alpha - (C^T H + S^T N)\dot{\beta} = Q(u + f), \quad (20)$$

где β_1 и β_2 — углы прецессии (поворота) гироскопов, α_1 и α_2 — проекции абсолютной угловой скорости площадки на ее оси, u_1 и u_2 — моменты дви-

гателей стабилизации (управления). $P = \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix}$, $Q = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 \end{pmatrix}$,

$$N = \begin{pmatrix} n_1 & 0 \\ 0 & n_2 \end{pmatrix}, \quad H = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix}, \quad J = \begin{pmatrix} j_x & 0 \\ 0 & j_y \end{pmatrix}, \quad S = \begin{pmatrix} -\sin \delta_1 & \cos \delta_1 \\ \cos \delta_2 & \sin \delta_2 \end{pmatrix},$$

$$C = \begin{pmatrix} -\cos \delta_1 & -\sin \delta_1 \\ -\sin \delta_2 & \cos \delta_2 \end{pmatrix}, \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}.$$

$p_1 = p_2 = 1 \text{ кг} \cdot \text{м}^2$, $q_1 = q_2 = 1$, $n_1 = n_2 = 4 \cdot 10^5 \text{ Н} \cdot \text{мс}$, $h_1 = h_2 = 10^3 \text{ Н} \cdot \text{мс}$, $j_x = 10^2 \text{ кг} \cdot \text{м} / \text{с}^2$, $j_y = 2 \cdot 10^2 \text{ кг} \cdot \text{м} / \text{с}^2$, $\delta_1 = -20^\circ$ и $\delta_2 = 30^\circ$.

Уравнение (20) с этими данными показано на экране интерфейса (рис. 1).

Возмущение $f_1(t) = 150 \text{sign}(\sin 2.1t)$ и $f_2(t) = 0$ и интервал дискретности $h = 0.0007 \text{ с}$.

В результате работа директива M011 по определению нижней границы собственных частот гироскопа по первому входу (u_1) и первому выходу (y_1) было получено число $\omega_{11} = 7.3$. Правильность этого числа можно проверить аналитически.

7. Заключение

В рамках системы MATLAB предложено программное обеспечение по самонастройке амплитуд и частот испытательного сигнала для моделирования процессов идентификации и адаптивного управления объектами, работающими в условиях неизвестных ограниченных внешних возмущений и помех измерений.

Разработаны MATLAB-функции, используя которые нетрудно сформировать директиву для самонастройки испытательного сигнала в реальных условиях. Для этого в директиве M011 MATLAB-функции **Analysis**, **Test** и **Dist** достаточно заменить файлами экспериментальных данных, получаемых с выхода реального объекта.

Литература

1. Льюнг Л. Идентификация систем. Теория для пользователя.— М.: Наука. 1991.— 432 с.
2. Александров А. Г. Адаптивное управление на основе идентификации частотных характеристик // Известия РАН. Теория и системы управления.— 1995.— №2.— С.63–71.
3. Граничин О. Н., Поляк Б. Т. Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах.— М.: Наука, 2003.— 291 с.
4. Дьяконов В. П., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник.— СПб.: Питер. 2002.— 448 с.
5. Александров А. Г., Орлов Ю. Ф., Михайлова Л. С. Программное обеспечение конечно-частотной идентификации и адаптивного управления многомерными объектами // Тр. II Международной конф. «Идентификация систем и задачи управления». SICPRO'03.— М.: ИПУ РАН, 2003. (CD-ROM N ISBN 5-201-14948-0).
6. Александров А. Г. Конечно-частотная идентификация: самонастройка испытательного сигнала // Труды ЭПИ МИСиС «Робастное управление

и частотная идентификация».—Электросталь: ЭПИ МИСиС, 2003.— С.67–97.

7. Александров А. Г. Синтез регуляторов многомерных систем. -М.: Машиностроение. 1986.— 272 С.

Приложение. Функции директивы M011

TimeNet — функция формирования временной сетки

Синтаксис:

t = TimeNet (number, Tdelay, Tfilter, Ndiv, omega)

Исходные данные:

Number — номер временного интервала,
Tdelay — время задержки (в периодах минимальной частоты),
Tfilter — время фильтрации (в периодах минимальной частоты),
Ndiv — число делений периода максимальной частоты,
omega — частоты ω_k ($k = \overline{1, \zeta}$) испытательного сигнала.

Результаты счета:

t — временная сетка.

Dist — функция внешних возмущений

Синтаксис:

fun = Dist (par, t)

Исходные данные:

Par — параметры внешних возмущений:
par(:,1) — типы n функций $f(t)$ внешних возмущений,
par(:,1)=0 — возмущение отсутствует: $f(t) = 0$,
par(:,1)=1 — ступенька: $f(t) = \text{par}(:,2) \cdot 1[t - \text{par}(:,3)]$,
par(:,1)=2 — гармоника: $f(t) = \text{par}(:,2) \cdot \sin[\text{par}(:,3) \cdot t + \text{par}(:,4)]$;
t — временная сетка.

Результаты счета:

fun — матрица (n строк) значений внешних возмущений на временной сетке.

Test — функция генератора испытательного сигнала

Синтаксис:

gen = Test (rho, lambda, omega, t)

Исходные данные:

rho — амплитуды ρ_k ($k = \overline{1, g}$) испытательного сигнала,
lambda — декремент λ затухания,
omega — частоты ω_k ($k = \overline{1, g}$) испытательного сигнала,
t — временная сетка.

Результаты счета:

gen — вектор значений испытательного сигнала на временной сетке.

Fourier — функция фильтра Фурье

Синтаксис:

[alf, bet] =

Fourier(rho,lambda,omega, Tdelay,Tfilter,Ndiv, y,t)

Исходные данные:

rho — амплитуды ρ_k ($k = \overline{1, \zeta}$) испытательного сигнала,
 lambda — декремент λ затухания,
 omega — частоты ω_k ($k = \overline{1, \zeta}$) испытательного сигнала,
 Tdelay — время задержки (в периодах минимальной частоты),
 Tfilter — время фильтрации (в периодах минимальной частоты),
 Ndiv — число делений периода максимальной частоты,
 y — фильтруемый сигнал,
 t — временная сетка.

Результаты счета:

alf,bet — оценки частотных параметров.

Analysis — функция моделирования при произвольных входных воздействиях

Синтаксис:

[y, x]=Analysis(A,[B M],C,[D N],[u;f],t,x0,flag)

Исходные данные:

A, [B M], C, [D N] — матрицы параметров объекта (1) в форме Коши:
 $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta$,
 [u;f] — матрица входных воздействий,
 t — временная сетка (вектор времени моделирования),
 x0 — начальный вектор состояния,
 flag — флаг построения графика:
 flag=0 — график строить, flag=1 — график не строить.

Результаты счета:

y — матрица выходных переменных,
 x — конечный вектор состояния.

TunRho — функция самонастройки амплитуд испытательного сигнала

Синтаксис:

[rho,t,x]=TunRho

(A,[B M],C,[D N], omega, p1,Ndiv, y_,u_, t,x)

Исходные данные:

A, [B M], C, [D N] — матрицы параметров объекта (1) в форме Коши:
 $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta$,
 omega — частота испытательного сигнала,
 p1 — число периодов минимальной частоты,
 Ndiv — число делений периода максимальной частоты,
 Y_ — ограничение y^* на выход объекта: $|y(t)| \leq y^*$,
 u_ — ограничение u^* на вход объекта: $|u(t)| \leq u^*$,
 t, x — начальное время и вектор состояния.

Результаты счета:

Rho — амплитуда испытательного сигнала,
 t, x — конечное время и вектор состояния.

TunOmega — функция определения границ испытательных частот

Синтаксис:

[bound, t,x] = TunOmega (A,[B M],C,[D N], omega, p1,Ndiv, gam, y_,u_, epsOm,epsT, t,x)

Исходные данные:

A, [B M], C, [D N] — матрицы параметров объекта (1) в форме Коши:
 $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta,$

omega — частота испытательного сигнала,

p1 — число периодов минимальной частоты,

Ndiv — число делений периода максимальной частоты,

gam — степень полинома при управлении (для определения нижней границы испытательных частот — gam — любое отрицательное число),

y_ — ограничение y^* на выход объекта: $|y(t)| \leq y^*$,

u_ — ограничение u^* на вход объекта: $|u(t)| \leq u^*$

t, x — начальное время и вектор состояния.

Результаты счета:

bound — нижняя граница (если gam<0) либо верхняя граница (если gam>=0) испытательных частот,

t, x — конечное время и вектор состояния.

FilTime — функция определения времени фильтрации из необходимых условий сходимости границ собственных частот по времени

Синтаксис:

[T,Dt, Tend,x]=FilTime(A,[B M],C,[D N],rho,omega,Tstep,gam,n,epsT,Tbegin,x)

Исходные данные:

A, [B M], C, [D N] — матрицы параметров объекта (1) в форме Коши:
 $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta,$

rho, omega — параметры испытательного сигнала,

Tstep — интервал дискретности,

gam — степень полинома при управлении (для определения нижней границы испытательных частот — gam — любое отрицательное число),

n — число шагов интегрирования,

epsT — $\varepsilon_{ij}^r (i = \overline{1, r}, j = \overline{1, m})$,

t, x — начальное время и вектор состояния.

Результаты счета:

T — время фильтрации,

Dt — текущие значения $\varepsilon_{ij}^r (i = \overline{1, r}, j = \overline{1, m})$,

t, x — конечное время и вектор состояния.

FilTimFF — функция определения времени фильтрации по заданным границам функций фильтруемости

Синтаксис:

[P1,Tend,x]=FilTimFF(A,[B M],C,[D N],rho,omega,p1,Ndiv,epsDa,epsDb,Tbegin,x)

Исходные данные:

A, [B M], C, [D N] — матрицы параметров объекта (1) в форме Коши:
 $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta,$

rho, omega — параметры испытательного сигнала,
p1 — число периодов минимальной частоты,
Ndiv — число делений периода максимальной частоты,
epsDa, epsDb — границы функций фильтруемости,
t, x — начальное время и вектор состояния.

Результаты счета:

P1 — время фильтрации,
t, x — конечное время и вектор состояния.

TestOm — функция вычисления испытательных частот

Синтаксис:

om = TestOm (n, omlow, omup)

Исходные данные:

n — порядок объекта (1),
omlow — нижняя граница испытательных частот,
omup — верхняя граница испытательных частот.

Результаты счета:

om — набор (n-вектор) испытательных частот.

УДК 519.6: 531.7.08

ВИРТУАЛЬНЫЕ ИСПЫТАНИЯ ДВУХКООРДИНАТНОГО СЕЙСМОПРЕОБРАЗОВАТЕЛЯ В СРЕДЕ MATLAB

Алексеев К. А., Исянов Р. Н.

Пензенский государственный университет, Пенза,

e-mail: ruslan@zato.ru

На сегодняшний день технические средства охраны достаточно разнообразны и в целом эффективны. Однако практически всем им присущ один существенный недостаток: они вырабатывают сигнал вторжения лишь после проникновения злоумышленника на территорию охраняемого объекта. Простым примером тому служат системы охранного видеонаблюдения. Эти системы, в большинстве своем, при помощи видеорегистраторов способны лишь подтвердить факт вторжения после того, как он уже произошел. Зачастую злоумышленник рассчитывает на временную задержку, которая проходит с момента проникновения на объект до момента срабатывания сигнализации. Основным фактором, определяющим эффективность любой охранной системы, является минимизация указанного интервала времени, и в этом смысле привлекательность периметральных систем охраны неоспорима.

Для обнаружения нарушения территориальности объекта (дома, склада, коттеджа) используются различные оповещатели. Окна, двери защищают контактными оповещателями, обычно герконами, которые показывают положение дверных и оконных створок, форточек. Дополнительно используются емкостные оповещатели, например, мобильные, устанавливаемые на ручки дверей или окон. Для защиты окон и дверей используются также оптикоэлектронные оповещатели, оповещатели разбития стекла. Для защиты дверей и стен от пролома используют вибрационные, сейсмические и микрофонные оповещатели.

Периметральная граница объекта является наилучшим местом для раннего обнаружения вторжения. Нарушитель, взаимодействуя, в первую очередь, с физическим периметром, создает возмущения, которые можно зарегистрировать специальными извещателями: будь то ограждение в виде металлической решетки (ее надо сперва перерезать или перелезть через нее), стена или барьер (их придется преодолевать сверху; если это стена или крыша здания, — их нужно разрушить). Открытую территорию необходимо пересечь. Итак, нарушитель, вызывая физический контакт с периметром, создает возможность электронными средствами обнаружить свое вторжение на первом рубеже охраны, т. е. на периметре.

Таким образом, системы периметровой охраны являются наиболее эффективными средствами защиты от несанкционированного проникнове-

ния, поскольку выдают сигнал тревоги задолго до того, как злоумышленник может проникнуть в особо важные зоны охраняемого объекта.

Первоначально в таких системах использовались однокоординатные сейсмоприемники, позволяющие определять наличие нарушителя в охраняемом периметре. Однако со временем простого детектирования нарушителя стало недостаточно, появилась необходимость в определении текущей координаты нарушителя, а также в вычислении траектории его движения (сегодня данные задачи являются главными для сейсмолокации)

Традиционным направлением решения этой задачи является использование групп однокоординатных сейсмоприемников, расположенных в контролируемой зоне в узлах квадратной или прямоугольной сетки. Оценка координат нарушителя осуществляется на основе данных о времени прихода поперечной волны сейсмозвмущения к различным сейсмоприемникам. Другим направлением решения задачи сейсмолокации является создание и внедрение двухкоординатных сейсмоприемников, работающих на продольных и поперечных поверхностных волнах. Главным достоинством данного направления является простота вычисления азимута источника возмущения и возможность исключения влияния негоризонтальности заделки сейсмопреобразователя в грунте.

Предлагаемая конструкция двухкоординатного резонансного сейсмоприемника имеет два ортогонально расположенных в горизонтальной плоскости чувствительных элемента, каждый из которых характеризуется главными осями чувствительности (OX и OY). Коэффициенты преобразования чувствительных элементов в сеймоприемнике зависят от угла α между осью чувствительности и азимутом источника сейсмозвмущения, а также от угла β наклона плоскости основания относительно горизонтальной плоскости:

$$K_y = K_1 \cdot \sin \alpha$$

$$K_x = K_2 \cdot \cos \alpha,$$

где K_1, K_2 — коэффициенты преобразования датчика при совпадении оси максимальной чувствительности сейсмоприемника по осям Y и X с направлением сейсмоизлучения; α — главный угол ($0 \leq \alpha \leq \frac{\pi}{2}$), определяемый от оси OX до оси, соединяющей сейсмоприемник с источником возмущения (см. рис. 1).

При использовании сейсмоприемников, настроенных на некоторую эффективную частоту ω возмущения, в момент воздействия входного сигнала на выходе преобразователя формируется два сигнала:

$$U_y = V_R \cdot K \sin \alpha \cdot \sin \omega_1 t$$

$$U_x = V_R \cdot K \cos \alpha \cdot \sin \omega_2 t,$$

где $\omega_1 \approx \omega_2$ — резонансные частоты двухкоординатного сейсмоприемника по осям Y и X , V_R — амплитуда скорости сейсмоволны, зависящая от расстояния R до источника сейсмозмущения.

Отметим, при конструировании сейсмопреобразователей имеет смысл обеспечить выполнение следующих условий: $\omega_1 \approx \omega_2$ и $K_1 \approx K_2$.

Задача сейсмолокации на продольных поверхностных волнах с использованием двухкоординатного сейсмоприемника связана с нахождением главного угла α , который может быть определен из выражения:

$$\alpha = \arctg \frac{U_y}{U_x}.$$

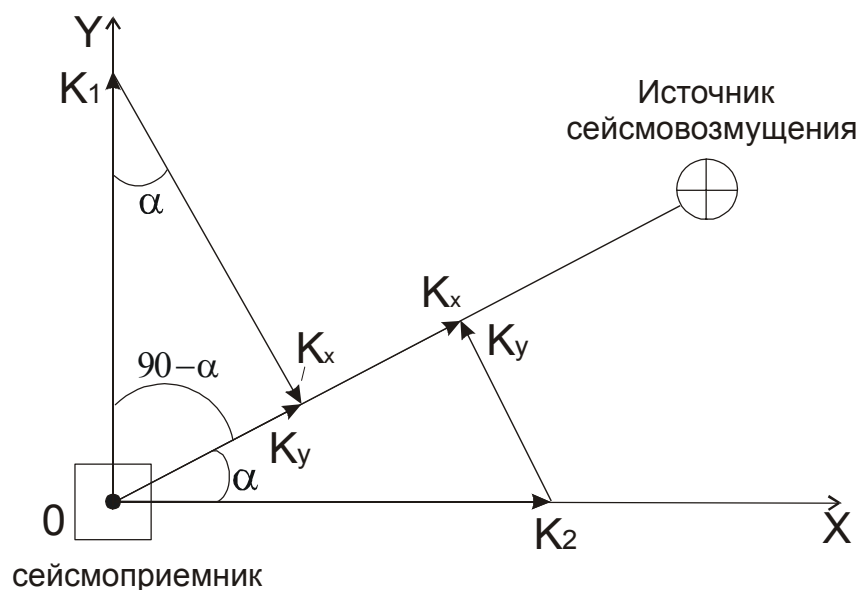


Рис. 1. Определение коэффициентов преобразования по осям X и Y двухкоординатного сейсмоприемника.

Известно, что в процессе эксплуатации сейсмоприемника и/или групп сейсмоприемников угол горизонтальной заделки в грунт не остается постоянным, а изменяется в широких пределах. Данная причина вызвана климатическими воздействиями на грунт (неравномерным замерзанием и оттаиванием грунта), эрозией почвы, сейсмической активностью земной коры и т. д. Следствием изменения угла горизонтальной заделки в грунт является изменение выходного сигнала сейсмопреобразователя.

В докладе приводятся результаты имитационного моделирования двухкоординатного сейсмопреобразователя в среде MATLAB. В качестве математической модели использована функция преобразования сейсмопреобразователя, послужившая основой для исследования диаграмм направленности преобразователя в случае абсолютной горизонтальности заделки его основания в грунт, а также случаев, когда угол горизонтальной заделки варьируется в пределах от 0 до 90° (см. рис. 2–5).

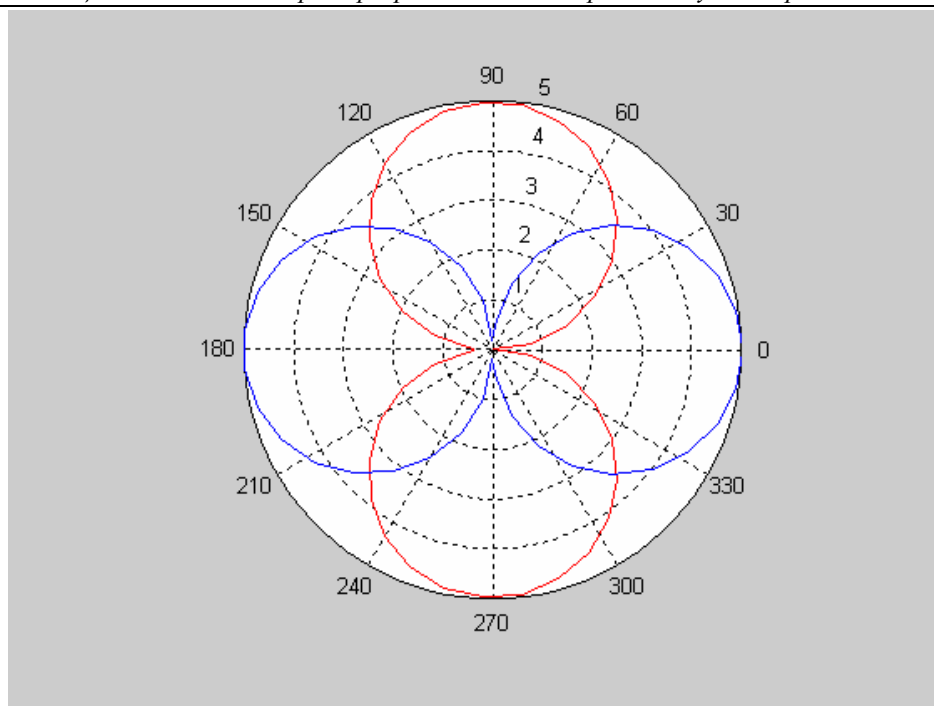


Рис. 2 Диаграмма направленности при $\beta = 0$.

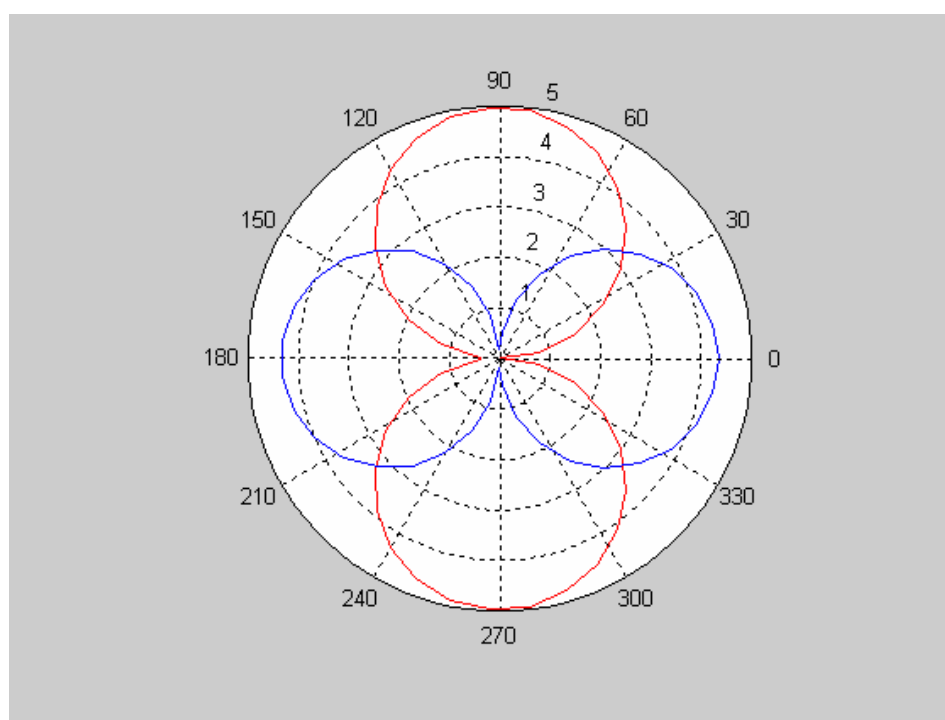


Рис. 3. Диаграмма направленности при $\beta = 30$.

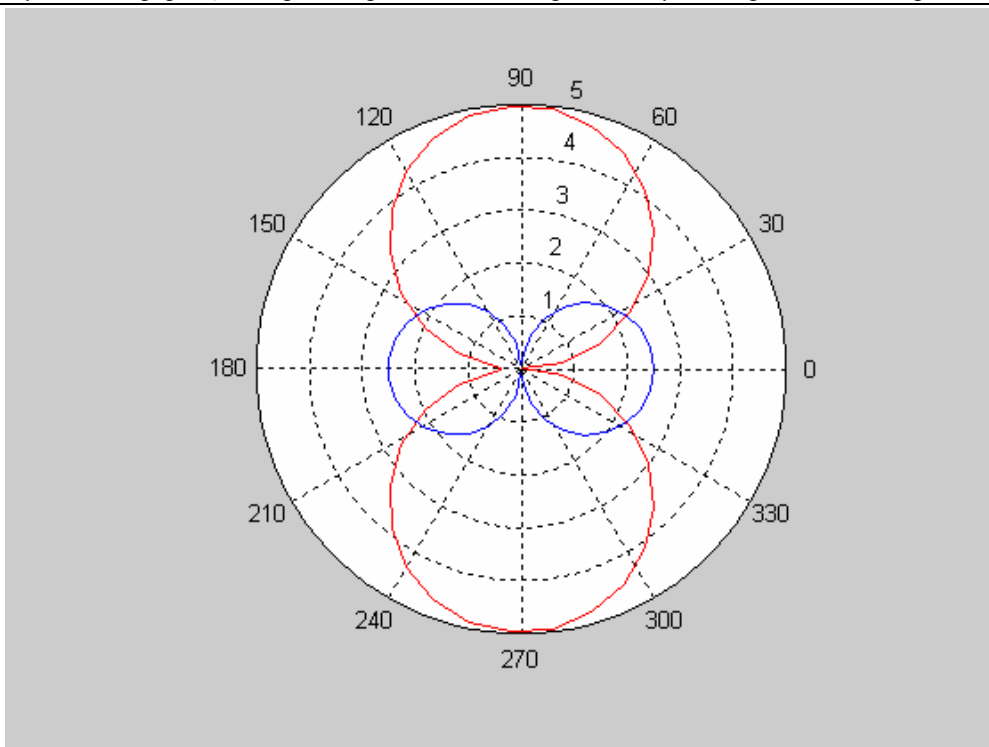


Рис. 4. Диаграмма направленности при $\beta = 60^\circ$.

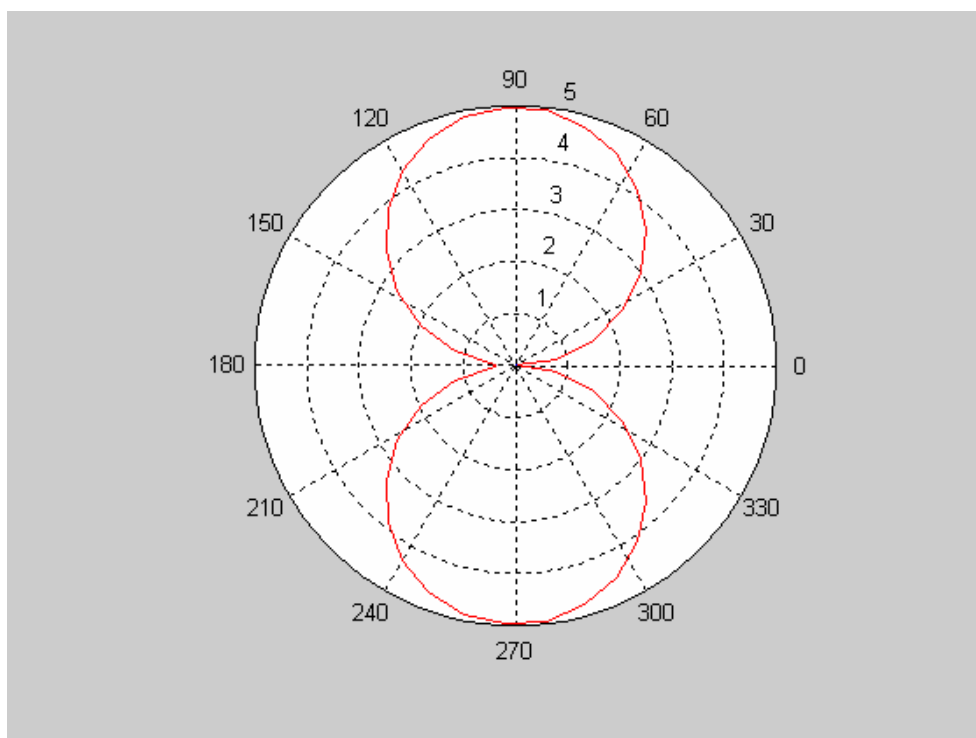


Рис. 5. Диаграмма направленности при $\beta = 90^\circ$.

Судя по полученным результатам, восстановление сигнала в интервале от 0 до 30° возможно с погрешностью, не превышающей 10%. Однако при натурных испытаниях, проводимых на полигоне, аналогичный резуль-

тат получен не был.

Причину несовпадения результатов виртуальных и натурных испытаний можно объяснить следующим образом: грунт при воздействии на него механическим способом совершает колебания по эллиптическим орбитам, причем вертикальная составляющая колебаний в 2–2,5 раза превышает горизонтальную. С учетом новых представлений о геофизике распространения сейсмоволн в грунтах была получена новая математическая модель и проведены виртуальные испытания в MATLAB. На рисунках 6–9 представлены диаграммы направленности, построенные для двухкоординатного сейсмопреобразователя с учетом вертикальной составляющей сейсмосигнала.

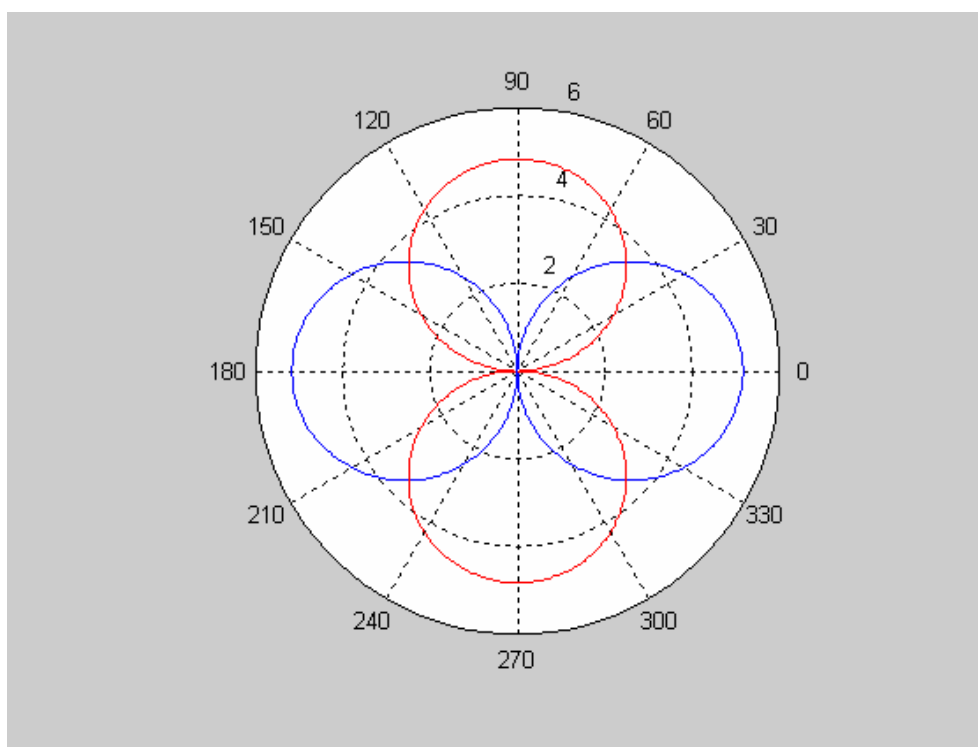


Рис. 6. Диаграмма направленности, полученная с учетом вертикальной составляющей сейсмосигнала ($\beta = 0$)

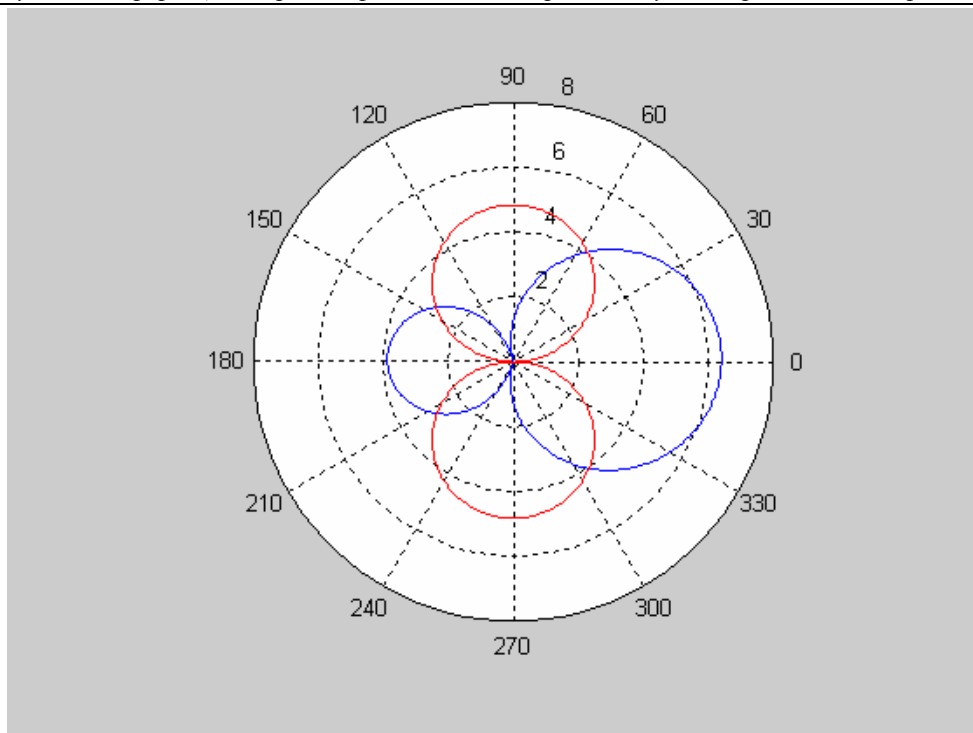


Рис. 7. Диаграмма направленности, полученная с учетом вертикальной составляющей сейсмосигнала ($\beta = 10$)

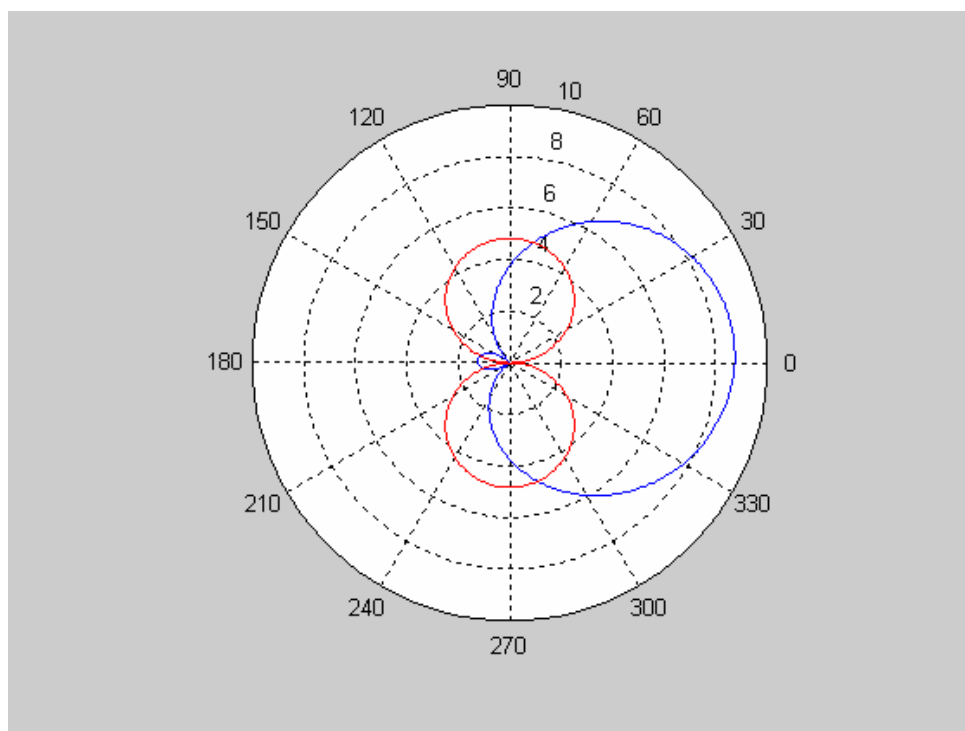


Рис. 8. Диаграмма направленности, полученная с учетом вертикальной составляющей сейсмосигнала ($\beta = 15$).

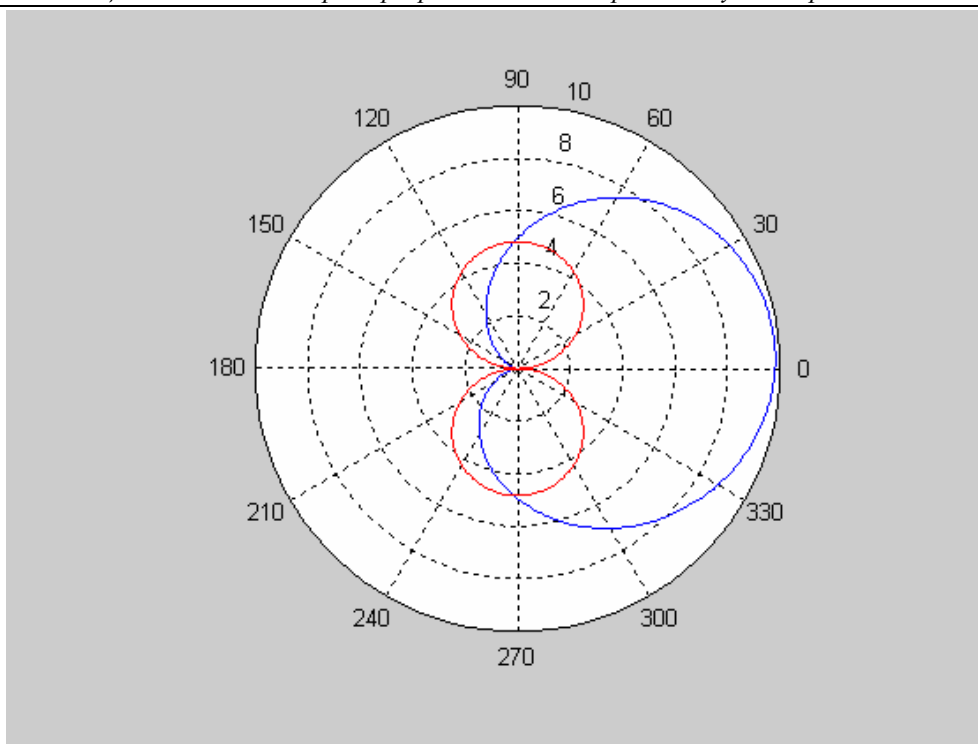


Рис. 9. Диаграмма направленности, полученная с учетом вертикальной составляющей сейсмосигнала ($\beta = 20$).

Итак, в результате моделирования стало ясным, что при наклоне основания сейсмопреобразователя на угол $\beta = 20$ относительно горизонтали сигнал по одному из каналов становится сильно искаженным. Данный угол можно считать критическим для определения азимута нарушителя: результат такого определения содержит значительную погрешность.

Синергетические свойства математических пакетов и в том числе среды MATLAB позволили значительно упростить решение задачи испытания разработанного двухкоординатного датчика, исследовать причины возникновения различного рода погрешностей, возникающих при работе преобразователя в грунте, а также предложить новые пути решения проблемы определения направления движения нарушителя при использовании систем периметровой охраны.

По результатам проведения виртуальных экспериментов предложено с целью устранения влияния изменения угла горизонтальной заделки основания сейсмопреобразователя на определение азимута нарушителя включить в конструкцию сейсмопреобразователя третью ортогональную координату, фиксирующую вертикальную составляющую сейсмических волн.

УДК 621.313.333

МОДЕЛИРОВАНИЕ ТЕПЛОВОГО СОСТОЯНИЯ АСИНХРОННЫХ ДВИГАТЕЛЕЙ ПРИ ПОНИЖЕННОМ НАПРЯЖЕНИИ В РЕЖИМАХ S1, S2, S3, S4

*Алексеев Е. Р., Федоров М. М., Чеснокова О. В., Литвинова Т.
Донецкий национальный технический университет, Донецк, Украина,
e-mail: aer@teacher.dn-ua.com*

Эксплуатация асинхронных двигателей (АД) зачастую сопровождается различного рода аномалиями. Одной из наиболее распространенных является нестабильное напряжение сети. Отклонения напряжения могут достигать 15 и более процентов, что приводит к росту тока в обмотках асинхронного двигателя, к увеличению потерь, росту температуры обмоток АД и в итоге — к досрочному выходу его из строя. В связи с этим задача исследования влияния напряжения на тепловые состояния представляется достаточно актуальной.

Одним из основных методов исследований тепловых процессов в асинхронных двигателях (АД) является метод эквивалентных тепловых схем замещения (ЭТС) [1]. Система дифференциальных уравнений, описывающая тепловые процессы в АД, имеет вид:

$$C \frac{d\theta}{dt} + \Lambda \theta = P \quad (1)$$

Здесь C — вектор-столбец теплоемкостей узлов двигателей, θ — вектор-столбец превышения температур соответствующих узлов над температурой окружающей среды θ_0 , Λ — матрица теплопроводностей узлов, P — вектор потерь мощностей в узлах АД. Методика расчета потерь при различных напряжениях и нагрузках изложена в [2]. В общем случае потери разделяют на постоянные $p_{пост}$, переменные $p_{пер}$ (в обмотках статора и ротора), добавочные $p_{доб}$. При изменении напряжения постоянные потери пропорциональны квадрату напряжения [2,3]

$$p_{пост} = p_{постн} \left(\frac{U_0}{U_{0н}} \right)^2,$$

где $p_{постн}$ — постоянные потери при номинальном напряжении, номинальной нагрузке. Переменные потери $p_{пер}$ находятся в функциональной зависимости от квадрата тока и температуры [2,3] $p_{пер} = p_{перн} (\beta_I^2 (1 + \alpha \theta))$, где $p_{перн}$ — переменные потери при номинальном напряжении, номинальной нагрузке и температуре окружающей среды, β_I — кратность тока, α —

температурный коэффициент материала обмотки, θ — разность между температурой обмотки и температурой окружающей среды. Добавочные потери пропорциональны квадрату тока [2, 3]

$$P_{доб} = P_{добн} \beta_I^2,$$

где $P_{добн}$ — добавочные потери при номинальном напряжении и номинальной нагрузке.

С помощью MATLAB проведено исследование теплового состояния АД МТН111-6 и АИУМ 225М4 в режимах S1, S2, S3 и S4.

1. Тепловое состояние АД в продолжительном режиме S1

Режим S1 характеризуется продолжительной работой асинхронного двигателя при постоянной нагрузке. При расчете превышения температур в установившемся состоянии используется система (1) с учетом, что $\frac{d\theta_i}{dt} = 0$. В этом случае тепловое состояние АД описывается системой линейных алгебраических уравнений

$$\Lambda \theta = P. \quad (2)$$

С помощью MATLAB были получены значения температур при различных значениях коэффициента нагрузки и напряжения сети. На рис. 1 представлен график зависимости температуры наиболее нагретого узла двигателя (лобовой части обмотки статора) АИУМ225М4

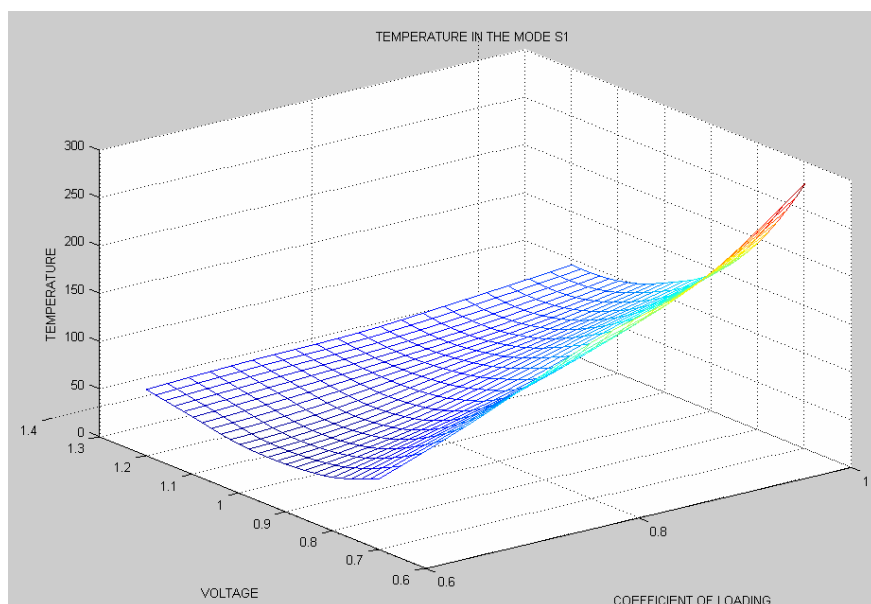


Рис. 1. Температура лобовой части обмотки статора при различных значениях напряжения и коэффициента нагрузки

В табл. 1 представлены значения температур узлов (1 — пазовая часть обмотки статора; 2 — пазовая часть обмотки ротора; 3 — пакет (же-

лезо) статора; 4 — пакет (железо) ротора; 5 — лобовая часть обмотки статора; 6 — лобовая часть обмотки ротора; 7 — внутренний воздух; 8 — корпус) АД МТН111-6 в абсолютных и относительных единицах. За базовую принята температура узла при номинальных напряжении и коэффициенте нагрузки ($\beta=1$).

Таблица 1.
Значения перегрева узлов АД МТН111-6.

β	U	Перегрев $\theta_{уст} / \theta_{уст}^*$ узлов							
		Номер узла							
		1	2	3	4	5	6	7	8
1.2	$U_{1н}$	122/1.24	142/1.33	96/1.25	135/1.31	143/1.25	142/1.33	128/1.29	69/1.25
	$0.9U_{1н}$	130/1.33	155/1.45	102/1.36	149/1.45	155/1.36	151/1.41	146/1.47	75/1.46
	$0.8U_{1н}$	162/1.65	216/2.02	127/1.65	203/1.97	196/1.72	216/2.02	187/1.89	94/1.71
1	$U_{1н}$	98/1	107/1	77/1	103/1	114/1	107/1	99/1	55/1
	$0.9U_{1н}$	100/1.02	114/1.07	79/1.03	110/1.07	118/1.04	112/1.05	108/1.09	57/1.04
	$0.8U_{1н}$	112/1.14	143/1.33	88/1.14	135/1.31	134/1.18	143/1.34	125/1.26	64/1.16
0.9	$U_{1н}$	87/0.89	92/0.86	69/0.9	89/0.86	101/0.89	92/0.86	86/0.87	49/0.89
	$0.9U_{1н}$	84/0.86	93/0.87	66/0.86	90/0.87	98/0.86	91/0.85	89/0.9	48/0.87
	$0.8U_{1н}$	96/0.98	121/1.07	76/0.99	114/1.11	114/1	121/1.13	107/1.08	56/1.02
0.8	$U_{1н}$	83/0.85	85/0.79	65/0.84	82/0.8	95/0.83	86/0.8	81/0.82	47/0.85
	$0.9U_{1н}$	74/0.76	80/0.75	59/0.77	78/0.76	86/0.75	79/0.74	77/0.78	42/0.76
	$0.8U_{1н}$	82/0.84	100/0.93	65/0.84	95/0.92	97/0.87	100/0.93	90/0.91	47/0.85

В результате моделирования теплового состояния АД при различных напряжениях и коэффициентах нагрузки было установлено, как должна изменяться нагрузка при изменении напряжения, чтобы превышение температуры соответствующего узла не превышало допустимого значения. Таким образом, были определены коэффициенты безаварийной работы при пониженных напряжениях (см. табл. 2, 3).

Таблица 2.
Значения коэффициентов нагрузки β АД МТН111-6 при условии, что температуры пазовой части обмотки ротора θ_5 и лобовой части обмотки статора θ_2 не превышают допустимого значения.

		$U_1 = 0.8U_{1н}$	$U_1 = 0.9U_{1н}$	$U_1 = U_{1н}$
Коэффициент нагрузки β	$\theta_5 = \theta_{дон}$	0.88	0.98	1
	$\theta_2 = \theta_{дон}$	0.867	0.972	1

Таблица 3.
Значения коэффициентов нагрузки β при условии, что температура пазовой и лобовой обмоток статора не превышает допустимого значения

		$U = 0.8U_{1н}$	$U = 0.9U_{1н}$	$U = U_{1н}$
Коэффициент нагрузки β	$\theta_1 = \theta_{дон}$	0.799	0.905	1
	$\theta_2 = \theta_{дон}$	0.79	0.899	1

Средствами MATLAB были исследованы значения температуры в узлах АД при изменении напряжений и постоянном токе статора ($I_1 = const$), который является основным контролируемым параметром в процессе эксплуатации АД. В табл. 4 и 5 приведены расчетные и экспериментальные значения температур узлов АД МТН111-6 и АИУМ225М4 при $I_1 = const$.

Таблица 4.

Значения превышения температуры узлов АД МТН111-6 при номинальном значении тока статора.

Узлы АД	$U_{1н}$		$0.9U_{1н}$		$0.8U_{1н}$	
	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$
Лобовая часть обмотки статора (расчет)	114	1	135	1.18	143	1.22
Лобовая часть обмотки статора (эксперимент)	115	1	131	1.16	140	1.24
Пазовая часть обмотки ротора (расчет)	107	1	133	1.24	143	1.34

Таблица 5.

Значения перегрева узлов АД АИУМ225М4 при номинальном значении тока статора.

Узлы АД	$U_{1н}$		$0.9U_{1н}$		$0.8U_{1н}$	
	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$	$\theta_{уст},$ $^{\circ}C$	$\frac{\theta_{уст}}{\theta_{устн}}$
Пазовая часть обмотки статора	94	1	112	1.19	139	1.48
Лобовая часть обмотки статора	110	1	133	1.21	167	1.52
Ротор	118	1	144	1.22	183	1.55

Было выявлено, что постоянство тока статора не может служить критерием безаварийной работы АД при пониженном напряжении.

В результате моделирования с помощью MATLAB теплового состояния АД в режиме S1 были определены безопасные режимы эксплуатации АД МТН111-6 и АИУМ225М4 при пониженном напряжении.

2. Тепловое состояние АД в кратковременном режиме S2

Режим S2 характеризуется работой при постоянной нагрузке в течение короткого периода времени ($t_p = 10, 30, 60$ и 90 мин), сопровождаемого остановкой; тепловой режим достигает максимума перед остановкой.

Для моделирования теплового состояния в режиме S2 решалась система дифференциальных уравнений с постоянными коэффициентами (1).

При решении системы в MATLAB использовалась функция **ode23t**, позволяющая получить достаточно точное решение. При моделировании теплового состояния предварительно пересчитывались значения вектора правых частей **P**. Это связано с тем, что в режиме S2 двигатель работает с другим номинальным током статора. Так, в двигателе АИУМ225М4 при $t_p=60$ мин номинальный ток статора на 25% выше, чем в режиме S1. В результате моделирования теплового состояния были получены значения температур АД при различных значениях напряжения и коэффициента нагрузки. На рис. 2 приведены зависимости температуры наиболее нагретого узла (лобовой части обмотки статора) АД АИУМ225М4 от напряжения при различных коэффициентах нагрузки в абсолютных и относительных единицах. За базовую принята температура в режиме S1 при номинальных напряжении и коэффициенте нагрузки(β).

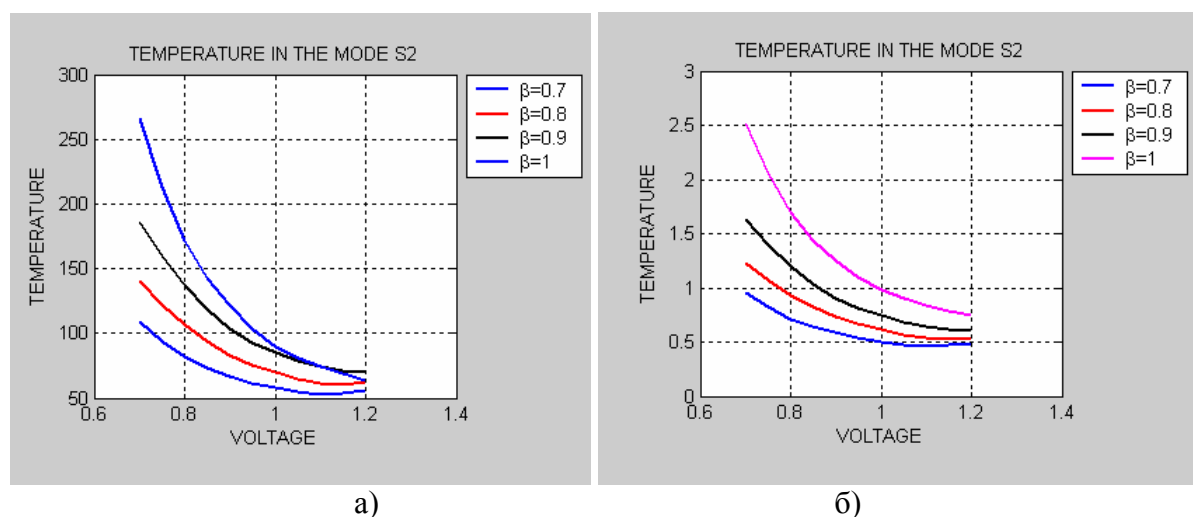


Рис. 2. Зависимость температуры от напряжения при различных коэффициентах нагрузки (а- абсолютные единицы, б- относительные)

Аналогично с режимом S1 были найдены коэффициенты безаварийной работы при пониженных напряжениях. На рис. 3 приведены зависимости $\beta(U_1)$ АД АИУМ225М4 при условии постоянства температуры в режимах S1 и S2.

3. Тепловое состояние АД в повторно-кратковременных режимах

В повторно-кратковременном режиме S3 цикл работы ($t_{\text{ц}}$) равен сумме рабочего периода (t_p) и паузы ($t_{\text{ц}}$) [4]. Повторно-кратковременный режим характеризуется продолжительностью включения $\text{ПВ} = \frac{t_p}{t_{\text{ц}}} \cdot 100\%$.

Согласно ГОСТ $t_{\text{ц}} = 10$ мин.

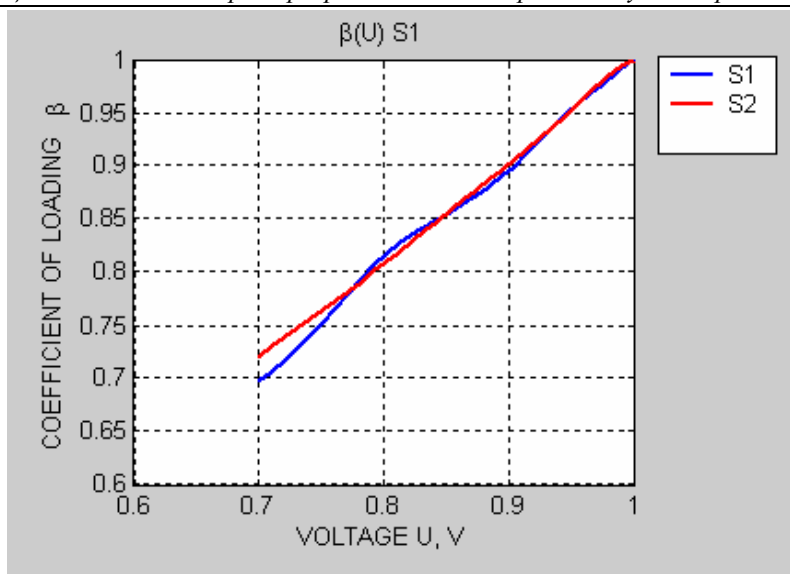


Рис. 3. Зависимость $\beta(U_1)$ АД АИУМ225М4 при условии постоянства температуры.

При моделировании теплового состояния в режиме S3 необходимо решать систему дифференциальных уравнений (1) с вектором потерь \mathbf{P} на рабочем интервале $[0, t_p]$ и при $\mathbf{P}=0$ в период паузы. Следует учесть, что матрица теплопроводностей узлов $\mathbf{\Lambda}$ различается на интервалах $[0, t_p]$ и $[t_p, t_{\text{ц}}]$. После определенного числа включений наступает квазиустановившееся состояние — такое состояние, когда закон изменения температуры $\theta(t)$ в каждом последующем цикле повторяется. Критерием достижения квазиустановившегося состояния являются величины максимального и минимального значения температуры, изменение которых не превышает 2°C в течение часа [4]. Основными характеристиками каждого цикла и квазиустановившегося состояния являются максимальное θ_{\max} , минимальное θ_{\min} , среднее θ_{cp} значение превышения температуры (см. рис. 4) и размах изменения температуры $\Delta\theta = \theta_{\max} - \theta_{\min}$.

В табл. 6, 7 представлены расчетные превышения температур θ_{\min} , θ_{\max} , θ_{cp} лобовых частей статора и ротора АД МТН11-6 и лобовой части обмотки статора короткозамкнутого АД АИУМ225М4 в квазиустановившемся состоянии в режиме S3 ($t_{\text{ц}} = 10 \text{ мин}$, ПВ=40%), и превышения температуры в режиме S1 при напряжениях $U_1 = U_{1н}$, $U_1 = 0.9U_{1н}$ и $U_1 = 0.8U_{1н}$ в абсолютных и относительных единицах ($\theta_i^* = \frac{\theta_i}{\theta_{in}}$). В качестве базового

значения превышения температуры узла принято превышение температуры узла в режиме S1 при $U_1 = U_{1н}$.

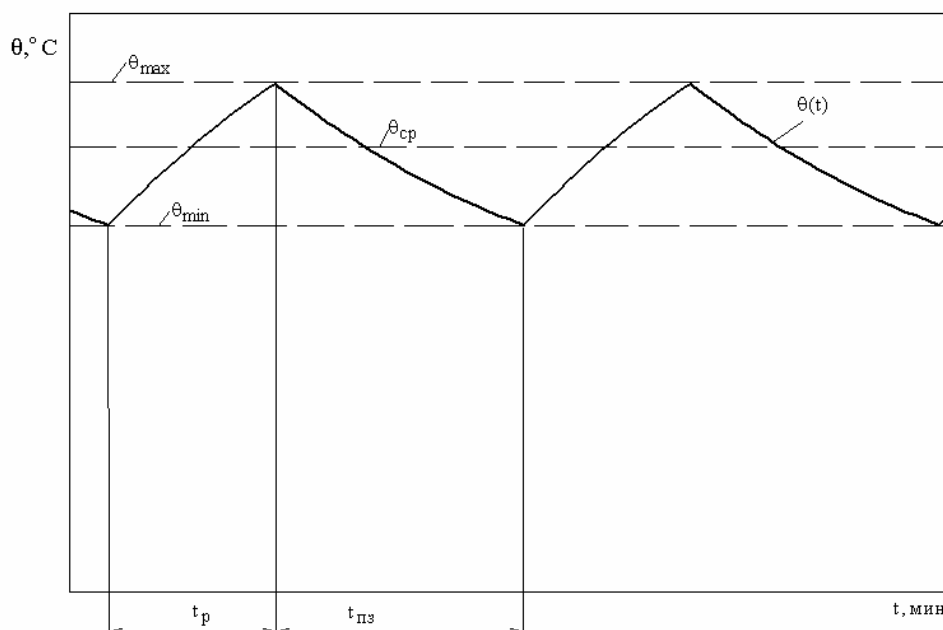


Рис. 4. Квазиустановившееся состояние в повторно-кратковременном режиме S3 и ПВ=40%.

Таблица 6.

Превышения температуры узлов АД МТН111-6 в режимах S3 и S1 при различных напряжениях в абсолютных и относительных единицах

Превышения температуры узлов АД МТН111-6

№ узла	$U_1 = U_{1н}$				$U_1 = 0.9U_{1н}$				$U_1 = 0.8U_{1н}$			
	θ_{max}	θ_{min}	$\theta_{ср}$	$\theta_{уст}$	θ_{max}	θ_{min}	$\theta_{ср}$	$\theta_{уст}$	θ_{max}	θ_{min}	$\theta_{ср}$	$\theta_{уст}$
5	95	63	79	114	107	71	89	118	135	90	112	134
	0.83	0.55	0.69	1	0.94	0.62	0.78	1.04	1.18	0.79	0.99	1.18
6	104	86	95	107	128	102	114	112	167	134	151	143
	0.97	0.8	0.89	1	1.18	0.95	1.07	1.05	1.56	1.25	1.41	1.34

Таблица 7.

Превышение температуры лобовой части обмотки статора АД АИУМ225М4 θ_{min} и θ_{max} в режимах S3 и S1

$U_1 = U_{1н}$			$U_1 = 0.9U_{1н}$			$U_1 = 0.8U_{1н}$		
θ_{max}	θ_{min}	$\theta_{уст}$	θ_{max}	θ_{min}	$\theta_{уст}$	θ_{max}	θ_{min}	$\theta_{уст}$
107/0.98	78/0.72	109/1	136/1.25	99/0.91	133/1.22	221/2.03	159/1.46	168/1.54

Температура узлов АД в режиме S3 при номинальном напряжении несколько ниже температуры в режиме S1. В результате моделирования теплового состояния АД в системе MATLAB выявлено, что темпы роста температуры при снижении напряжения в короткозамкнутом АД АИУМ225М4 в режиме S3 выше, чем в АД с фазным ротором МТН111-6.

Из проведенного расчета следует, что, как и в режиме S1, при снижении входного напряжения во избежание выхода двигателя из строя необходимо снижать нагрузку на валу. Выявлено (табл. 8 - для АД МТН111-6, табл. 9 — АД АИУМ225М4), как должна изменяться нагрузка в режиме S3 при изменении напряжения, чтобы температура лобовой части обмотки статора оставалась постоянной.

Таблица 8.

Значения нагрузки в режимах S3 АД МТН111-6 при напряжениях $U_1 = 0.9U_{1н}$,

$U_1 = 0.8U_{1н}$ при постоянной температуре лобовой обмотки статора

№ узла	$U_1=0.9U_{1н}$				$U_1=0.8U_{1н}$			
	β	θ_{max}	θ_{min}	θ_{cp}	β	θ_{max}	θ_{min}	θ_{cp}
5	0.94	94	63	79	0.83	94	63	79
6		111	90	100		114	93	103

Таблица 9.

Значения нагрузки в режимах S3 и S1 АД АИУМ225М4 при напряжениях $U_1 = 0.9U_{1н}$,

$U_1 = 0.8U_{1н}$ и постоянной температуре лобовой обмотки статора

$U_1 = 0.9U_{1н}$					$U_1 = 0.8U_{1н}$				
S3				S1	S3				S1
β	θ_{max}	θ_{min}	θ_{cp}	β $\theta_{уст}$	β	θ_{max}	θ_{min}	θ_{cp}	β $\theta_{уст}$
0.9	109	79	94	0.9 109	0.78	105	76	91	0.8 109

Особенностью повторно-кратковременного режима с частыми пусками S4 является наличие периода пуска соизмеримого времени цикла. В режиме S4 цикл работы $t_{ц}$ равен сумме периода пуска $t_{п}$, рабочего периода $t_{р}$ и паузы $t_{пз}$, и характеризуется продолжительностью включения

$$ПВ = \frac{t_{п} + t_{р}}{t_{ц}} \cdot 100\%. \text{ Длительность цикла в режиме S4 составляет 0.5, 1 и 2}$$

мин. В режиме S4, как и в режиме S3, наибольшие тепловые нагрузки АД испытывает в квазиустановившемся состоянии. Особенностью моделирования теплового состояния АД является учет периода пуска. На этом этапе необходимо решать систему дифференциальных уравнений (1) с вектором потерь \mathbf{P} , соответствующих пусковому току статора.

В табл. 10 приведено среднее превышение температуры лобовой части статора в квазиустановившемся состоянии при различных напряжениях в режимах S3, S4 ($t_{ц} = 2$ мин, ПВ=40%). В этой же таблице для сравнения приведено установившееся превышение температуры в режиме S1 АД АИУМ225М4 в абсолютных и относительных единицах. За базовое превышение температуры узла принято превышение температуры в режиме S1 при номинальных значениях напряжения и нагрузки.

Таблица 10.

Превышение температуры лобовой части обмотки статора АД АИУМ225М4 при различных напряжениях в режимах S1, S3 и S4 ($t_{\text{ц}} = 10 \text{ мин}$, ПВ=40%) в абсолютных (θ) и относительных (θ^*) единицах.

$U_1 = U_{1\text{н}}$			$U_1 = 0.9U_{1\text{н}}$			$U_1 = 0.8U_{1\text{н}}$		
S1	S3	S4	S1	S3	S4	S1	S3	S4
$\frac{\theta_{\text{уст}}}{\theta_{\text{уст}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$	$\frac{\theta_{\text{уст}}}{\theta_{\text{уст}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$	$\frac{\theta_{\text{уст}}}{\theta_{\text{уст}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$	$\frac{\theta_{\text{ср}}}{\theta_{\text{ср}}^*}$
$\frac{109}{1}$	$\frac{90}{0.83}$	$\frac{103}{0.94}$	$\frac{133}{1.22}$	$\frac{114}{1.05}$	$\frac{129}{1.18}$	$\frac{168}{1.54}$	$\frac{184}{1.69}$	$\frac{201}{1.84}$

Как видно из табл. 10, при $U_1 = U_{1\text{н}}$ и $U_1 = 0.9U_{1\text{н}}$ наибольшие тепловые нагрузки обмотки статора короткозамкнутого двигателя испытывают в режиме S1. При снижении напряжения на 20% наибольшие тепловые нагрузки АД наблюдаются в режиме S4.

В табл. 11 приведены превышения температуры лобовой части обмотки статора при напряжениях $U_1 = U_{1\text{н}}$, $U_1 = 0.9U_{1\text{н}}$, $U_1 = 0.8U_{1\text{н}}$ в режиме S4 (ПВ=40%) при различных длительностях цикла ($t_{\text{ц}}=0.5, 1, 2 \text{ мин}$) в абсолютных и относительных единицах.

Таблица 11.

Превышения температуры лобовой части обмотки статора (θ/θ^*) АД АИУМ225М4 при различных напряжениях в режиме S4 (ПВ=40%).

$t_{\text{ц}}, \text{мин}$	$U_{1\text{н}}$		$0.9U_{1\text{н}}$		$0.8U_{1\text{н}}$	
	θ/θ^*	$t_{\text{п}}^*, \%$	θ/θ^*	$t_{\text{п}}^*, \%$	θ/θ^*	$t_{\text{п}}^*, \%$
2	103/0.94	0.19	129/1.18	0.25	201/1.84	0.35
1	117/1.07	0.38	141/1.29	0.5	215/1.9	0.7
0.5	138/1.27	0.76	169/1.55	1	247/2.27	1.4

При уменьшении $t_{\text{ц}}$ с 2 до 0.5 мин превышение температуры лобовой части обмотки статора при номинальном напряжении увеличивается на 33% и достигает 138°C, что выше допустимого.

4. Сравнительный анализ теплового состояния различных АД

На последнем этапе исследования полученные результаты обобщены на широкий класс АД. Выявлено, что различные количественные значения рабочих характеристик (в первую очередь токов и потерь) являются следствием различных степеней насыщения магнитной системы двигателя, важнейшим показателем которой является ток холостого хода. В связи с тем, что токи и потери являются исходными данными для расчета значений температур узлов АД, отличие величин относительных температур

также можно объяснить различными значениями токов холостого хода. Был проведен анализ различных общепромышленных АД серии 4А, рассчитанных на напряжение до 1000В. В табл. 12 представлены значения относительных температур наиболее нагретых узлов АД (лобовые части обмоток статора θ_5 и ротора θ_6) в режиме S1 при различных значениях тока холостого хода в случае напряжения питания, отличного от номинального. За базовую величину принята температура узла при номинальном входном напряжении.

Таблица 12.

Влияние величины тока холостого хода на температуры узлов АД при изменении напряжения

U	$\frac{I_0}{I_{1н}} = 0.3$		$\frac{I_0}{I_{1н}} = 0.4$		$\frac{I_0}{I_{1н}} = 0.5$		$\frac{I_0}{I_{1н}} = 0.6$		$\frac{I_0}{I_{1н}} = 0.7$		$\frac{I_0}{I_{1н}} = 0.8$	
	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$	$\frac{\theta_5}{\theta_{5н}}$	$\frac{\theta_6}{\theta_{6н}}$
	$\theta_{5н}$	$\theta_{6н}$	$\theta_{5н}$	$\theta_{6н}$	$\theta_{5н}$	$\theta_{6н}$	$\theta_{5н}$	$\theta_{6н}$	$\theta_{5н}$	$\theta_{6н}$	$\theta_{5н}$	$\theta_{6н}$
$U_{1н}$	1	1	1	1	1	1	1	1	1	1	1	1
$0.9U_{1н}$	1.2	1.24	1.16	1.23	1.14	1.2	1.1	1.16	1.06	1.14	0.99	1.04
$0.8U_{1н}$	1.61	1.7	1.55	1.68	1.49	1.66	1.42	1.6	1.36	1.53	1.24	1.36
$0.7U_{1н}$	2.52	2.87	2.44	2.8	2.33	2.73	2.2	2.6	2.06	2.49	1.85	2.21

На рис. 5-6 приведены зависимости температур обмоток статора и ротора от напряжения питания при различных значениях тока холостого хода.

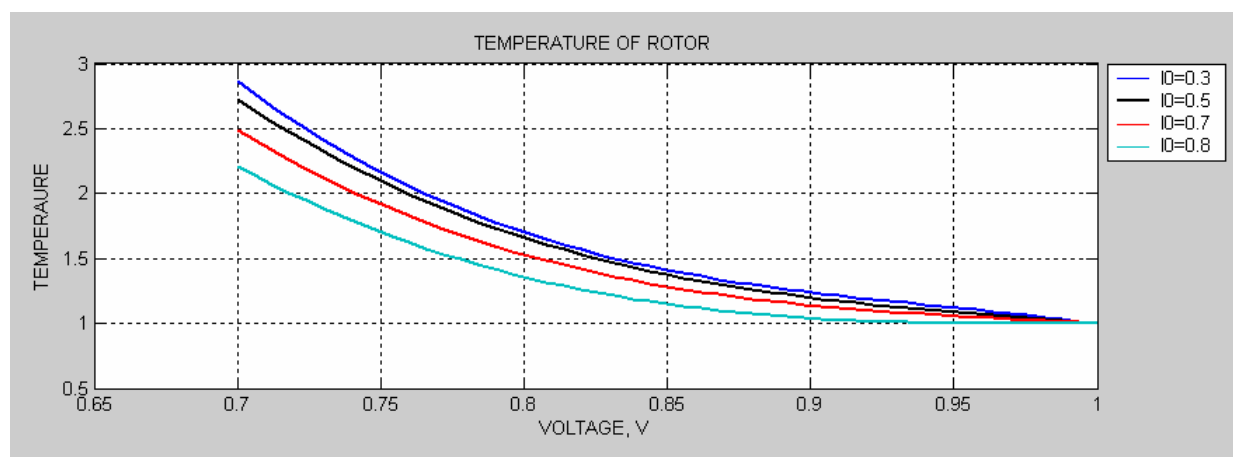


Рис. 5. Зависимость $\theta_p(U_1)$ в относительных единицах при различных значениях тока холостого хода.

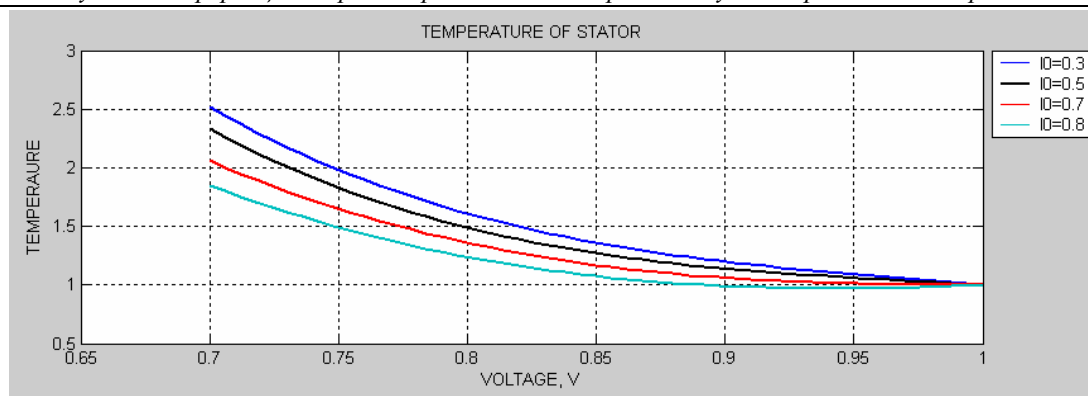


Рис. 6. Зависимость $\theta_c(U_1)$ в относительных единицах при различных значениях тока холостого хода

Из проведенного анализа следует, что снижение тока холостого хода приводит к увеличению темпов роста температур узлов АД при снижении входного напряжения. В ходе анализа влияния величины I_0^* на температуры узлов АД было установлено, как должна изменяться нагрузка при изменении напряжения, чтобы превышение температуры соответствующего узла не превышало допустимого значения при различных значениях тока холостого хода АД в режиме S1 (см. табл. 13, рис. 7) и S3 (табл. 14). Значения коэффициентов снижения нагрузки при различных I_0^* отличаются от соответствующих значений β для АД МТН111-6 и АИУМ225М4 не более, чем на 6%.

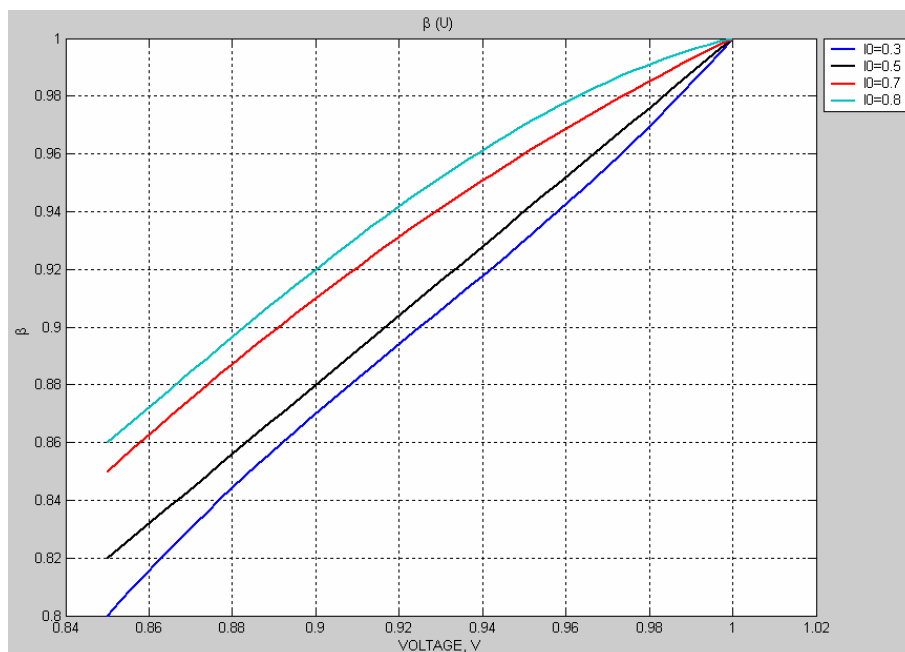


Рис. 7. Зависимости $\beta(U_1)$ в режиме S1 при условии, что перегрев узлов АД не превысит допустимого значения при различных значениях тока холостого хода

Таблица 13.

Влияние тока холостого хода на величину нагрузки при постоянной температуре и изменении напряжения в режиме S1.

	$I_0^* = 0.3$	$I_0^* = 0.5$	$I_0^* = 0.7$	$I_0^* = 0.8$
U_{1H}	1	1	1	1
$0.95U_{1H}$	0.93	0.94	0.96	0.97
$0.9U_{1H}$	0.87	0.88	0.91	0.92
$0.85U_{1H}$	0.8	0.82	0.85	0.86

Таблица 14.

Влияние тока холостого хода на величину нагрузки при постоянной температуре и изменении напряжения в режиме S3.

	$I_0^* = 0.3$	$I_0^* = 0.5$	$I_0^* = 0.7$	$I_0^* = 0.8$
U_{1H}	1	1	1	1
$0.95U_{1H}$	0.96	0.95	0.93	0.93
$0.9U_{1H}$	0.92	0.91	0.89	0.88
$0.85U_{1H}$	0.87	0.86	0.84	0.84

Выводы

Средствами MATLAB проведен анализ теплового состояния различных общепромышленных АД, рассчитанных на напряжение до 1000В, у которых ток холостого хода I_0^* находится в пределах $0.2 \div 0.8$. Получены температуры АД в различных режимах работы и построены зависимости допустимых нагрузок от напряжения питания для широкого класса АД.

Литература

1. Бурковский А. И., Ковалев Е. Б., Коробов В. К. Нагрев и охлаждение электродвигателей взрывонепроницаемого исполнения.— М.: Энергия, 1970.— 184 с.
2. Захарченко П. И., Федоров М. М., Денник В. Ф., Алексеев Е. Р., Карась С. В. Система автоматизированных испытаний и прогнозирования характеристик АД // Уголь Украины.— 1998.— №7.— С.45–48.
3. Miloje M. Kostić. Indirect method for measuring the additional no-load losses in induction motors. ICEM 2000 Proceedings // International Conference of Electrical Machines, 28-30 August 2000.— Helsinki, 2000.— P.1946–1950.
4. Гольдберг О. Д. Испытания электрических машин.— М.: Высшая школа, 2000.—256 с.

УДК 66.045.1

ОРГАНИЗАЦИЯ ИНТЕРАКТИВНОГО ПРОЦЕССА В ЗАДАЧЕ ОПТИМИЗАЦИИ ТЕПЛООБМЕННЫХ УСТРОЙСТВ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ MATLAB

Андрижневский А. А.,

*«Белорусский государственный технологический университет», Минск, Беларусь,
e-mail: aaandril@mail.ru*

Веремеева О. Н.,

*Объединенный институт энергетических и ядерных
исследований (ОИЭЯИ) — «Сосны», Минск, Беларусь,
e-mail: brennt2@mail.ru*

Трифонов А. Г.

*Объединенный институт энергетических и ядерных
исследований (ОИЭЯИ) — «Сосны», Минск, Беларусь,
e-mail: tral@sosny.bas-net.by, tral@list.ru*

При проектировании технологических установок приходится решать традиционный вопрос об оптимальных технико-экономических параметрах теплообменных устройств.

Наиболее распространенной конструкцией теплообменного устройства можно считать теплообменник типа «труба в трубе». Преимущество данной конструкции заключается в разнообразии возможных компоновок и, соответственно, широкой областью ее приложения. Вместе с тем, достаточно высокие удельные характеристики поверхностей теплообмена теплообменников типа «труба в трубе» делают особенно актуальной задачу оптимизации их массогабаритных характеристик на основе выбранных критериев оптимальности. Наиболее общей постановкой оптимальной задачи служит выражение критерия оптимальности в виде экономической оценки.

Авторами разработана программа оптимизации теплообменника типа «труба в трубе» на основе целевой функции, представляющей собой сумму капитальных и эксплуатационных затрат. Программа выполнена в среде MATLAB и дополнена «дружественным» интерфейсом, допускающим интерактивное управление последовательностью выполнения множества *m*-функций. Интерактивная среда позволяет пользователю оперативно вводить и корректировать исходные числовые и текстовые данные, а также визуализировать промежуточные и конечные результаты в текстовом или графическом формате.

Программный комплекс MATLAB совместно с реализованными в нем элементами управления и меню является удобным механизмом отладки многопараметрических задач.

Рис. 1. Блок-схема.

По вычисленным значениям длины происходит проверка на условие $L/d > 50$. Если условие не соблюдается, то вводится поправочный коэффициент e_i и снова производится расчет Re , α , ζ , k , F и L .

Если же расчетные значения L попадают в заданную область, то производится расчет скоростей потоков и далее — необходимой мощности на

на прокачку N , капитальных $Skap$ и эксплуатационных затрат $Sexp$. Результаты расчета приведенных затрат формируются в виде квадратной матрицы, из которой затем выбрасываются нулевые значения и определяется минимум функции, а также значения d_1 и d_2 при найденном минимуме.

Если же при заданных условиях оптимума не существует, в командное окно MATLAB выводится сообщение об ошибке.

Результаты оптимизации записываются в командном окне MATLAB. Для наглядности можно изобразить трехмерный график зависимости целевой функции f или длины от диаметров d_1 и d_2 или скоростей.


Описание графического интерфейса.

Программа расчета и оптимизации теплообменного устройства типа «труба в трубе» реализуется в виде удобного и практичного интерфейса.

При запуске программы появляется окно задания исходных данных, в котором от пользователя требуется задать температуры входа и выхода потоков, расходные характеристики, а также некоторые экономические параметры (см. рис. 2).

исходные данные	
в трубе	
400	температура входа, К
375	температура выхода, К
0.75	расход, кг/с
в межтрубном пространстве	
290	температура входа, К
0.8	расход, кг/с
Экономические параметры	
0.22	стоимость электроэнергии, \$/кВт
7.0	стоимость материала труб, \$/кг

Рис. 2. Окно задания исходных данных.

Для гладких труб расчет оптимальных параметров теплообменника производится при нажатии программной кнопки .

Кроме стандартного набора в меню добавлены новые пункты: *Свойства* и *Оребрение*. К каждому из них присоединены ниспадающие меню, содержащие набор подпунктов.

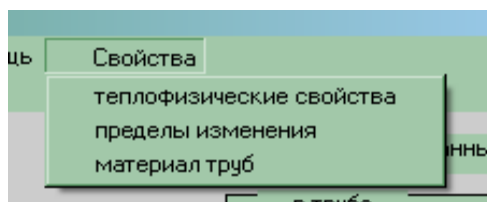


Рис. 3. Подпункты меню.

Пункт *Свойства* содержит три подпункта: *Теплофизические свойства*, *Пределы изменения* и *Материал труб* (см. рис. 3), которые вызывают при нажатии на них мышью новое окно.

При выборе подпункта *Теплофизические свойства* появляется окно (см. рис. 4), где пользователь может выбрать из списка необходимый ему вид теплоносителя как в трубе, так и в межтрубном пространстве или задать теплофизические свойства других теплоносителей.

Задание свойств теплоносителей, не внесенных в список, производится выбором из списка строки «заданные свойства» и нажатием программной кнопки «задать свойства». При этом возникает окно (см. рис. 5), в котором задаются температура и соответствующие ей значения плотности, теплоемкости, теплопроводности и кинематической вязкости рабочей среды.

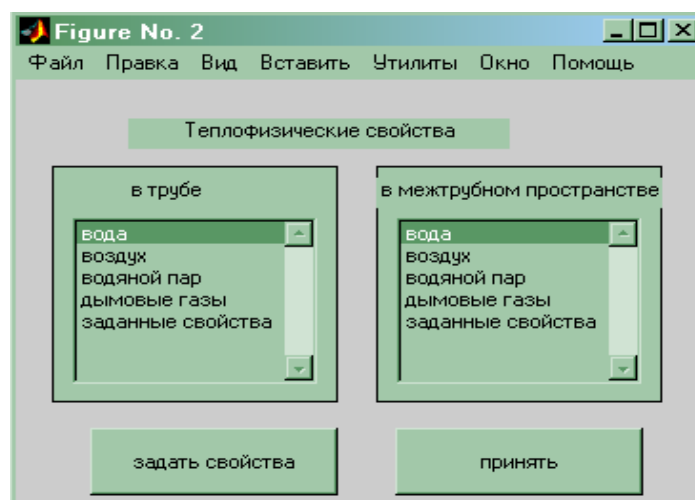


Рис. 4. Окно теплофизических свойств.

Заданные свойства можно сохранить и использовать в дальнейших расчетах. Необходимо отметить, что при сохранении свойств окно автоматически закрывается.

Подпункт *Пределы изменения* выводит на экран окно, показанное на рисунке 6. Здесь можно задать начальные и конечные значения наружного и внутреннего диаметров, по которым будет проводиться выборка их оптимальных значений, а также границы по длине теплообменника.

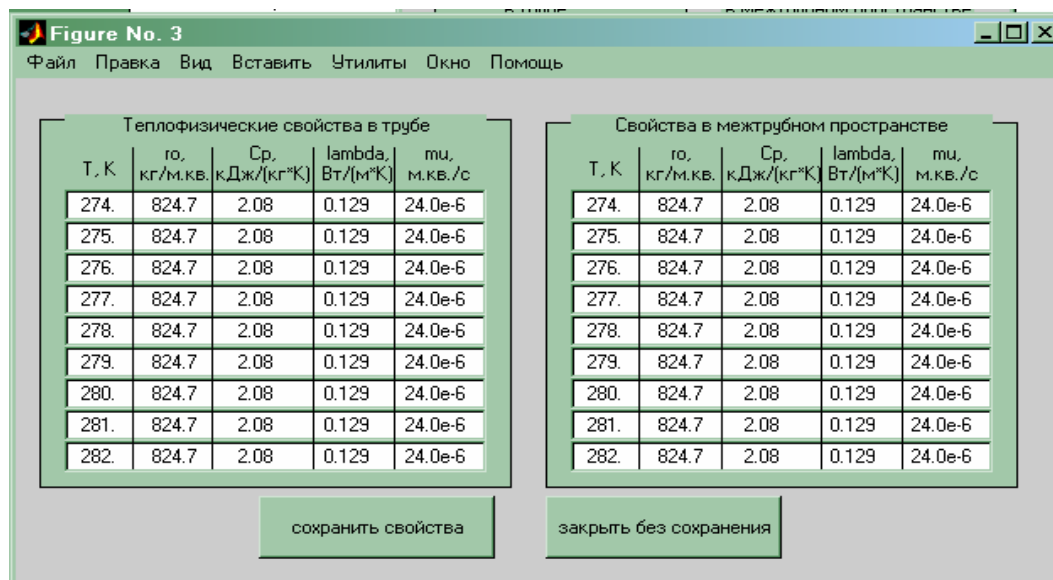


Рис. 5. Окно задания теплофизических свойств.

Подпункт *Материал труб* позволяет выбрать материал, из которого изготовлены трубы. Материал выбирается из соображений коррозионной и химической стойкости, теплопроводности, ударной и вибрационной прочности и т. д.

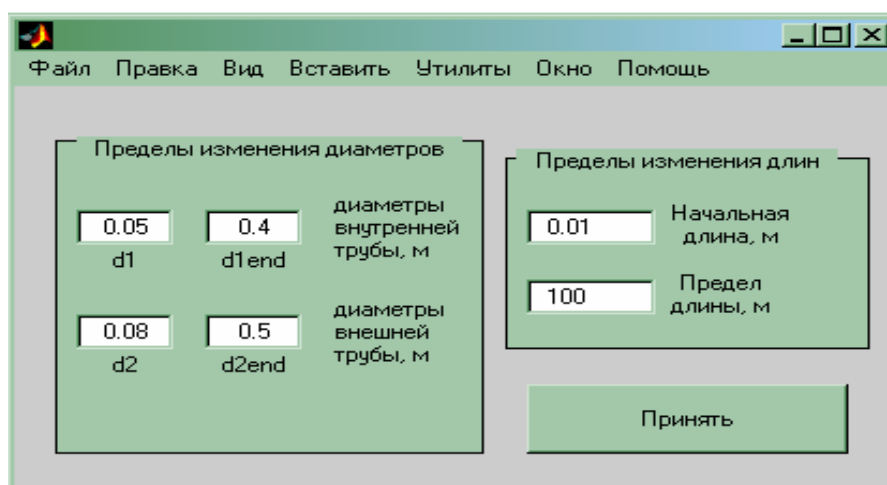


Рис. 6. Окно задания пределов изменения.

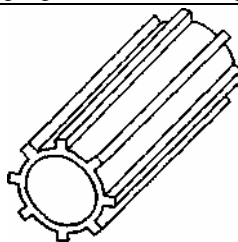


Рис. 7. Окно вида оребрения труб: а — поперечное оребрение трубы; б — продольное оребрение трубы.

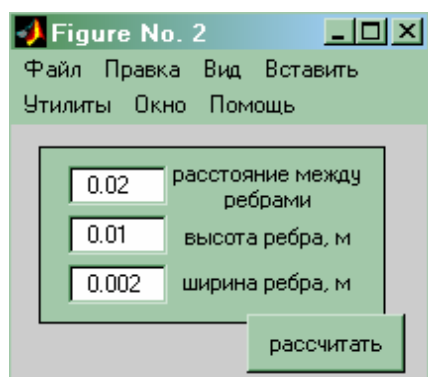


Рис. 8. Окно характеристик оребрения труб.

Пункт Оребрение содержит два подпункта: поперечное (рис. 7, а) и продольное оребрение внутренней трубы (рис. 7, б).

При выборе вида оребрения открывается окно (см. рис.8), в котором задаются характеристики ребер.

После проведения расчетов результаты можно наглядно представить в виде трехмерного графика зависимости целевой функции от различных параметров. Для этого служит кнопка выбрать оси X,Y, которая выводит на экран следующее окно выбора осей (см. рис. 9).



Рис. 9. Окно выбора осей X и Y.

Основные результаты расчета и исходные данные можно посмотреть, нажав на кнопку показать расчетные данные. На экране появляется текстовый файл (см. рис. 10), который легко можно вывести на печать.

1	РАСЧЕТ ТЕПЛООБМЕННИКА "ТРУБА В ТРУБЕ"				
2	ОИЗЯИ-"Сосны"				
3	13-Feb-2004				
4					
5					
6	ИСХОДНЫЕ ДАННЫЕ:				
7	1. Варьируемые диаметры				
8	Заданные диаметры внутренней трубы:				
9	0.0500 м	0.4000 м			
10	Заданные диаметры внешней трубы:				
11	0.0800 м	0.5000 м			
12	-----				
13	2. Температура и расход				
14	Температура на входе в трубу, К				
15	400.00				
16	Температура на выходе, К				
17	375.00				
18	Расход теплоносителя в трубе, кг/с				
19	0.75				
20	Температура на входе в межтрубное пространство, К				
21	290.00				
22	Расход теплоносителя в межтрубном пространстве, кг/с				
23	0.80				
24	-----				
25	3. Ограничения по длине:				
26	0.0100 м	100.0000 м			
27	-----				
28	4. Теплофизические свойства				
29	-в трубе: вода				
30	Температура,	Плотность,	Теплоемкость,	Теплопроводность,	Вязкость,
31	К	кг/м.куб.	кДж/(кг*К)	Вт/(м*К)	м.кв./с
32	293.00	998.2000	4.1830	0.6000	1.006000e-006
33	313.00	992.2000	4.1740	0.6350	6.590000e-007
34	333.00	983.2000	4.1780	0.6600	4.780000e-007
35	353.00	971.8000	4.1950	0.6760	3.660000e-007
36	373.00	968.4000	4.2200	0.6830	2.910000e-007
37	393.00	943.1000	4.2500	0.6870	2.520000e-007
38	413.00	926.1000	4.2870	0.6860	2.160000e-007

Рис. 10. Исходные данные и основные результаты расчета.

Пример расчета

В качестве примера рассчитаем оптимальные конструктивные характеристики теплообменника типа «труба в трубе» для следующей задачи.

Требуется охладить дымовые газы с температуры 600 К до температуры 385 К. Так как внутренняя труба легче поддается очистке, дымовые газы пускаются по трубе, а охлаждающий поток — по межтрубному пространству. В качестве охлаждающего теплоносителя возьмем воздух с температурой на входе 293 К и расходом 0.3 кг/с.

- Исходные данные демонстрационного расчета:

- варьируемые диаметры;
- заданные диаметры внутренней трубы: 0.0500 м 0.4000 м;
- заданные диаметры внешней трубы: 0.0800 м 0.5000 м;
- Температурные и расходные характеристики:
 - температура на входе в трубу: 600.00 К;
 - температура на выходе: 385.00 К;
 - расход теплоносителя в трубе: 0.10 кг/с;
 - температура на входе в межтрубное пространство: 293.00 К;
 - расход теплоносителя в межтрубном пространстве: 0.30 кг/с;
- Ограничения по длине — 0.01 м 100.00 м.

Результаты расчета представлены (см. рис. 11) в виде графика зависимости целевой функции — минимальных затрат — от диаметров труб. Точка минимума (оптимума) показана на графике красной стрелкой.

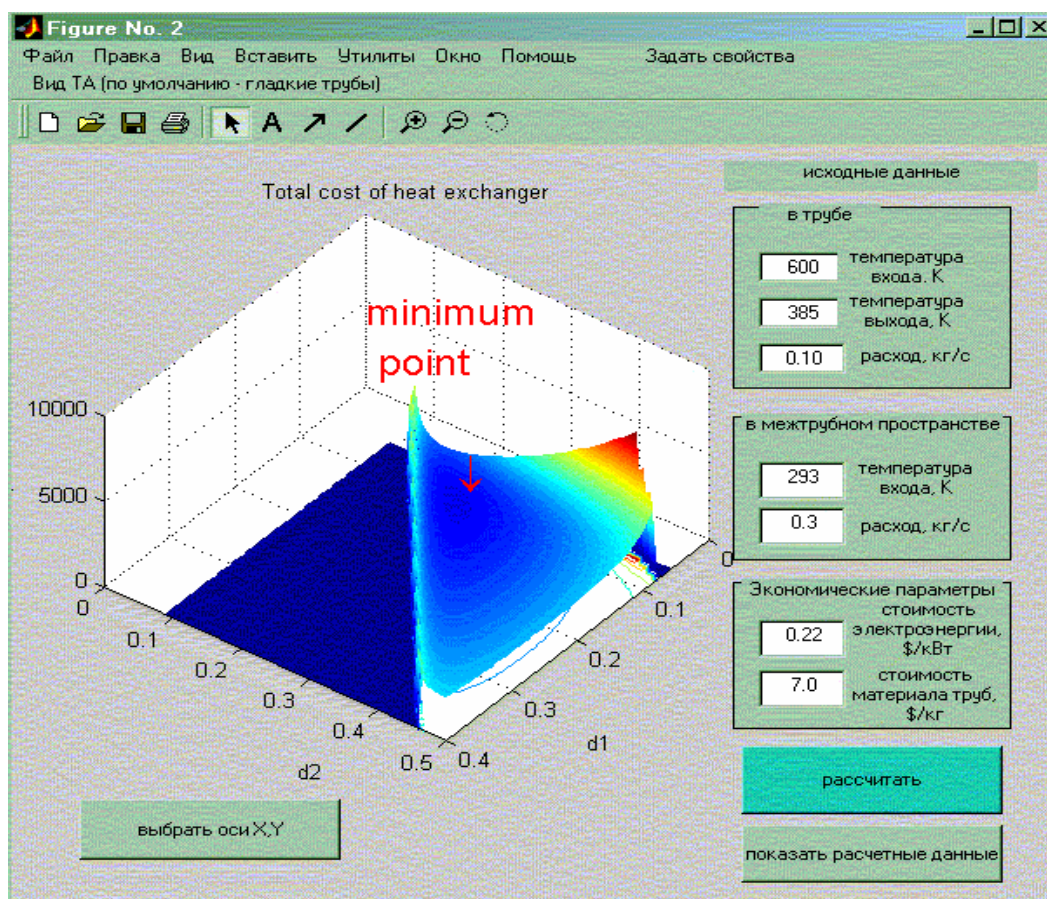


Рис. 11. График минимума целевой функции.

- Полученные в результате расчета цифры:
- Оптимальные диаметры при заданных условиях
 - для внутренней трубы: 0.1060 м;
 - для наружной трубы: 0.2312 м;
 - Скорость потоков при оптимальных диаметрах
 - в трубе: $u_1=15.40$ м/с;

- в межтрубном пространстве: $u_2=12.51$ м/с;
- расчетная длина теплообменника: 20.58 м;
- минимальная стоимость: 997.99 \$.

При оребрении внутренней трубы происходит интенсификация теплопередачи и соответственно уменьшается капитальная составляющая затрат на теплообменник, что можно увидеть из цифр:

- Оптимальные диаметры при заданных условиях
 - для внутренней трубы: 0.0850 м;
 - для наружной трубы: 0.2480 м;
- Скорость потоков при оптимальных диаметрах
 - в трубе: $u_1=20.44$ м/с;
 - в межтрубном пространстве: $u_2=7.46$ м/с;
- Расчетная длина теплообменника: 11.10 м;
- Минимальная стоимость: 374.14 \$.

Но при этом надо учитывать, что стоимость оребренной трубы несколько выше гладкой, что необходимо закладывать в графу «стоимость материала труб».

Заключение

Представленная выше программа оптимизации теплообменника типа «труба в трубе» выполнена в среде MATLAB с использованием интерактивных средств отладки и представления результатов.

Разработанный интерфейс позволяет вводить числовые исходные данные, осуществлять выбор теплофизических и конструкционных параметров теплообменника и визуализировать конечные результаты, как в виде графика, так и в виде текстовой информации.

Программа расчета и оптимизации конструкций теплообменника может быть распространена на более сложные типы теплообменного оборудования, а также на теплообменные системы.

Программа может быть также использована в качестве иллюстративного материала в вузовских дисциплинах, связанных с теплофизикой, моделированием и оптимизацией.

Литература

1. Справочник по теплообменникам: В 2-х т. Т.2 / Пер. с англ. под ред. О. Г. Мартыненко и др.— М.: Энергоатомиздат, 1987.— 352 с.
2. Мартынов Н. Н., Иванов А. П. MATLAB 5.x. Вычисления, визуализация, программирование.— М.: КУДИЦ-ОБРАЗ, 2000.— 336 с.
3. Кафаров В. В., Глебов М. Б. Математическое моделирование основных процессов химических производств: Учеб. пособие для ВУЗов.— М.: Высш. шк., 1991.— 400 с.

УДК 62-50

О ВЫЧИСЛЕНИИ ОБРАТНОЙ ДРАЙЗИНА В СРЕДЕ MATLAB

Асмыкович О. И.

*Белорусский государственный университет, Минск, Беларусь,
e-mail: warmage@tut.by*

Многие математические модели реальных систем управления технологическими процессами наиболее адекватно описываются системами дифференциальных уравнений первого порядка, неразрешенными относительно производной. Причем, если координаты состояния объекта связаны как дифференциальными так и алгебраическими связями, что обычно имеет место в теории электрических цепей, экономических теориях, либо система составлена из взаимосвязанных подсистем, то матрица при производных будет обязательно вырожденной [2]. В теории дифференциальных уравнений такие системы изучались достаточно подробно, но интерес к ним в качественной теории управления динамическими системами резко возрос лишь в последние десятилетия. В настоящей работе используется термин дескрипторные (от английского description-описание), который, по-видимому, достаточно точно отражает специфику таких систем.

Для явной записи решений дескрипторных систем по аналогии с формулой Коши для обыкновенных систем дифференциальных уравнений можно использовать псевдообратные матрицы, в частности, обратную матрицу Драйзина [1–3]. Такие представления получены для регулярных дескрипторных систем с запаздыванием [6], но в них нет реальных алгоритмов расчета обратной матрицы Драйзина. Как отметили авторы [4] во многих системах компьютерной алгебры отсутствуют процедуры непосредственно выполняющие псевдообращение матриц, несмотря на явную практическую в этом потребность. Но в среде MATLAB есть операция нахождения обобщенной обратной, которая и используется в данном докладе.

Следует отметить, что большинство, опубликованных к настоящему времени работ, посвящено регулярным дескрипторным системам, т. е. системам, у которых матрица при производной квадратная и пучок матриц, соответствующих системе, является регулярным. Если же матрица прямоугольная, либо пучок является вырожденным [2], то изучение таких систем значительно сложнее.

Задачи на управление по принципу обратной связи всегда были основополагающими в теории управления. Для дескрипторных систем разрешимость таких задач существенно осложняется тем обстоятельством, что некоторые качественные характеристики этих систем не являются ин-

вариантными относительно линейной обратной связи. Кроме того, важной проблемой является вопрос о возможности модального управления в конкретных системах с помощью конкретных регуляторов и их расчет в системах компьютерной алгебры [7, 8].

1. Обратная матрица Драйзина и ее свойства

Обозначим через $C_{n,m}$ множество всех постоянных матриц размером $n \cdot m$ над полем C .

Определение 1. Наименьшее неотрицательное число k_0 называется индексом матрицы $A_0 \in C_{n,n}$, если $\text{rank } A_0^{k_0} = \text{rank } A_0^{k_0+1}$. Индекс матрицы обозначается через $\text{ind}(A_0)$.

Определение 2. Матрица $X \in C_{n,n}$, являющаяся решением системы матричных уравнений

$$A_0 X = X A_0, \quad X A_0 X = X, \quad X A_0^{k_0+1} = A_0^{k_0},$$

где $k_0 = \text{ind}(A_0)$, называется обратной Драйзина [1] для матрицы $A_0 \in C_{n,n}$.

Если $A \in C^{n \times n}$ и $\text{ind}(A) = k > 0$, то для нее существует матрица P такая что

$$A = P \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} P^{-1},$$

где C — невырожденная матрица, N — нильпотентная матрица с индексом нильпотентности k . Каноническое представление для обратной матрицы Драйзина выглядит следующим образом:

$$A^D = P \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^{-1}.$$

Для любой квадратной матрицы A_0 обратная Драйзина существует, единственна [1] и обозначается A_0^D . Свойства и применение обратной Драйзина можно найти в работах [1–3]. Отметим некоторые ее свойства. Пусть $A_0, B \in C_{n,n}$ и $A_0 B = B A_0$. Тогда $A_0 B^D = B^D A_0$, $A_0^D B = B A_0^D$, $A_0^D B^D = B^D A_0^D$.

Для любой матрицы $A_0 \in C_{n,n}$ индекса k_0 существуют единственные матрицы $M, N \in C_{n,n}$ такие, что $A_0 = M + N$, причем $\text{ind}(M)$ равен 0 или 1, а N является нильпотентной матрицей с индексом нильпотентности k_0 и $MN = NM = 0$. Такое представление называется нильпотентной декомпозицией матрицы A_0 . Матрицы M и N можно вычислить по формулам $M = A_0^2 A_0^D$, $N = A_0 (E_n - A_0^D A_0)$. Кроме того, справедливы равенства

$$A_0^D = M^D, \quad M^D M = M A_0^D, \quad M^D M = A_0^D A_0, \quad (E_n - M^D M) A_0 = N(E_n^* - A_0^D A_0).$$

и если $A \in C^{n \times n}$ и $\text{ind}(A) = k$ то

1. $R(A^D) = R(A^k)$.
2. $N(A^D) = N(A^k)$.
3. Для $A \in C^{n \times n}$, $AA^D A = A$ тогда и только тогда когда $\text{ind}(A) \leq 1$.

Определение 3. Пусть $A \in C_{n,n}$ то $C_A = AA^D A = A^2 A^D = A^D A^2$ называется ядром матрицы A . При этом ядро матрицы A обладает следующим свойством — $A - C_A = N_A$ нильпотентная матрица с индексом нильпотентности $k = \text{ind}(A)$.

Полученная матрица $N_A = A - C_A = (I - AA^D)A$ — называется нильпотентной частью A . Декомпозиция $A = C_A + N_A$ называется ядерно-нильпотентной декомпозицией A .

Относительно ее известен следующий результат [3].

Пусть $A \in C_{n,n}$ и $A = P \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} P^{-1}$, то $C_A = P \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} P^{-1}$ и $N_A = P \begin{bmatrix} 0 & 0 \\ 0 & N \end{bmatrix} P^{-1}$. При этом C_A и N_A являются единственными.

Если $A \in C_{n,n}$ и p — положительное целое число то $C_A^p = C_{A^p}$ и $N_A^p = N_{A^p}$ а также $A^p = C_{A^p} + N_{A^p}$. Кроме того, справедливы следующие свойства:

$$N_A C_A = C_A N_A = 0, \quad N_A A^D = A^D N_A = 0, \quad C_A A A^D = A A^D C_A = C_A, \\ (A^D)^D = C_A, \quad A = C_A \text{ тогда и только тогда когда } \text{ind}(A) \leq 1. \quad ((A^D)^D)^D = A^D, \\ A^D = C_A^D, \quad (A^D)^T = (A^*)^D.$$

2. Алгоритмы нахождения обратной матрицы Драйзина

1. По теореме о канонической форме представления для A находим жорданову форму матрицы A : $J(A) = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}$ с помощью оператора

$$[V, J] := \text{jordan}(A);$$

где V — преобразующая матрица.

2. Для жордановой формы находим преобразующую матрицу V и обратную к ней по команде

$$\text{inv}(V).$$

3. Выделяем невырожденную часть от жордановой формы матрицы $[C]$ путем нахождения ненулевых собственных чисел с помощью функции

eig(A)

и формирования соответствующей диагональной матрицы для матрицы простой структуры или блочно-диагональной в противоположном случае.

4. Далее по команде

inv(C),

находим обратную матрицу для нее C^{-1} .

5. Используя матричное умножение, получаем обратную матрицу Драйзина по формуле $A^D = V \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^{-1}$.

6. Далее выполняем проверку свойств по определению 2.

Другой метод вычисления обратной матрицы Драйзина основан на использовании эшелонной эрмитовой формы [3] и состоит в следующем.

1. Возьмем p — целое число большее индекса матрицы A .

2. Приводим A^p к эшелонной эрмитовой форме и в ней выделяем ненулевые столбцы (v_1, v_2, \dots, v_r) .

3. Строим матрицу $I - H_A$ и также сохраняем ненулевые столбцы $(v_{r+1}, v_{r+2}, \dots, v_n)$.

4. Составляем матрицу P из столбцов (v_1, v_2, \dots, v_n) .

5. Подсчитываем P^{-1} .

6. Формируем матрицу $P^{-1}AP = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}$ и выделяем невырожденную часть C .

7. Находим $A^D = P \begin{bmatrix} C^{-1} & 0 \\ 0 & N \end{bmatrix} P^{-1}$.

8. Выполняем проверку свойств обратной матрицы Драйзина по определению 2.

В отличие от пакетов Maple и Mathematica (см. [4]) в MATLAB для нахождения псевдообратных матриц существует достаточно надежная функция **pinv(A)**, которая имеет широкие применения в линейной алгебре и прикладных задачах математической статистики. Обратная Драйзина и псевдообратная матрица будут всегда отличаться в случае если исходная матрица является вырожденной. Это видно из определения $AA^+A = A$ для псевдообратной и $A^DAA^D = A^D$ для обратной Драйзина и следующего свойства $(A^+)^+ = A$. Но $(A^D)^D = C_A^D$. И поэтому $A^D = A^+$ только если A — невырожденная матрица.

Рассмотрим матрицу размерности 3×3 :

$$A := \begin{bmatrix} 5 & 0 & 0 \\ -1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

Для нее псевдообратная матрица будет выглядеть следующим образом:

$$A^D := \begin{bmatrix} 0.1852 & -0.0370 & -0.0370 \\ 0 & 0.25 & -0.25 \\ 0 & 0.25 & -0.25 \end{bmatrix}.$$

Вычисление обратной матрицы Драйзина для нее будет выглядеть следующим образом/

Данная матрица имеет Жорданову форму вида:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 5 \end{bmatrix}.$$

Здесь $N = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, а $C = [5]$, следовательно $C^{-1} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

Переходная матрица

$$P = \begin{bmatrix} 0 & 0 & 1 \\ \frac{2}{5} & \frac{7}{25} & \frac{-7}{25} \\ \frac{-2}{5} & \frac{3}{25} & \frac{-3}{25} \end{bmatrix},$$

обратная к ней

$$P^{-1} = \begin{bmatrix} 0 & \frac{3}{4} & \frac{-7}{4} \\ 1 & \frac{5}{2} & \frac{5}{2} \\ 1 & 0 & 0 \end{bmatrix},$$

отсюда обратная матрица Драйзина

$$A^D = P \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{5} \end{bmatrix} P^{-1} = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ \frac{-7}{125} & 0 & 0 \\ \frac{-3}{125} & 0 & 0 \end{bmatrix}.$$

При проверке по определению получили, что данная матрица является обратной матрицей Драйзина.

Литература

1. *Drazin M. P.* Pseudoinverses in associate rings and semigroups // Amer. Math. Monthly.— 1958.— V.65.— P.506–514.
2. *Dai L.* Singular Control Systems. Lecture Notes in Control and Information Sciences. V.118.— Berlin, Springer-Verlag, 1989.
3. *Campbell S. L., Meyer C. D., Rose N. J.* Applications of Drazin inverse to the linear systems of differential equations with singular constant coefficients // SIAM J. Appl. Math.— 1976.— V.31.— N.3.— P.411–425.
4. *Икрамов Х. Д., М. Мартин фар* О компьютерно-алгебраических процедурах для псевдообращения матриц // Журнал вычислительной математики и математической физики.— 2003.— Т.43.— №2.— С.163–168.
5. *Толочко О. И., Федоряк Р. В.* Автоматизация синтеза регуляторов и наблюдателей в среде пакета MATLAB // Тр. Всероссийской науч. конф. «Проектирование научных и инженерных приложений в среде MATLAB» (28–29 мая 2002, Москва, Россия).— М.: ИПУ РАН, 2002.— С.482–496.
6. *Игнатенко В. В., Крахотко В. В.* Управляемость линейных дескрипторных систем с запаздыванием // Вестник БГУ. Сер. I.— 1993.— №3.— С.70–73.
7. *Крахотко В. В., Асмыкович О. И.* О стабилизации линейных регулярных дескрипторных систем с запаздыванием // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: Материалы V Республик. науч. конф. студентов и аспирантов (18–20.03.2002).— Гомель, 2002.— С.153–154.
8. *Asmykovich O. I., Krachotko V. V.* Application of package Maple for synthesis descriptor systems with delay // Тез. докл. III Междунар. науч. конф. (Минск, 24–28 сент. 2002).— Мн.: БГУ, 2002.— С.22.

УДК 004.932.4

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЮСТИРОВКИ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ИЗОБРАЖЕНИЙ В СРЕДЕ MATLAB

Бабаян П. В.

*Рязанская государственная радиотехническая академия, Рязань,
e-mail: atr_art@rgta.ryazan.ru*

Проблема юстировки изображений возникает во многих задачах, связанных с обработкой изображений, например, при выявлении изменений в последовательностях видеок кадров, при мультиспектральном анализе изображений. Сущность юстировки состоит в приведении изображений к одной и той же системе координат таким образом, чтобы точки изображений, имеющие одинаковые координаты, являлись проекциями одной и той же точки пространства.

Для решения подобных задач Image Processing Toolbox предоставляет пользователю набор функций **cpselect**, **cp2tform**, **cpcorr**. Однако, в ряде случаев, особенно при анализе видеопоследовательностей, возможностей этих функций оказывается недостаточно. Это обстоятельство привело к разработке нового программного приложения, обладающего расширенными возможностями.

Приложение, разработанное в среде MATLAB, предназначено для юстировки двух видеопоследовательностей, снятых в различных спектральных диапазонах, например, в видимом и тепловом диапазоне. Определение параметров геометрических преобразований осуществляется на основе четырех типов опорных соответствий между исходными изображениями: точка-точка, прямая-прямая, точка-прямая, прямая-точка [1–2]. Учет дополнительных соответствий позволяет более точно оценить параметры преобразований. Выбор опорных соответствий производится оператором с помощью специально разработанного графического интерфейса. В качестве модели преобразования можно выбрать аффинную, проективную модели или преобразование подобия. Одновременно, пользователь может произвести коррекцию временного сдвига между исходными видеопоследовательностями.

Набор опорных соответствий является достаточным для определения параметров преобразований, если выполняется условие:

$$2N + M + L + 2K \geq R,$$

где N — число соответствий точка-точка, M — количество соответствий точка-прямая, L — количество соответствий прямая-точка, K — количество соответствий прямая-прямая, R принимает значение 8, 6, 4 для проек-

тивного преобразования, аффинного преобразования и преобразования подобия соответственно.

Исходные видеопоследовательности и результаты юстировки могут быть представлены как в виде одного файла avi, так и в виде папки с файлами bmp.

В дальнейшем планируется усовершенствовать приложение с путем добавления алгоритма автоматического поиска опорных соответствий.

Литература

1. Алпатов Б. А., Бабаян П. В. Исследование алгоритма электронной юстировки изображений при двухспектральном наблюдении // Цифровая обработка сигналов и ее применения: Тез. докл. 5-й международной конф.— М.: МЦНТИ, 2003.— С.410–412.
2. Бабаян П. В. Оценивание параметров проективного преобразования по опорным элементам в задаче электронной юстировки двухспектральных видеопоследовательностей // Вестник Рязанской государственной радиотехнической академии.— 2003.— Вып.12.— С.27–31.

УДК 539.3

ИСПОЛЬЗОВАНИЕ ПАКЕТА MATLAB ДЛЯ ВЫЧИСЛЕНИЯ УПРУГИХ СВОЙСТВ ДИСПЕРСНОУПРОЧНЕННЫХ МАТРИЧНЫХ КОМПОЗИТОВ С НЕИЗОМЕТРИЧНЫМИ ВКЛЮЧЕНИЯМИ

Бардушкин В. В.

*Московский институт электронной техники (технический университет), Москва,
e-mail: bardushkin@mail.ru*

Описание свойств и вычисление физико-механических характеристик многокомпонентных неоднородных материалов основывается на использовании тензорных (матричных) величин. При этом приходится производить достаточно сложные численные расчеты, что требует привлечения специализированных пакетов прикладных программ. Система MATLAB, ориентированная в первую очередь на матричные вычисления, позволяет достаточно эффективно осуществлять подобные расчеты.

В статье на основе использования пакета MATLAB проведено вычисление эффективных упругих характеристик дисперсноупрочненных матричных композитов, широко применяемых в технике в качестве конструкционных и другого назначения материалов. Армирование их, как правило, производится включениями, имеющими неизометричную форму (например, короткие волокна или эллипсоиды), а также определенным образом ориентированными своими главными осями друг относительно друга. Подобное упорядочение (так называемая текстура формы включений) приводит к существенной анизотропии физико-механических свойств материалов, что необходимо учитывать при создании конструкций из них. Необходимость в разработке методов моделирования структуры, подбора компонент и прогнозирования влияния последних на свойства композиционных материалов обусловлена дороговизной технологий их создания [1, 2]. Для поиска новых материалов, обладающих заданными свойствами, в [3] разработан достаточно простой и в то же время эффективный метод расчета упругих свойств матричных композитов с включениями эллипсоидальной формы. В основе предложенного метода лежит метод обобщенного сингулярного приближения теории случайных полей [1, 2]. Особенностью данного метода является учет взаимодействий включений и возможности анализа формы элемента неоднородности (включения).

Рассмотрим матричный композит, состоящий из двух компонент с объемными концентрациями ν_1 и ν_2 (индекс «1» относится к включению, а «2» — к матрице), представительный элемент объема которого приведен

на рис. 1. Пусть имеются включения эллипсоидальной формы с полуосями l_1 , l_2 и l_3 , ориентированные главной осью l_3 в направлениях x , y и z , с концентрациями h_1 , h_2 и h_3 соответственно ($h_1 + h_2 + h_3 = v_1$). Положим, что материал включений и материал матрицы изотропны.

В приближении однородности полей напряжений и деформаций в пределах каждого включения в [3] получено удобное для численного анализа соотношение для вычисления компонент тензора модулей упругости композиционных материалов (индекс «*» указывает на то, что вычисляются эффективные характеристики):

$$c^* = (h_1 \langle c_1 (I - g_1 c_1'')^{-1} \rangle_{v_1} + h_2 \langle c_1 (I - g_2 c_1'')^{-1} \rangle_{v_1} + h_3 \langle c_1 (I - g_3 c_1'')^{-1} \rangle_{v_1} + v_2 \langle c_2 \rangle_{v_2}) \times \\ \times (h_1 \langle (I - g_1 c_1'')^{-1} \rangle_{v_1} + h_2 \langle (I - g_2 c_1'')^{-1} \rangle_{v_1} + h_3 \langle (I - g_3 c_1'')^{-1} \rangle_{v_1} + v_2 I)^{-1},$$

где I — единичный симметричный тензор четвертого ранга; угловые скобки в выражении определяют усреднение по объему; двойным штрихом обозначена разность между величинами неоднородной среды и тела сравнения (параметры тела сравнения равны упругим модулям матрицы [2]); тензоры g_1 , g_2 и g_3 — интегралы от сингулярной составляющей тензора Грина уравнения равновесия, зависящие только от дельта-функции Дирака (но не от радиус-вектора) и ориентированные в пространстве эйлеровых углов в направлениях осей x , y и z соответственно.

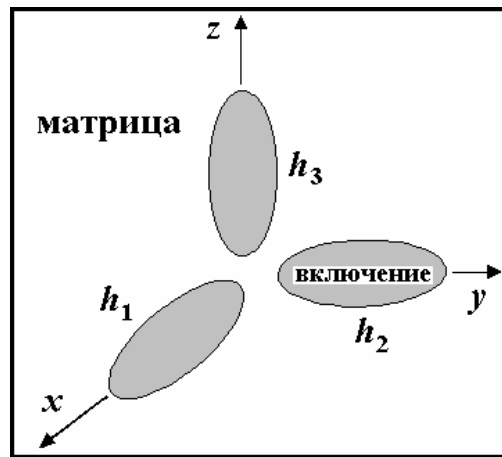


Рис. 1. Представительный элемент объема матричного композита.

На основе данного метода в работе исследовано влияние объемной доли v_1 неизометричных включений на анизотропию упругих свойств матричных композитов в направлениях x и z (эти параметры определялись по формулам $A_x = (c_{11}^* - c_{12}^*) / (2c_{44}^*)$ и $A_z = (c_{33}^* - c_{23}^*) / (2c_{66}^*)$). Анизотропия в направлении y аналогична анизотропии в направлении x . Рассматривался случай «мягкая матрица — жесткие включения», т. е. $(c_{11}^e / c_{11}^M) > 1$ (верхние

индексы «м» и «в» относятся к матрице и включениям соответственно), а также варианты армирования, когда эллипсоидальные включения (диски и волокна) ориентированы большими полуосями в трех взаимно перпендикулярных направлениях x, y, z ; только в направлениях x и y ; только в направлении z .

В результате проведенных с помощью пакета MATLAB расчетов эффективных упругих свойств матричных композитов при армировании дисками ($l_1 = l_2 = 1, l_3 = 0,01$) и волокнами ($l_1 = l_2 = 1, l_3 = 100$) было установлено, что при $h_1 = h_2 = h_3 \neq 0$ получается кубическая, при $h_1 = h_2 \neq 0$ и $h_3 = 0$ — тетрагональная, при $h_1 = h_2 = 0$ и $h_3 \neq 0$ — гексагональная симметрии эффективных свойств материала.

При исследовании зависимости коэффициентов анизотропии A_x и A_z от объемной доли включений (см. рис. 2 и 3) величина v_1 варьировалась в пределах от 0 до 0,5. Ограничение сверху на переменную v_1 определяется условиями модели. При $v_1 > 0,5$ роль включений будет играть матрица, а роль матрицы — включения, тогда получится другая модель с совершенно другим представительным элементом объема.

Точка $v_1 = 0$ на рис. 2, 3 соответствует материалу, в котором отсутствуют армирующие элементы. В ней параметры анизотропии равны 1, что определяет изотропию эффективных упругих свойств матрицы.

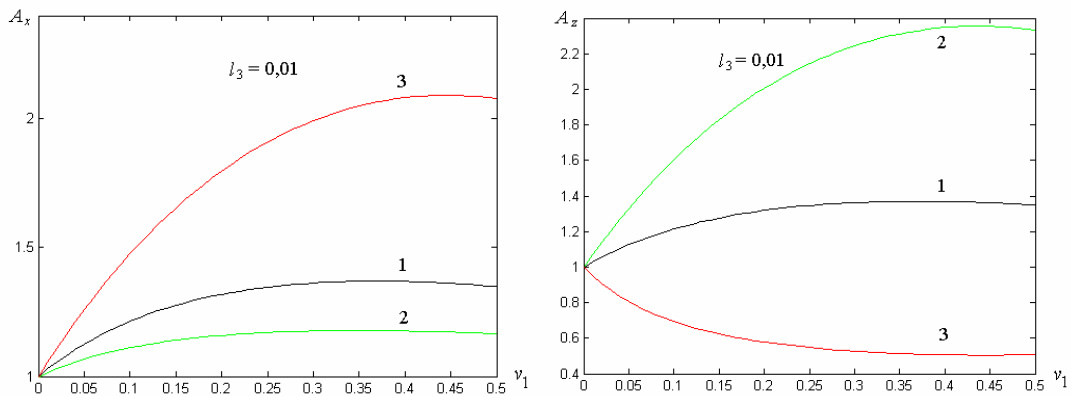


Рис. 2. Параметры анизотропии при армировании дисками в случае $(c_{11}^e / c_{11}^m) = 10$; 1 — кубическая, 2 — тетрагональная, 3 — гексагональная симметрии эффективных свойств.

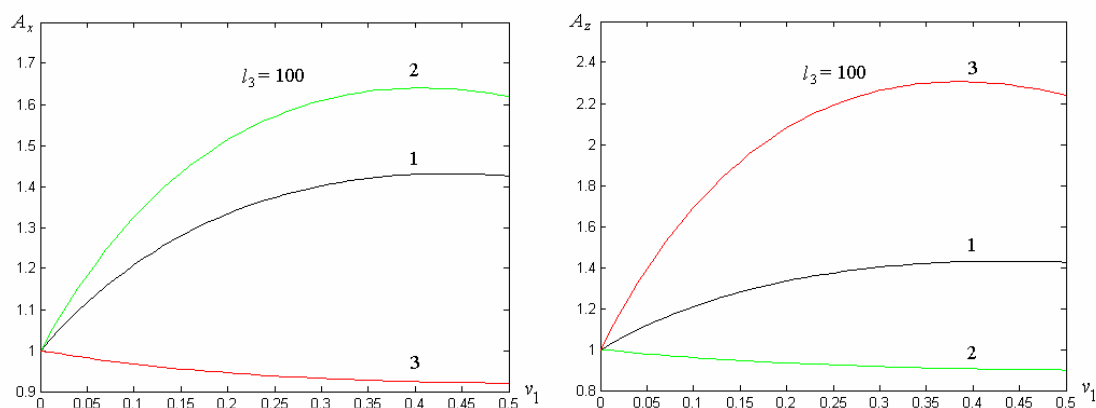


Рис. 3. Параметры анизотропии при армировании волокнами в случае $(c_{11}^e / c_{11}^m) = 10$; 1 — кубическая, 2 — тетрагональная, 3 — гексагональная симметрии эффективных свойств.

На основании расчетов зависимости коэффициентов анизотропии A_x и A_z от объемной доли включений при армировании дисками и волокнами для случая «мягкая матрица — жесткие включения» можно сделать следующие выводы:

- как при армировании дисками, так и при армировании волокнами зависимости коэффициентов A_x и A_z от объемной доли включений имеют нелинейный характер, причем у исследуемых параметров наблюдаются экстремумы при $v_1 \approx 0,4$;
- при армировании дисками наименьшая вариация параметра A_x от v_1 наблюдается для материалов с тетрагональной симметрией эффективных свойств, а параметра A_z — для материалов с кубической симметрией; наибольшая вариация параметра A_x от v_1 наблюдается для материалов с гексагональной симметрией эффективных свойств, а параметра A_z — для материалов с тетрагональной симметрией;
- при армировании волокнами наименьшая вариация параметра A_x от v_1 наблюдается для материалов с гексагональной симметрией эффективных свойств, а параметра A_z — для материалов с тетрагональной симметрией; наибольшая вариация параметра A_x от v_1 наблюдается для материалов с тетрагональной симметрией эффективных свойств, а параметра A_z — для материалов с гексагональной симметрией.

В работе также проведено исследование зависимости параметров анизотропии от объемной доли включений при изменении отношения упругих модулей включений и матрицы.

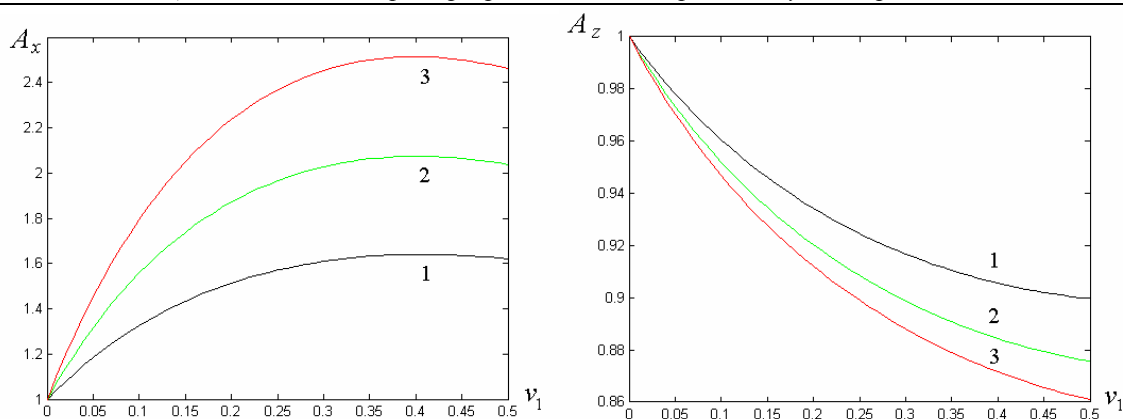


Рис. 4. Вариация параметров анизотропии от величины (c_{11}^s / c_{11}^m) при армировании волокнами в случае тетрагональной симметрии эффективных свойств.

На рис. 4 приведены графики этих зависимостей только для тетрагональной симметрии эффективных свойств материала при армировании волокнами. Рассматривались случаи $(c_{11}^s / c_{11}^m) = 10$ — кривая 1; $(c_{11}^s / c_{11}^m) = 15$ — кривая 2; $(c_{11}^s / c_{11}^m) = 20$ — кривая 3.

Следует отметить, что в случаях тетрагональной (при армировании дисками), кубической и гексагональной симметрий эффективных свойств наблюдалась аналогичная ситуация. С повышением жесткости включений по отношению к жесткости матрицы происходило увеличение анизотропии с сохранением характера данной зависимости.

Литература

1. Шермергор Т. Д. Теория упругости микронеоднородных сред.— М.: Наука, 1977.— 399 с.
2. Васильев В. А., Митин Б. С., Яковлев В. Б. и др. Высокоскоростное затвердевание расплава. (Теория, технология, материалы).— М.: ИнтерметИнжиниринг, 1998.— 395 с.
3. Канович М. З., Бардушкин В. В., Яковлев В. Б., Верецагина Н. В. Анизотропия физико-механических свойств дисперсноупрочненных матричных композитов с неизометричными включениями // Сб. докладов межд. конгресса «Мехтриботранс-2003».— Ростов-на-Дону, 2003.— Т.1.— С.398–401.

УДК 548.735.6 : 552.112

ИСПОЛЬЗОВАНИЕ СИСТЕМЫ MATLAB ПРИ МОДЕЛИРОВАНИИ НАПРЯЖЕННО- ДЕФОРМИРОВАННОГО СОСТОЯНИЯ ОЛИВИНОВ

*Бардушкина И. В., Бардушкин В. В.,
Московский институт электронной техники
(государственный технический университет), Москва,
e-mail: bardushkin@mail.ru,*

*Никитин А. Н.,
Объединенный институт ядерных исследований, ЛНФ, Дубна;
Сычев А. П.,*

*Ростовский государственный университет путей сообщения, Ростов-на-Дону,
Яковлев В. Б.*

*Московский институт электронной техники
(государственный технический университет), Москва,*

На основе данных нейтронографического эксперимента (реактор ИБР-2 в ОИЯИ г. Дубна) восстановлена функция распределения ориентаций кристаллографических осей зерен неоднородностей для исследуемого образца оливина. Методом обобщенного сингулярного приближения рассчитаны эффективные упругие постоянные, операторы концентраций напряжений, деформаций и тензор напряжений данного поликристаллического материала. Вычисления проведены в системе MATLAB.

Проектирование и моделирование реальных объектов, как правило, приводит к задачам, требующим численного решения. Задачи механики и физики композиционных материалов и связанные с ними задачи геофизики, конструкционного материаловедения, функциональной электроники с вычислительной точки зрения усложнены в связи с использованием тензорных и матричных величин, а также из-за необходимости проведения их серьезного численного анализа и отображения полученных результатов. Поэтому для решения такого типа задач встает проблема выбора удобного, относительно простого для использования, но в то же время эффективного пакета прикладных программ. Для проведения моделирования в данной области нами использовались различные пакеты и языки программирования. Однако наиболее удобной, как нам кажется, является система MATLAB.

В работе проведено моделирование свойств (локального напряженно-деформированного состояния) поликристаллической среды (оливины естественного происхождения, составляющие основу мантии Земли, — объект пристального исследования геофизики). Основными факторами, определяющими свойства подобных поликристаллических систем, явля-

ются минеральный состав, свойства образующих их минералов и текстура. Последняя имеет большое значение, ее наличие влияет на многие физико-механические свойства этих материалов. Текстура присуща как естественным поликристаллам, так и искусственным. Для поликристаллических систем различают два вида текстур — текстура формы и ориентационная текстура. Первая связана с формой и размерами компонентов. Ориентационная текстура определяется преимущественной ориентировкой кристаллитов в поликристалле. Если исключено влияние других причин, то возможно вычисление эффективных характеристик таких материалов [1, 2]. Исследование текстур, включающее определение размеров и взаимной ориентации кристаллитов, осуществляется главным образом рентгенографическими, электронографическими и оптическими методами. В последнее время для получения более полной информации о текстуре массивных образцов применяют нейтронографические методы. К настоящему времени разработан ряд методов, которые позволяют количественно описывать текстуру исследуемого материала при помощи функции распределения кристаллитов по ориентациям (ФРО). При этом ее получение является основной задачей текстурного анализа. К сожалению, ФРО не поддается прямому измерению. Это приводит к задаче количественного текстурного анализа — вычислению ФРО по конечному числу полюсных фигур, являющихся наглядным способом представления предпочтительных ориентировок кристаллитов [3, 4]. Полюсной фигурой называется графическое изображение функции распределения нормалей (полюсов) к одной определенной кристаллографической плоскости $\{hkl\}$ образца. Для графического изображения функции используется стереографическая проекция. При этом функция распределения полюсов представляет собой вероятность совпадения нормали к кристаллографической плоскости $\{hkl\}$ с различными направлениями в образце.

В работе показано, как, используя данные нейтронографического эксперимента на пучке №7а импульсного реактора ИБР-2 (Объединенный институт ядерных исследований г. Дубна), с помощью специально разработанного комплекса программ в системе MATLAB могут быть решены задачи восстановления ФРО и определения эффективных упругих характеристик, а также рассчитаны операторы концентраций напряжений и деформаций поликристаллических материалов. Вычисления в пакете MATLAB эффективных упругих модулей и операторов концентраций напряжений и деформаций продемонстрированы на примере поликристаллического материала естественного происхождения оливина BQ4, образцы которого были отобраны из базальтовых лав массива Кукес Албанской мирдитовой зоны.

Рассмотрим общую схему расчета эффективных упругих характеристик текстурированных поликристаллов в рамках теории случайных функций. Для поликристаллических материалов тензоры напряжений σ , дефор-

маций ε , модулей упругости c и податливости s являются случайными функциями координат и могут быть представлены в виде суммы средних значений и флуктуаций

$$\sigma(\mathbf{r}) = \langle \sigma(\mathbf{r}) \rangle + \sigma'(\mathbf{r}), \quad \varepsilon(\mathbf{r}) = \langle \varepsilon(\mathbf{r}) \rangle + \varepsilon'(\mathbf{r}), \\ c(\mathbf{r}) = \langle c(\mathbf{r}) \rangle + c'(\mathbf{r}), \quad s(\mathbf{r}) = \langle s(\mathbf{r}) \rangle + s'(\mathbf{r}).$$

В рамках линейной теории флуктуации линейно зависят от средних значений

$$\sigma'(\mathbf{r}) = P(\mathbf{r})\langle \sigma(\mathbf{r}) \rangle, \quad \varepsilon'(\mathbf{r}) = Q(\mathbf{r})\langle \varepsilon(\mathbf{r}) \rangle. \quad (1)$$

Здесь тензорные операторы $P(\mathbf{r})$ и $Q(\mathbf{r})$ являются в общем случае интегральными операторами, описывающими взаимодействие между кристаллическими включениями. Угловые скобки в выражениях определяют усреднение по ансамблю, которое для статистически однородных поликристаллов, т. е. при выполнении гипотезы эргодичности, совпадает с усреднением по объему. Тогда связь между локальными и средними напряжениями и деформациями в материале может быть представлена в виде

$$\sigma(\mathbf{r}) = (I + P(\mathbf{r}))\langle \sigma(\mathbf{r}) \rangle, \quad \varepsilon(\mathbf{r}) = (I + Q(\mathbf{r}))\langle \varepsilon(\mathbf{r}) \rangle, \quad (2)$$

где I — единичный симметричный тензор четвертого ранга.

Таким образом, локальные напряжения и деформации можно охарактеризовать безразмерными операторами концентраций напряжений и деформаций (тензоры четвертого ранга), представляющими собой отношения локальных к средним значениям

$$K^\sigma(\mathbf{r}) = I + P(\mathbf{r}), \quad K^\varepsilon(\mathbf{r}) = I + Q(\mathbf{r}). \quad (3)$$

Удобство такого представления заключается в том, что при данных предположениях операторы концентраций напряжений и деформаций должны зависеть только от материальных параметров среды и микроструктуры материала, а не от прикладываемых нагрузок.

Из уравнений (1)–(3) легко получить в предположении однородности полей деформаций ($\varepsilon'(\mathbf{r}) = 0$) и напряжений ($\sigma'(\mathbf{r}) = 0$) в поликристаллическом материале, что операторы концентраций напряжений и деформаций имеют вид

$$K^\sigma(\mathbf{r}) = c(\mathbf{r})\langle c(\mathbf{r}) \rangle^{-1}, \quad K^\varepsilon(\mathbf{r}) = s(\mathbf{r})\langle s(\mathbf{r}) \rangle^{-1}.$$

Другим предельным значением для этих соотношений является единичный тензор I .

Для приближений, учитывающих взаимодействие зерен неоднородностей (кристаллитов) между собой, вычисления усложняются. Для корректного анализа операторов концентраций напряжений и деформаций в этом случае необходимо решать систему дифференциальных уравнений равновесия с соответствующими граничными условиями

$$L_{ij}(\mathbf{r})u_j(\mathbf{r}) = -f_i(\mathbf{r}), \quad (4)$$

где $L_{ij}(\mathbf{r}) = \nabla_k c_{ijkl}(\mathbf{r}) \nabla_l$ — дифференциальный оператор, а $f_i(\mathbf{r})$ и $u_i(\mathbf{r})$ — компоненты вектора объемных сил и вектора смещения. Дифференциаль-

ное уравнение (4) является стохастическим дифференциальным уравнением 2-го порядка. Для бесконечной среды решение этого уравнения обычно ищут, используя метод функций Грина с помощью специально вводимого однородного тела сравнения [2, 5]. Обозначим операторы и характеристики тела сравнения верхним индексом «с», а разность между величинами неоднородной среды и тела сравнения двойным штрихом. После преобразований [2, 5] получим, что связь между локальными и средними по материалу деформациями будет иметь вид

$$\varepsilon(\mathbf{r}) = (I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} < (I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} >^{-1} < \varepsilon(\mathbf{r}) >.$$

Откуда, согласно определению, получим оператор концентраций деформаций

$$K^{\varepsilon}(\mathbf{r}) = (I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} < (I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} >^{-1}.$$

Принимая во внимание выражение для расчета эффективных характеристик [2, 5]

$$c^* = < c(\mathbf{r})(I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} > < (I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} >^{-1},$$

получим формальное выражение для оператора концентраций напряжений

$$K^{\sigma}(\mathbf{r}) = c(\mathbf{r})(I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} < c(\mathbf{r})(I - Q(\mathbf{r})c''(\mathbf{r}))^{-1} >^{-1}.$$

В работе использовалось обобщенное сингулярное приближение теории случайных функций, которое предполагает наложение некоторых ограничений на ядро интегрального оператора Q . Воспользуемся для вычислений только сингулярной составляющей тензора Грина уравнения равновесия (4), зависящей лишь от дельта-функции Дирака, но не от радиус-вектора. В этом случае интегральная свертка дельта-функции переводит интегральный оператор Q в функционал g . Физический смысл такого приближения заключается в предположении однородности полей напряжений и деформаций только в пределах неоднородности. Расчетной формулой обобщенного сингулярного приближения теории случайных функций является выражение

$$c^* = < c(\mathbf{r})(I - gc''(\mathbf{r}))^{-1} > < (I - gc''(\mathbf{r}))^{-1} >^{-1}. \quad (5)$$

Используя эти предположения, получаются следующие соотношения для операторов концентраций полей напряжений и деформаций

$$K^{\sigma}(\mathbf{r}) = c(\mathbf{r})(I - gc''(\mathbf{r}))^{-1} < c(\mathbf{r})(I - gc''(\mathbf{r}))^{-1} >^{-1}, \quad (6)$$

$$K^{\varepsilon}(\mathbf{r}) = (I - gc''(\mathbf{r}))^{-1} < (I - gc''(\mathbf{r}))^{-1} >^{-1}. \quad (7)$$

Соотношения (5)–(7) являются основополагающими в рамках обобщенного сингулярного приближения теории случайных полей и уже учитывают взаимодействия между зернами кристаллитов.

Соотношение (5) можно представить в несколько ином, более удобном для вычислений виде

$$c^* = < (c(\mathbf{r}) - b)^{-1} >^{-1} + b, \quad b = c^c + g^{-1}. \quad (8)$$

Если принять, что кристаллиты неоднородного материала отличаются лишь ориентировкой кристаллографических осей, то процедура усред-

нения в выражении (8) сводится к интегрированию только по всевозможным эйлеровым углам $\varphi_1, \Phi, \varphi_2$

$$(c^* - b)^{-1} = \frac{1}{8\pi^2} \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} f(\varphi_1, \Phi, \varphi_2) (c(\varphi_1, \Phi, \varphi_2) - b)^{-1} \sin \Phi d\Phi d\varphi_1 d\varphi_2, \quad (9)$$

где $f(\varphi_1, \Phi, \varphi_2)$ — функция распределения зерен кристаллитов по ориентациям (ФРО) [2, 5].

Одним из методов решения данной задачи для поликристаллических текстурированных материалов является представление ФРО в виде разложения в ряд по обобщенным шаровым функциям [1] (здесь g — элемент эйлерова пространства вращений, определяемый углами $\varphi_1, \Phi, \varphi_2$). Обобщенные шаровые функции определяются формулой

$$T_l^{mn}(g) = \exp(im\varphi_1) \cdot P_l^{mn}(\cos \Phi) \cdot \exp(in\varphi_2),$$

здесь $P_l^{mn}(\cos \Phi)$ — обобщенные функции Лежандра. При $x = \cos \Phi$ они имеют вид

$$P_l^{mn}(x) = \frac{(-1)^{l-m} \cdot i^{n-m}}{2^l \cdot (l-m)!} \sqrt{\frac{(l-m)!(l+n)!}{(l+m)!(l-n)!}} \cdot (1-x)^{-\frac{n-m}{2}} \cdot (1+x)^{-\frac{n+m}{2}} \times \\ \times \frac{d^{l-n}}{dx^{l-n}} [(1-x)^{l-m} \cdot (1+x)^{l+m}].$$

При этом

$$f(g) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{n=-l}^l C_l^{mn} T_l^{mn}(g),$$

где C_l^{mn} — коэффициенты разложения.

При вычислении ФРО в качестве базисных можно выбрать чисто действительные функции $S_l^{mn}(g)$. Согласно [1]

$$S_l^{mn}(g) = \frac{i^{m-n}}{\sqrt{2}} \cdot (T_l^{mn}(g) + T_l^{-m-n}(g)), \quad m-n \equiv 0 \pmod{2};$$

$$S_l^{mn}(g) = \frac{i^{m-n+1}}{\sqrt{2}} \cdot (T_l^{mn}(g) - T_l^{-m-n}(g)), \quad m-n \not\equiv 0 \pmod{2};$$

$$S_l^{00}(g) = T_l^{00}(g).$$

Группы симметрии поликристаллического образца и кристаллитов накладывают на ФРО ряд ограничений, которые можно учесть введением симметричных обобщенных функций $G_l^{\mu\nu}(g)$

$$G_l^{\mu\nu}(g) = \sum_{m=-l}^l \sum_{n=-l}^l A_l^{n\nu} B_l^{m\mu} S_l^{mn}(g),$$

где коэффициенты $A_l^{n\nu}$ выбираются так, чтобы была учтена симметрия образца (симметрия внешнего воздействия), коэффициенты $B_l^{m\mu}$ учитывают симметрию кристаллитов. Эти коэффициенты должны быть ортонормиро-

ванными и их можно выбрать соответственно $N(l)$ и $M(l)$ линейно независимыми способами. Окончательно, ФРО $f(g)$ представляется в виде

$$f(g) = \sum_{l=0}^{\infty} \sum_{\mu=1}^{M(l)} \sum_{\nu=1}^{N(l)} \sum_{m=-l}^l \sum_{n=-l}^l C_l^{\mu\nu} A_l^{n\nu} B_l^{m\mu} S_l^{mn}(g).$$

Следует отметить [1], что:

- при вычислении эффективных характеристик материалов, описываемых тензорами ранга не выше r , можно ограничиться только первыми $l \leq r$ членами ряда разложения ФРО по обобщенным шаровым функциям;
- при отбрасывании высших членов ряда по l ФРО может принимать отрицательные значения, т. к. в разложении $f(g)$ используются знакопеременные функции.

Поскольку упругие характеристики исследуемых образцов оливинов описываются тензорами, ранг которых не превышает четырех, то, как правило, для вычисления свойств ограничиваются только тремя четными членами разложения ФРО в ряд по обобщенным шаровым функциям. Поэтому используется функция $f_4(g)$

$$f_4(g) = 1 + \sum_{\mu, \nu} C_2^{\mu\nu} \cdot G_2^{\mu\nu}(g) + \sum_{\mu, \nu} C_4^{\mu\nu} \cdot G_4^{\mu\nu}(g).$$

Коэффициенты $C_2^{\mu\nu}$ и $C_4^{\mu\nu}$ для оливина BQ4 были получены с помощью нейтронографического эксперимента на реакторе ИБР-2 (ОИЯИ, г. Дубна). Их значения представлены в табл. 1.

Таблица 1.

Коэффициенты $C_l^{\mu\nu}$ разложения ФРО в ряд по обобщенным шаровым функциям для оливина BQ4.

l	$\mu \backslash \nu$	– 2	0	2
2	– 2	– 0,092242	0,852391	– 0,098927
2	– 1	0,000000	0,372512	0,085377
2	1	0,453009	0,289194	0,130338
2	2	0,000000	1,513207	– 1,784058

l	$\mu \backslash \nu$	– 4	– 2	0	2	4
4	– 4	– 0,176831	0,463667	0,238963	0,450697	– 0,392826
4	– 2	0,000000	– 0,035040	0,056616	0,521923	0,306626
4	– 1	– 0,326880	– 0,225110	– 0,683218	– 0,109341	0,392633
4	1	0,000000	0,539451	0,496082	– 0,878265	0,045996
4	2	– 0,161919	0,807373	– 0,321464	1,394163	– 1,285078
4	4	0,000000	0,240315	– 0,127011	– 0,288006	– 0,424693

Графики, соответствующие сумме первых трех членов разложения по l ФРО, отражающие особенности текстуры исследуемого образца оливина BQ4 для некоторых углов φ_1 , Φ , φ_2 , представлены на рис. 1.

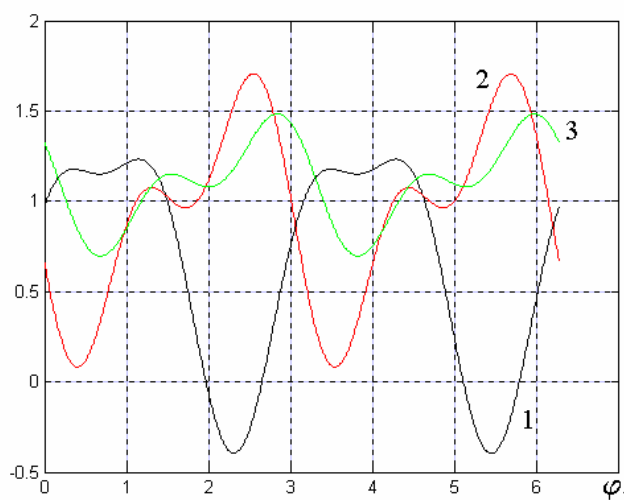


Рис. 1. Сечения неполной ФРО, соответствующей первым трем четным членам разложения при различных значениях Φ и φ_2 : (1) $\Phi = \pi/6$, $\varphi_2 = \pi/3$; (2) $\Phi = \pi/2$, $\varphi_2 = 3\pi/4$; (3) $\Phi = 3\pi/4$, $\varphi_2 = 7\pi/6$.

Таким образом, для получения численных значений эффективных упругих характеристик по формуле (9) необходимо знать ФРО $f(\varphi_1, \Phi, \varphi_2)$ и выбрать параметры тела сравнения. Фактически указанные параметры — это эффективные характеристики поликристалла. Они могут быть рассчитаны в различных приближениях. Если предположить однородность полей напряжений, то надо принять $c^c = 0$ и в (8) получится приближение Ройсса. Если предположить однородность полей деформаций, то надо принять $c^c = \infty$ и в (8) получится приближение Фойгта. Опыт показывает, что наилучший выбор параметров тела сравнения в методе самосогласования получается при использовании приближения Хилла $c^c = c_{\text{Hill}}^*$ (полусумма эффективных характеристик по Фойгту и Ройссу).

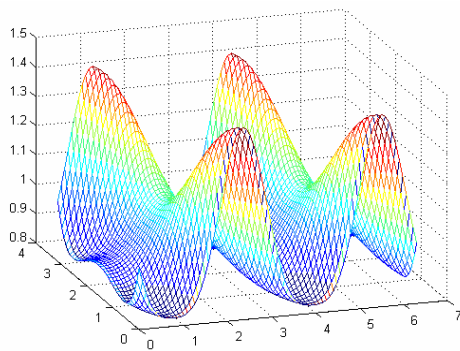
На основе разработанного в системе MATLAB комплекса программ были проделаны вычисления упругих постоянных и операторов концентраций напряжений и деформаций оливина BQ4. В табл. 2 представлены отличные от нуля эффективные упругие постоянные c^* исследуемого образца, рассчитанные в приближениях Фойгта, Ройсса, Хилла и методом самосогласования (в качестве параметров тела сравнения при этом использовано приближение Хилла $c^c = c_{\text{Hill}}^*$).

Таблица 2.

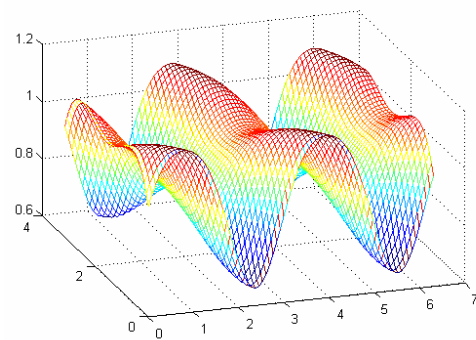
Значения отличных от нуля эффективных модулей упругости c_{ij}^* образца BQ4, ГПа.

	c_{11}^*	c_{22}^*	c_{33}^*	c_{44}^*	c_{55}^*	c_{66}^*	c_{12}^*	c_{13}^*	c_{23}^*
c_{Voight}^*	236,71	250,04	243,51	82,25	82,61	82,50	78,90	75,01	82,46
c_{Reuss}^*	226,50	243,46	234,85	79,37	79,38	79,40	76,20	72,16	80,26
c_{Hill}^*	231,61	246,75	239,18	80,81	80,99	80,95	77,55	73,59	81,36
c_s^*	233,77	248,21	241,08	81,43	81,66	81,60	78,18	74,26	81,88
Монокристалл оливина	324,00	198,00	249,00	66,70	81,00	79,30	59,00	79,00	78,00

На рис. 2–4 для наглядного представления о значениях операторов концентраций напряжений и деформаций приведены поверхности, устанавливающие зависимости этих операторов от углов φ_1 и Φ при $\varphi_2 = \pi/4$ для оливина BQ4.

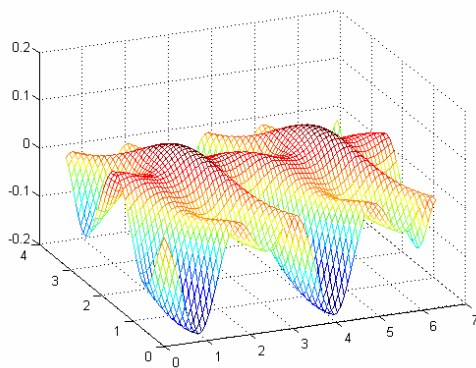


а)

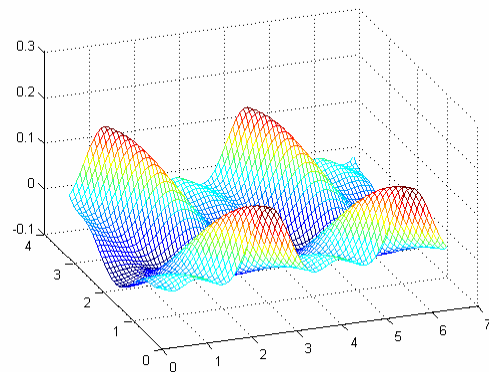


б)

Рис. 2. Зависимости операторов а) K_{1111}^{σ} и б) K_{1111}^{ε} от углов φ_1 и Φ при $\varphi_2 = \pi/4$.



а)



б)

Рис. 3. Зависимости операторов а) K_{2211}^{σ} и б) K_{2211}^{ε} от углов φ_1 и Φ при $\varphi_2 = \pi/4$.

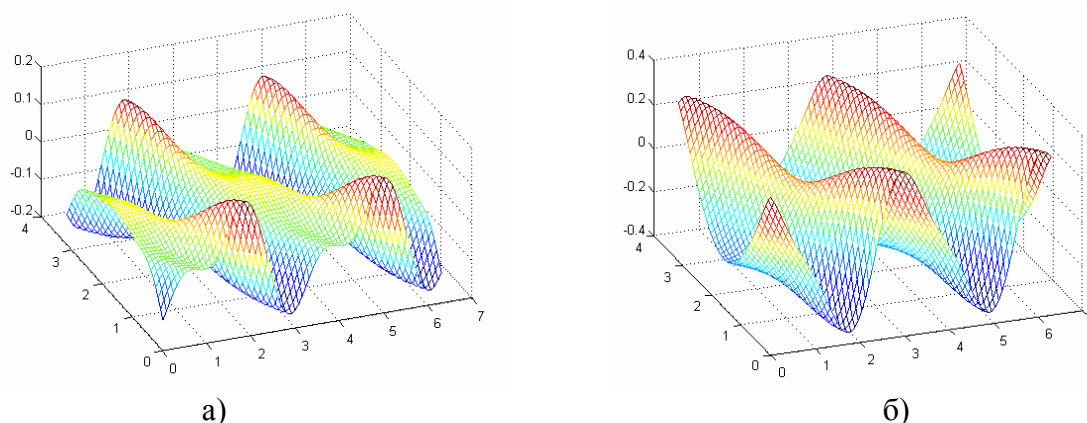


Рис. 4. Зависимости операторов а) K_{1211}^{σ} и б) K_{1211}^{ε} от углов φ_1 и Φ при $\varphi_2 = \pi/4$.

Также в работе для образца ВQ4 проделаны расчеты тензора напряжений для всевозможных эйлеровых углов при различном внешнем воздействии. Зависимости компонент тензора напряжений σ_{11} , σ_{22} , σ_{33} и их среднего арифметического $\bar{\sigma} = 1/3(\sigma_{11} + \sigma_{22} + \sigma_{33})$ от углов φ_1 и Φ при $\varphi_2 = 3\pi/4$ для оливина ВQ4 при всестороннем сжатию 400 МПа представлены на рис. 5–8.

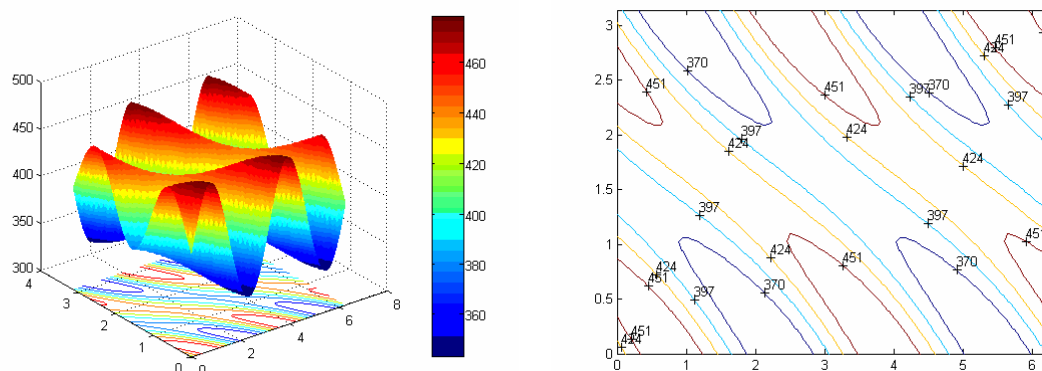


Рис. 5. Поверхность σ_{11} и ее линии уровня на опорной плоскости φ_1 , Φ .

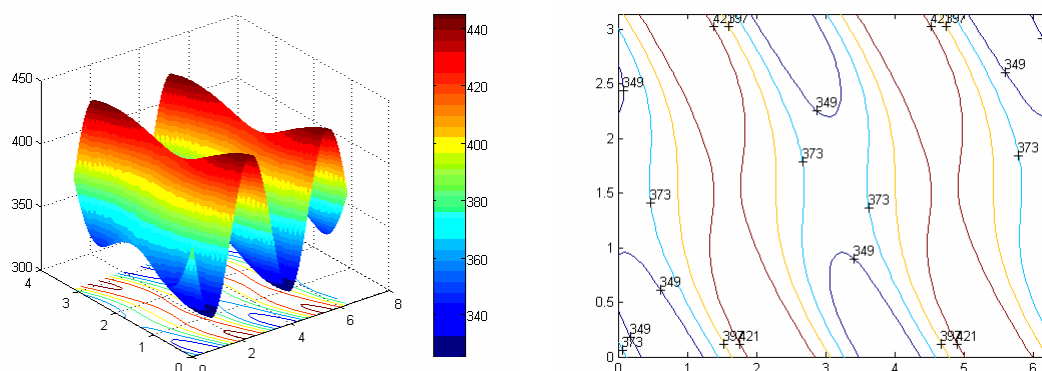
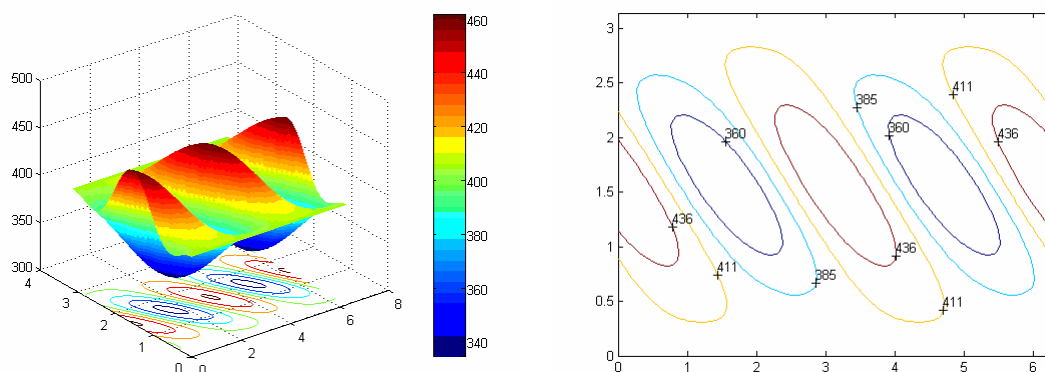
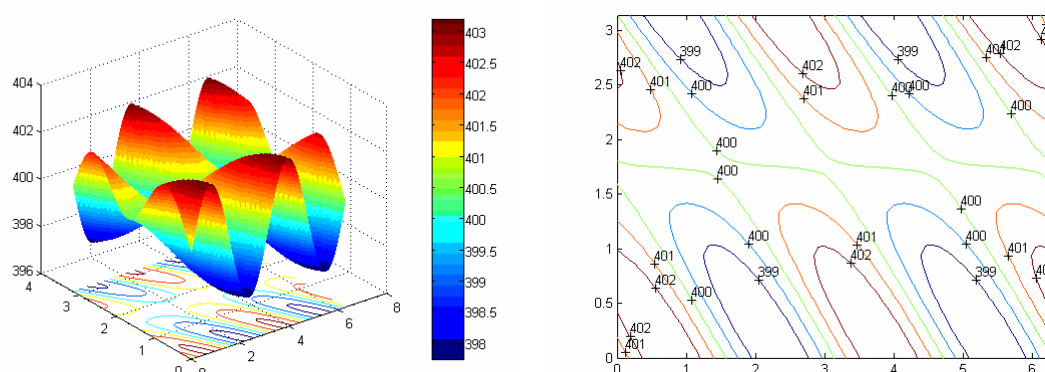


Рис. 6. Поверхность σ_{22} и ее линии уровня на опорной плоскости φ_1 , Φ .

Рис. 7. Поверхность σ_{33} и ее линии уровня на опорной плоскости φ_1 , Φ .Рис. 8. Поверхность $\bar{\sigma}$ и ее линии уровня на опорной плоскости φ_1 , Φ .

На основании проведенных расчетов ФРО, эффективных упругих постоянных, операторов концентраций напряжений и деформаций, также тензора напряжений образца BQ4 можно сделать следующие выводы. Первое, как монокристалл оливина, так и исследуемый поликристалл имеют одну и ту же орторомбическую симметрию свойств (табл. 2). Второе, значения операторов K_{1111}^{σ} и K_{1111}^{ε} близки к единице (рис. 2), а K_{2211}^{σ} , K_{2211}^{ε} и K_{1211}^{σ} , K_{1211}^{ε} — к нулю (рис. 3, 4). Это определяется тем, что данные операторы описывают связь между внешним приложенным воздействием в определенном направлении (вторая пара индексов) и внутренним напряженно-деформированным состоянием (откликом) также в определенном направлении (первая пара индексов). Третье, при моделировании внешнего воздействия (всестороннего сжатия) компоненты σ_{11} , σ_{22} , σ_{33} тензора напряжений имеют довольно большие (до 15%) отклонения от значения 400 МПа, однако их среднее $\bar{\sigma}$ незначительно отклоняется от этой величины.

Таким образом, пакет MATLAB при моделировании столь сложных систем, как поликристаллические материалы, дает большие возможности

по получению информации, ее изучению и обработке. Причем позволяет осуществить это в наглядной и доступной форме.

Литература

1. *Bunge H.-J.* Texture Analysis in Material Science.— L.: Butterworth, 1982.— 330 p.
2. *Шермергор Т. Д.* Теория упругости микroneоднородных сред.— М.: Наука. 1977.— 399 с.
3. *Вальтер К., Исаков Н. Н., Никитин А. Н., Уллемайер К., Хайнитц Й.* Исследования текстурного строения геоматериалов дифракционным методом с помощью нейтронного спектрометра высокого разрешения в лаборатории нейтронной физики им. И. М. Франка Объединенного института ядерных исследований // Изв. РАН. Физика Земли.— 1993.— №6.— С.37–44.
4. *Соболев Г. А., Никитин А. Н.* Нейтронография в геофизике // Физика элементарных частиц и атомного ядра.— 2001.— Т.32.— Вып.6.— С.1359–1404.
5. *Шермергор Т. Д., Яковлев В. Б.* Концентрация напряжений на поверхности полости в текстурированной геофизической среде // Физика Земли.— 1998.— №1.— С.81–89.

УДК 681.3.06

РАЗРАБОТКА ПАКЕТА ПРОГРАММ ДЛЯ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

Бутковский О. Я., Логунов М. Ю.

Владимирский государственный университет, Владимир

e-mail: lab432@mail.ru, olegb@vpti.vladimir.ru

В работе описан пакет программ Lab432, разработанный для анализа и построения моделей наблюдаемых временных рядов, обладающих сложным поведением. Пакет обладает единым унифицированным графическим интерфейсом пользователя (GUI), который позволяет работать с методами анализа в удобной графической оболочке, сохраняя при этом возможность работы из командной строки и написания командных файлов — сценариев.

Введение

В последние десятилетия показано, что многие процессы в физике и смежных областях науки демонстрируют сложное хаотическое поведение [1]. В связи с этим построение моделей и исследование будущего поведения таких процессов является актуальной задачей.

Особенностью подобных систем и процессов является их исключительно высокая чувствительность к малым возмущениям, в том числе к шумовым воздействиям. Малые возмущения в хаотических системах возрастают по экспоненциальному закону и довольно быстро достигают размеров аттрактора.

В настоящее время в литературе появилось много работ, в которых рассматриваются отдельные аспекты проблемы исследования и моделирования таких систем (см. библиографию в [1]). В них обсуждается оценка различных информативных, специфических характеристик функционирования этих систем (как геометрических, так и динамических), таких, как различного рода размерностей, энтропии, спектральных, статистических, моментных и прочих характеристик.

Одним из наиболее обсуждаемых вопросов является построение математической модели наблюдаемых процессов. Для этого разработано достаточно много методов реконструкции. Целью данной работы является реализация в едином пакете некоторых наработок по методам исследования и реконструкции моделей хаотических систем.

Структура пакета

При построении моделей наблюдаемых временных рядов представляется наиболее эффективным алгоритм действий, изображенный на рис. 1.

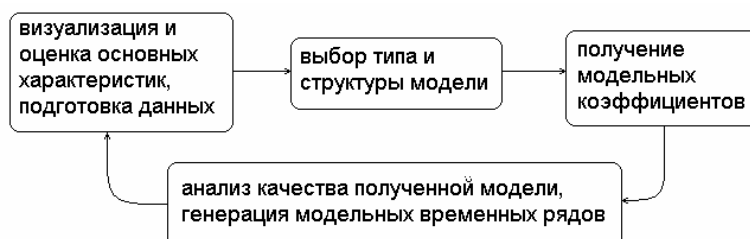


Рис. 1. Схема построения моделей временных рядов.

Разработанный пакет программ позволяет проводить весь изображенный комплекс операций и получать математические модели наблюдаемых рядов с помощью удобного графического интерфейса не прибегая к программированию «вручную». На рис.2 приведена его структурная схема.

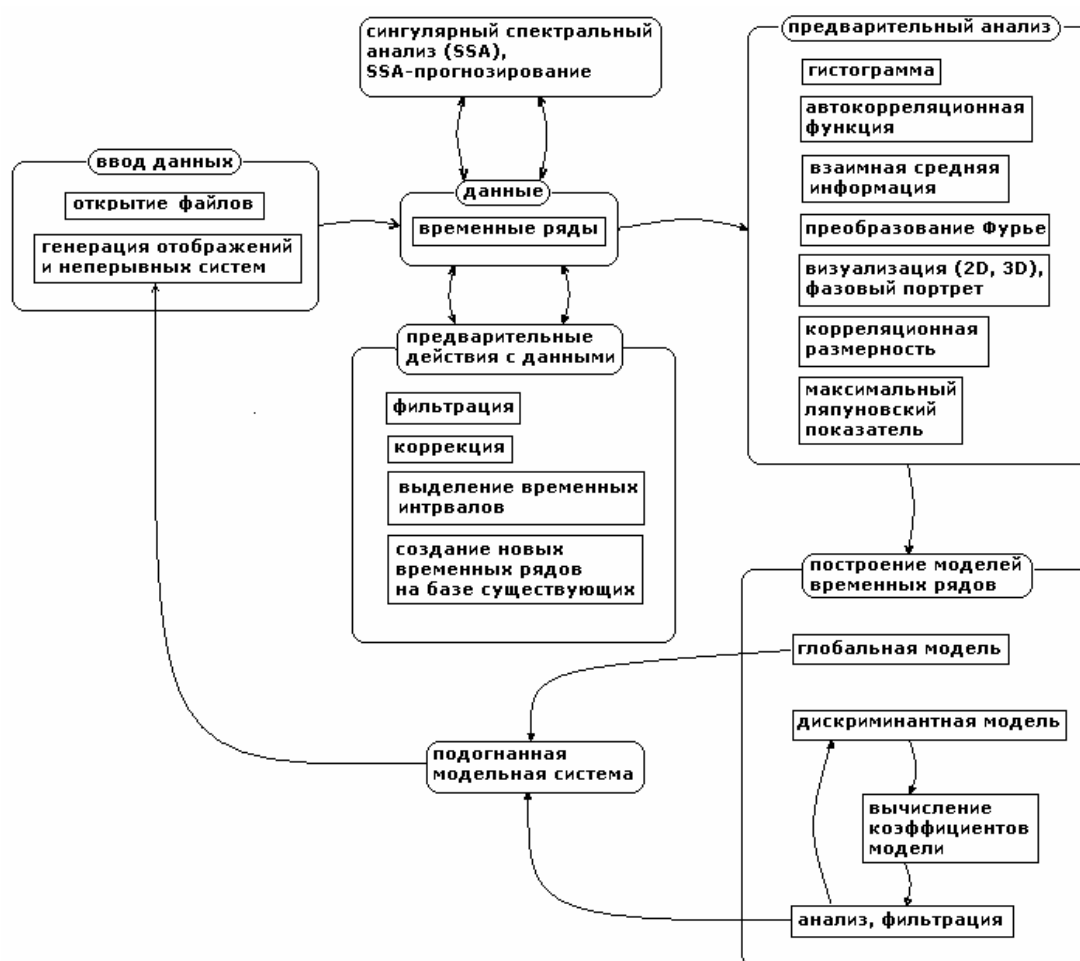


Рис. 2. Структурная схема пакета.

Программная реализация

Разработанный пакет программ функционирует в среде MATLAB 6.5. «Внутренняя» модель работы графического интерфейса пользователя, используемая системой проектирования GUIDE, в этой версии MATLAB'a отличается от предыдущих версий и не совместима с ними. Вместе с тем, она является более современной и удобной для разработчика, поэтому именно ей было отдано предпочтение при создании интерфейса пакета.

Особенность реализованных методов такова, что они слабо, или совсем не связаны друг с другом (как, например, методика глобальной реконструкции и SSA). Это определило следующий принцип построения пакета.

Основное, или управляющее окно (рис.3) связывает воедино функции пакета. Оно содержит систему меню, из которого посредством callback — откликов доступна вся функциональность пакета, также оно является «контейнером» для размещения визуальных элементов некоторых функций обработки. По причине несвязанности реализованных методов, при разработке управляющего окна было решено отказаться от внедрения в него графических элементов (кнопок, осей, списков и пр.) на стадии проектирования (т. е. хранения их в fig — файле окна). Все его графические элементы за исключением строки меню *динамические*, т. е. создаются и удаляются на шаге исполнения соответствующих функций пакета, вызываемых из меню. Также отметим, что эти функции являются *внешними* относительно основного окна.

Такой подход позволил максимально упростить структуру пакета: управляющее окно с пустым «телом» содержит в строке меню все функции пакета, которые либо формируют свои графические окна, либо временно размещают интерфейсные элементы в управляющем окне. При этом такие свойства пакета, как *целостность* и *простота* обеспечиваются унифицированным доступом к его подпрограммам, принципом «один пункт меню — одна функция обработки», независимостью отдельных частей друг от друга и использованием всего лишь двух глобальных переменных, одна из которых централизованно хранит полную структуру данных для методов пакета, а вторая — служебную информацию, нужную, в частности, для правильной установки свойств enable/disable меню управляющего окна, а также других вспомогательных целей.

Функции с графическим интерфейсом, размещаемым в отдельных окнах, создавались на основе fig — файлов в системе GUIDE «визуальными» средствами. С одной стороны, это позволило значительно ускорить процесс проектирования и сопровождения изменений интерфейса, а с другой стороны, хотя таким образом созданные функции обработки и могут быть активны одновременно, это не нарушает логики работы пакета, поскольку, как уже было сказано, методы пакета не связаны друг с другом.

В качестве плюсов подобной реализации отметим модульность и расширяемость пакета. Чтобы добавить в него пользовательские методы достаточно нескольких несложных манипуляций с меню управляющего окна и глобальной переменной со служебной информацией.

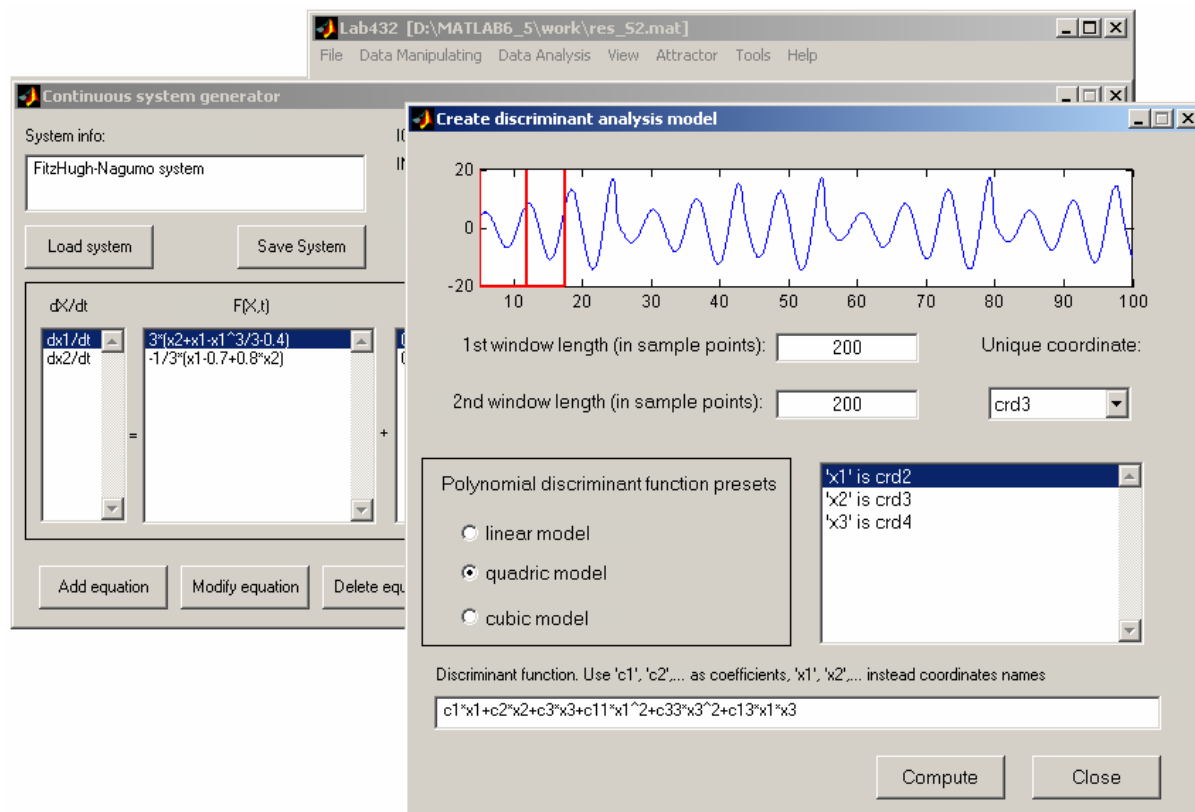


Рис. 3. Вид графических окон пакета. На заднем плане основное окно.

В настоящий момент пакет Lab432 имеет следующие функциональные возможности:

- загрузка данных из mat — файлов;
- получение данных как численное решение практически любых систем ОДУ, которые могут быть решены с помощью стандартных MATLAB — функций ODE**;
- генерация дискретных отображений различной размерности;
- наличие расширяемого банка, содержащего на данный момент более 20 наиболее распространенных систем ОДУ и отображений;
- фильтрация, коррекция, добавление и другие предварительные действия с данными;
- функции вычисления гистограмм, автокорреляции, взаимной информации, преобразования Фурье;
- проведение сингулярного спектрального анализа (SSA, в русской литературе - метод «Гусеница»);
- модифицированный дискриминантный анализ [1, 3];

- метод глобальной реконструкции [2];
- просмотр фазового портрета траектории движения;
- вычисление корреляционной размерности и максимального ляпуновского показателя временных рядов;
- возможность работы как в графическом интерфейсе пользователя, так и через файлы — сценарии.

Заключение

В работе описан пакет программ анализа и восстановления динамических уравнений из хаотических временных рядов. Включенные в него методы могут оказаться полезными для решения ряда задач, среди которых:

- идентификация хаотических систем (ХС);
- предсказание поведения ХС;
- выявление нестационарностей ХС, в т.ч. с использованием малоразмерных динамических моделей;
- выявление потенциальных бифуркаций в моделях исследуемых систем.

Описанный на примере Lab432 подход к разработке GUI — приложений совмещает в себе как простоту реализации, так и легкость сопровождения и расширения приложений. Этот подход может использоваться при создании достаточно больших пакетов программ, имеющих сложный графический интерфейс пользователя.

Литература

1. Аносов О. Л., Бутковский О. Я., Кравцов Ю. А. Восстановление динамических систем по хаотическим временным рядам (краткий обзор) // Изв. вузов. ПНД.— Т.8.— 2000.— №1.— С.29–48.
2. Безручко Б. П., Смирнов Д. А. Построение модельных отображений по хаотическим временным рядам.— Саратов: ГосУНЦ «Колледж», 2000.
3. Anosov O. L., Butkovskii O. Ya., Kravtsov Yu. A. Nonlinear chaotic systems identification from observed time series // Mathematical models and methods in applied sciences. V.7.— 1997.— N.1.— P.49–59.
4. Потемкин В. Г. Инструментальные средства MATLAB 5.X.— М.: Диалог-МИФИ, 2000.

УДК 535.361:551.463.5:551.593.5

РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ ТЕОРИИ ПЕРЕНОСА ИЗЛУЧЕНИЯ В СРЕДЕ MATLAB

Будак В. П., Козельский А. В., Савицкий Е. Н.

*Московский энергетический институт (технический университет), Москва,
e-mail: BudakVP@mpei.ru*

Теория переноса описывает широкий круг кинетических явлений в природе на основе уравнения переноса, представляющего запись закона сохранения энергии, импульса, вещества, заряда и т. п. для элементарного фазового объема статистического ансамбля локализованных в пространстве частиц. Уравнение переноса является одним из классических уравнений математической физики [1] и имеет различные модификации в зависимости от описываемого явления. Для определенности дальнейшее изложение проводится для уравнения переноса излучения (УПИ), которое было предложено в 1890 году О. Д. Хвольсоном, но, не смотря на свою более чем 100-летнюю историю, так и не имеет общего аналитического решения.

Стремительное развитие вычислительной техники, численных методов и программных средств, происходящее в последнее время позволило создать алгоритмы решения многих практически важных задач теории переноса. Однако большинство природных образований, будь-то атмосфера или морская вода, имеют взвешенные частицы с размерами, существенно превышающими длину волны, что приводит к сильно анизотропному рассеянию света в них. В этих условиях сложности решения краевых задач УПИ имеют принципиальный характер: в методе сферических гармоник (СГ) плохо обусловленная матрица приводит к сильным осцилляциям в решении, в методе Монте-Карло обратное рассеяние является маловероятным событием с большим весом. В настоящей статье показано, что векторная обработка данных и матричный анализ, лежащие в основе системы MATLAB, позволяют реализовать эффективный алгоритм решения УПИ для слоя мутной среды при произвольной степени анизотропии, оптической толщине и направлении ее освещения.

Краевая задача УПИ для слоя мутной среды с оптической толщиной τ_0 при освещении плоским мононаправленным источником (ПМ) в направлении $\hat{\mathbf{l}}_0 = \{\sqrt{1-\mu_0^2}, 0, \mu_0\}$, $\mu_0 = \cos \vartheta_0$, имеет вид

$$\begin{cases} \mu \frac{\partial L(\tau, \mu, \varphi)}{\partial \tau} + L(\tau, \mu, \varphi) = \frac{\Lambda}{4\pi} \oint x(\hat{\mathbf{l}}, \hat{\mathbf{l}}) L(\tau, \mu', \varphi') d\hat{\mathbf{l}}', \\ L(\tau, \mu, \varphi)|_{\tau=0, \mu \geq 0} = \delta(\hat{\mathbf{l}} - \hat{\mathbf{l}}_0), \quad L(\tau, \mu, \varphi)|_{\tau=\tau_0, \mu < 0} = 0; \end{cases} \quad (1)$$

где $L(\tau, \mu, \varphi) \equiv L(\tau, \hat{\mathbf{I}})$ — яркость светового поля на оптической глубине τ в направлении визирования $\hat{\mathbf{I}} = \{\sqrt{1-\mu^2} \cos \varphi, \sqrt{1-\mu^2} \sin \varphi, \mu\}$, $\mu = \cos \vartheta$; ось OZ , от которой отсчитывается зенитный угол ϑ , перпендикулярна к границе слоя; Λ — альбеда однократного рассеяния; $x(\hat{\mathbf{I}}, \hat{\mathbf{I}}')$ — индикатриса рассеяния. Здесь и далее символом крышечка над буквой будем обозначать единичные вектора.

Краевая задача адекватно описывает световые поля от Солнца в атмосфере, океане, снежном и растительном покровах.

Сущностью метода СГ является разложение искомой яркости светового поля и индикатрисы рассеяния по сферическим функциям, что приводит к бесконечной системе дифференциальных уравнений с постоянными коэффициентами [3]. Для решения систему делают конечной, полагая все коэффициенты разложения поля, начиная с номеров больших некоторого N равными нулю (P_N -приближение). Однако при освещении ПМ-источником в решении содержится сингулярность, которая определяется прямым нерассеянным излучением источника. Поэтому при любом N решение сильно сглаживается, что приводит к сильным его осцилляциям. Для устранения этого из решения вычитается прямое излучение, а краевая задача формулируется только для рассеянного излучения. Нетрудно видеть, что разложение рассеянного излучения имеет тот порядок разложения, что и индикатриса рассеяния.

Система дифференциальных уравнений с постоянными коэффициентами метода СГ имеет аналитическое решение в виде линейной комбинации экспонент с показателями, являющимися собственными числами матрицы системы [3]. Поскольку матрица имеет парные собственные числа с различными знаками, сильно отличающиеся по абсолютной величине, то решение системы становится неустойчивым уже при $N > 30$ [3]. Для устойчивого решения такой системы в [3] предложено использовать масштабное преобразование. Метод сферических гармоник в форме [3] позволяет получать решение при любом N , но для углов падения отличных от нормального это приводит к осцилляциям из-за ограничений количества азимутальных гармоник, что породило различные сглаживающие процедуры [4, 5]. Однако они оказываются не только требовательными ко времени счета, но и вносят неконтролируемый произвол в строгий алгоритм численного решения.

Для устранения указанной неустойчивости воспользуемся модификацией метода СГ [2] и будем искать решение краевой задачи в виде суммы:

$$L(\tau, \mu, \varphi) = \tilde{L}(\tau, \mu, \varphi) + L_{\text{муп}}(\tau, \hat{\mathbf{I}}, \hat{\mathbf{I}}_0), \quad (2)$$

где решение в малоугловом приближении [6] имеет вид:

$$L_{\text{МУП}}(\tau, \hat{\mathbf{l}}, \hat{\mathbf{l}}') = \sum_{k=0}^{\infty} \frac{2k+1}{4\pi} Z_k(\tau) P_k(\hat{\mathbf{\Pi}}_0) = \sum_{k=0}^{\infty} \frac{2k+1}{4\pi} Z_k(\tau) \sum_{m=0}^k (2-\delta_{m0}) Q_k^m(\mu_0) Q_k^m(\mu) \cos m\varphi, \quad (3)$$

причем $Z_k(\tau) = \exp\left\{-\frac{(1-\Lambda x_k)\tau}{\mu_0}\right\}$; $P_k(\mu)$ — присоединенные полиномы Ле-

жандра, при этом $P_k(\mu) \equiv P_k^0(\mu)$; $Q_k^m(\mu) = \sqrt{\frac{(k-m)!}{(k+m)!}}$ — перенормированные полиномы Лежандра [3], для которых справедлива нормировка

$$\int_{-1}^1 Q_k^m(\mu) Q_l^m(\mu) d\mu = \frac{2}{2k+1} \delta_{kl}. \quad (4)$$

Решение (3) удовлетворяет краевой задаче аналогичной (1), но с граничными условиями в пренебрежении обратным рассеянием

$$L_{\text{МУП}}(\tau, \hat{\mathbf{l}}, \hat{\mathbf{l}}') \Big|_{\tau=0} = \delta(\hat{\mathbf{l}} - \hat{\mathbf{l}}_0), \quad (5)$$

содержит всю анизотропную часть поля, что делает $\tilde{L}(\tau, \mu, \varphi)$ гладкой функцией.

Представим решение и индикатрису рассеяния в виде разложения в ряды по сферическим гармоникам

$$\tilde{L}(\tau, \mu, \varphi) = \sum_{l=0}^{\infty} \sum_{n=0}^l \frac{2l+1}{4\pi} C_l^n(\tau) Q_l^n(\mu) e^{-in\varphi}, \quad x(\hat{\mathbf{l}}, \hat{\mathbf{l}}') = \sum_{l=0}^{\infty} (2l+1) x_l P_l(\hat{\mathbf{\Pi}}'). \quad (6)$$

Представление индикатрисы рассеяния в (6) с учетом теоремы сложения для полиномов Лежандра [3], можно переписать в виде

$$x(\hat{\mathbf{l}}, \hat{\mathbf{l}}') = 4\pi \sum_{l=0}^{\infty} \sum_{n=-l}^l x_l Y_l^n(\hat{\mathbf{l}}) \overline{Y_l^n(\hat{\mathbf{l}}')}, \quad (1)$$

где $Y_l^n(\hat{\mathbf{l}}) = \sqrt{\frac{2l+1}{2}} Q_l^n(\mu) e^{-in\varphi}$ — сферические функции, полную образующие ортонормированную систему на сфере; чертой сверху обозначено комплексное сопряжение.

Подставим выражения (3) и (6) в УПИ краевой задачи (1) и учтем (7). После чего умножим полученное уравнение на $Q_l^n(\mu) e^{-in\varphi}$ и проинтегрируем по всей области изменения μ, φ . Тогда с учетом ортогональности сферических функций получим бесконечную систему связанных дифференциальных уравнений метода сферических гармоник

$$\begin{aligned} & \frac{1}{2k+1} \frac{d}{d\tau} \left[\sqrt{(k-m+1)(k+m+1)} C_{k+1}^m + \sqrt{(k-m)(k+m)} C_{k-1}^m \right] + (1-\Lambda x_k) C_{k+1}^m(\tau) = \\ & = -\frac{1}{2k+1} \frac{1}{\mu_0} \frac{d}{d\tau} \left[\sqrt{(k-m+1)(k+m+1)} (1-\Lambda x_{k+1}) Q_{k+1}^m(\mu_0) Z_{k+1} \right. \\ & \quad \left. + \sqrt{(k-m)(k+m)} (1-\Lambda x_{k-1}) Q_{k-1}^m(\mu_0) Z_{k-1} \right] - (1-\Lambda x_k) Q_k^m(\mu_0) Z_k(\tau) \end{aligned} \quad (8)$$

Решение бесконечной системы (8) невозможно, поэтому ее делают конечной, положив $C_k^m(\tau) \equiv 0, \forall k \geq N$. Однако при этом возникает известная проблема при решении краевой задачи: приближенное решение не может удовлетворить точным граничным условиям, которые должны быть заме-

нены приближенными. Наилучшим вариантом являются граничные условия в форме Маршака [7], выражающие закон сохранения энергии в форме равенства потоков излучения на границе раздела среды

$$\forall j \in \frac{m+1}{2}, \frac{N+1}{2}, \forall m \in \overline{0, N}: \quad (9)$$

$$\int_{\Omega_+} L(0, \hat{\mathbf{l}}) Y_{2j-1}^m(\hat{\mathbf{l}}) d\hat{\mathbf{l}} = 0, \quad \int_{\Omega_-} L(\tau_0, \hat{\mathbf{l}}) Y_{2j-1}^m(\hat{\mathbf{l}}) d\hat{\mathbf{l}} = - \int_{\Omega_-} L_{\text{МВП}}(\tau_0, \hat{\mathbf{l}}, \hat{\mathbf{l}}_0) Y_{2j-1}^m(\hat{\mathbf{l}}) d\hat{\mathbf{l}}$$

где $\Omega_{\pm} = \{\hat{\mathbf{l}}: (\hat{\mathbf{l}}, \hat{\mathbf{z}})_{\pm} > 0\}$, $\hat{\mathbf{z}}$ — единичный вектор нормали к границе.

Конечная система дифференциальных уравнений (8) с постоянными коэффициентами имеет в решении $N+1-m$ константу, которые определяются из системы граничных условий (9). Практически оказывается лучше нечетные N для четных m и наоборот [3]. Поэтому далее предполагается, что система (8) имеет $N_m = N + \delta_m$ уравнений, где

$$\delta_m = \begin{cases} 1, & \text{при четном } m; \\ 1, & \text{при нечетном } m; \end{cases}$$

С учетом разложений (3) и (6) граничные условия (9) приобретают вид

$$C_{2j+1}^m(0) + \sum_{l=m/2-1}^{\frac{N_m-1}{2}} G_{jl}^m C_{2l-2}^m(0) = 0, \quad C_{2j+1}^m(\tau_0) - \sum_{l=m/2-1}^{\frac{N_m-1}{2}} G_{jl}^m C_{2l-2}^m(\tau_0) = -Y_j^m(\tau_0, \mu_0), \quad (10)$$

где

$$Y_j^m(\tau_0, \mu_0) = \sum_{k=0}^{\infty} (2k+1) Z_k(\tau_0) Q_k^m(\mu_0) \int_{-1}^0 Q_k^m(\mu) Q_{2j-1}^m(\mu) d\mu =$$

$$= Z_{2j-1}(\tau_0) Q_{2j-1}^m(\mu_0) - \sum_{l=0}^{\infty} G_{jl}^m Z_{2l-2}(\tau_0) Q_{2l-2}^m(\mu_0);$$

— коэффициенты $G_{jl} = (4l-3) \int_0^1 Q_{2j-1}^m(\mu) Q_{2l-2}^m(\mu) d\mu$ введены в [8], где для них получены простые рекуррентные соотношения.

Для аналитического удобства конечной системе (8) можно придать матричный вид аналогично [3, 8]

$$\bar{\mathbf{A}}^m \frac{d}{d\tau} \bar{\mathbf{C}}(\tau) + \bar{\mathbf{D}} \bar{\mathbf{C}}(\tau) = (\bar{\mathbf{A}}^m / \mu_0 - \bar{\mathbf{I}}) \bar{\mathbf{D}} \bar{\mathbf{Q}}^m \bar{\mathbf{Z}}(\tau) + a_{N+1}^m \bar{\mathbf{Z}}_{N+1}(\tau), \quad (11)$$

где

$$(\bar{\mathbf{A}}^m)_{i,i+1} = \frac{\sqrt{(i-m)(i+m)}}{2i-1}, \quad (\bar{\mathbf{A}}^m)_{i,i-1} = \frac{\sqrt{(i-m-1)(i+m-1)}}{2i-1},$$

$$\bar{\mathbf{C}} = \{C_{i-1}^m(\tau)\}, \quad \bar{\mathbf{D}} = \text{Diag}\{(1 - \Lambda x_{i-1}) / \mu_0\}, \quad \bar{\mathbf{Q}}^m = \text{Diag}\{Q_{i-1}^m(\mu_0)\}, \quad \bar{\mathbf{Z}} = \{Z_{i-1}(\tau)\},$$

$$a_{N+1}^m = \frac{\sqrt{(N+1-m)(N+1+m)}}{2N+1} \frac{1 - \Lambda x_{N+1}}{\mu_0} Q_{N+1}^m(\mu_0), \quad \bar{\mathbf{Z}}_{N+1} = \left\{ \underbrace{0 \dots 0}_N, Z_{N+1} \right\}.$$

Здесь и далее двойной стрелкой над символом будем обозначать матрицу, одинарной правой — вектор-столбец, одинарной левой — вектор-строку. Для упрощения записи там, где очевидно, будем опускать азимутальный индекс m .

В P_N -приближении метода СГ в правую часть уравнения входит с учетом выражения (8) коэффициент $Z_{N+1}(\tau)$, что матрица \bar{A}^m непосредственно не учитывает, поскольку формируется в предположении $C_{N+1}^m(\tau) = 0$. Это определяет последний член в правой части уравнения (11).

Решение системы дифференциальных уравнений (11) можно записать в форме выражения

$$-\bar{C}(0) + e^{-\bar{B}\tau_0} = \frac{1}{\mu_0} \int_0^{\tau_0} e^{-\bar{B}t} (\bar{I} - \mu_0 \bar{A}^{-1}) \bar{D} \bar{Q} \bar{Z}(t) dt + a_{N+1}^m \int_0^{\tau_0} e^{-\bar{B}t} \bar{A}^{-1} \bar{Z}_{N+1}(t) dt, \quad (12)$$

где $\bar{B} = \bar{A}^{-1} \bar{D}$.

Матричная экспонента представима в виде

$$e^{-\bar{B}t} = \bar{U} e^{-\bar{\Gamma}t} \bar{U}^{-1}, \quad (13)$$

где $\bar{\Gamma}$ — диагональная матрица собственных значений, а \bar{U} — матрица соответствующих собственных векторов матрицы \bar{B} .

С учетом последнего выражения (13) интеграл, входящий в уравнение (12), можно переписать в форме следующего выражения

$$\bar{J} = \frac{1}{\mu_0} \int_0^{\tau_0} e^{-\bar{\Gamma}t} \bar{T} \bar{Z}(t) dt + a_{N+1}^m \int_0^{\tau_0} e^{-\bar{\Gamma}t} \bar{Z}_{N+1}(t) dt \bar{U}^{-1} \bar{A}_{N+1}^{-1}, \quad (14)$$

где $T = \bar{U}^{-1} (\bar{I} - \mu_0 \bar{A}^{-1}) \bar{D} \bar{Q}$.

Поскольку матрица $e^{-\bar{\Gamma}t}$ диагональная, а $\bar{Z}(t) = \{\exp(-(1 - \Lambda x_{i-1})t / \mu_0)\}$ — вектор столбец, то все элементы столбца \bar{J} легко вычисляются

$$\bar{J} = \left[\sum_{j=1}^{N+1} (\bar{T})_{ij} \frac{\{1 - \exp(-\Gamma_{ii} \tau_0 - (1 - \Lambda x_{j-1}) \tau_0 / \mu_0)\}}{\Gamma_{ii} + (1 - \Lambda x_{j-1})} \right] a_{N+1}^m \left(\bar{\Gamma} + b_{N+1} \bar{I} \right)^{-1} \left(\bar{I} - \exp(-(\bar{\Gamma} + b_{N+1} \bar{I}) \tau_0 / \mu_0) \right) \bar{U}^{-1} \bar{A}_{N+1}^{-1}, \quad (15)$$

а выражение (12) приобретает форму

$$-\bar{C}(0) + \bar{U} e^{-\bar{\Gamma}\tau_0} \bar{U}^{-1} \bar{C}(\tau_0) = \bar{U} \bar{J}(\tau_0). \quad (16)$$

Последнее соотношение (16) представляет собой систему из $(N_m - m)$ линейных алгебраических уравнений с $2(N_m - m)$ неизвестными. Недостающие $(N_m - m)$ уравнений дают граничные условия (10), которым так же придадим аналогично [8] матричный вид:

$$\bar{C}_{odd}(0) + \bar{G} \bar{C}_{even}(0) = 0, \quad \bar{C}_{odd}(\tau_0) - \bar{G} \bar{C}_{even}(\tau_0) = -\bar{Y}(\tau_0, \mu_0), \quad (17)$$

где $(\bar{G})_{jl} = G_{jl}$, $\bar{Y}(\tau_0, \mu_0) = \{Y_j^m(\tau_0, \mu_0)\}$, а индексы *odd-even* обозначают столбцы, составленные соответственно из нечетных или четных элементов \bar{C} .

Устранить проблему разделения в граничных условиях четных и нечетных элементов можно, введя матрицу, сортирующую столбцы на четные или нечетные элементы [8]

$$\bar{P} \bar{C} \equiv \begin{bmatrix} \bar{C}_{odd} \\ \bar{C}_{even} \end{bmatrix},$$

что позволит граничным условиям (17) придать следующую форму

$$\begin{bmatrix} 1 & \tilde{G} \end{bmatrix} \tilde{P}\tilde{C}(0) = \tilde{0}, \quad \begin{bmatrix} 1 & -\tilde{G} \end{bmatrix} \tilde{P}\tilde{C}(\tau_0) = \begin{bmatrix} \tilde{0} \\ -\tilde{Y}(\tau_0, \mu_0) \end{bmatrix}. \quad (18)$$

Тогда выражения (16) и (18) образуют замкнутую систему линейных алгебраических уравнений, из решения которой можно определить искомые коэффициенты. Обусловленность матрицы системы быстро ухудшается с ростом толщины слоя. Для устранения этого явления следует воспользоваться масштабным преобразованием [3]. Предположим, что собственные значения матрицы \tilde{B} отсортированы по возрастанию:

$$\tilde{\Gamma} = \begin{bmatrix} \tilde{\Gamma}_- & 0 \\ 0 & \tilde{\Gamma}_+ \end{bmatrix}, \quad \tilde{\Gamma}_+ = -\tilde{\Gamma}_- = \text{Diag}\{\gamma_i\}, \quad \gamma_i < \gamma_{i+1},$$

что позволяет записать $e^{-\tilde{\Gamma}\tau} = \begin{bmatrix} e^{-\tilde{\Gamma}_-\tau} & 0 \\ 0 & e^{-\tilde{\Gamma}_+\tau} \end{bmatrix}$.

Тогда после умножения на матрицу $\tilde{S}\tilde{U}^{-1}$ уравнение (16) можно привести к виду

$$-\tilde{S}\tilde{U}^{-1}\tilde{C}(0) + \tilde{H}\tilde{U}^{-1}\tilde{C}(\tau_0) = \tilde{S}\tilde{J}(\tau_0), \quad (19)$$

где $S = \begin{bmatrix} 0 & \tilde{I} \\ e^{-\tilde{\Gamma}_+\tau_0} & 0 \end{bmatrix}$, $\tilde{H} = \tilde{S}e^{-\tilde{\Gamma}\tau_0} = \begin{bmatrix} 0 & \tilde{I} \\ e^{-\tilde{\Gamma}_+\tau_0} & 0 \end{bmatrix} \begin{bmatrix} e^{-\tilde{\Gamma}_-\tau_0} & 0 \\ 0 & e^{-\tilde{\Gamma}_+\tau_0} \end{bmatrix} = \begin{bmatrix} 0 & e^{-\tilde{\Gamma}_+\tau_0} \\ \tilde{I} & 0 \end{bmatrix}$

Нетрудно видеть, что в уравнение (19) входят экспоненты только с отрицательными показателями и уравнение сохраняет устойчивость при любых значениях N и τ_0 . Из решения системы линейных уравнений (18) и (19) определяются искомые коэффициенты разложения тела яркости на верхней и нижней границах среды.

Матричный язык системы MATLAB является на удивление адекватным описанному алгоритму решения, что позволяет, с одной стороны, очень компактно записать его код, с другой стороны, максимально векторизовать и оптимизировать алгоритм вычислений. Практически полный текст программы приведен на рис. 1. В листинге программы на рис.1 через идентификаторы обозначены переменные: $N1=Nm-m+1$, $N2=N1/2$, Nm — количество гармоник N_m , m — азимутальный индекс, по которому в программе организован цикл, V — альбедо однократного рассеяния Λ , x_k — коэффициенты разложения индикатрисы рассеяния по полиномам Лежандра x_k , Tau0 — оптическая толщина слоя τ_0 . Время счета программы на компьютере с процессором Intel Pentium IV 2.4 ГГц не при каких входных данных не превышало 20 с.

```
% Формирование матриц системы
I = eye(N2); % Единичная матрица
O = zeros(N2); % Нулевая матрица
K = [m+1: Nm];
A = diag(sqrt(K.^2-m^2)./(2*K-1),1) ...
    + diag(sqrt(K.^2-m^2)./(2*K+1),-1);
xV = 1.0-V*xk(m+1:Nm+1);
D = diag(xV);
```

```

AI = inv(A);
B = AI * D;
% Матричная экспонента
[U, b] = eig(B);
% Сортировка собственных значений
[e, ie] = sort(diag(b));
UI = inv(U(:,ie));
% Матрицы масштабного преобразования
S = [O I; diag(exp(e(1:N2)*Tau0)) O] * UI;
H = [O diag(exp(-e(N2+1:N1)*Tau0)); I O] * UI;
... Формирование вектор-столбца J
... Формирование матрицы граничных условий G
... Формирование матрицы сортировки P
% Решение системы линейных уравнений
AA = [[[I G]*P O O; O O [I -G]*P]; -H S];
BB = [zeros(N2,1); -Ybb; Zk];
f = AA \ BB;
% Определение коэффициентов разложения
% На верхней границе слоя
YTk(m+1,1:N1) = f(1:N1)';
% На нижней границе слоя
YBk(m+1,1:N1) = f(N1+1:2*N1)';

```

Рис. 1. Листинг программы.

Для определения связи оптических характеристик среды с микрофизическими (функция распределения частиц по размерам и показатель преломления частиц) программа была дополнена результатами теории Ми в форме [9], представляющей разложение индикатрисы рассеяния непосредственно по сферическим функциям.

Изложенный алгоритм относится к случаю $\Lambda \neq 1$. При $\Lambda = 1$ матрица \tilde{B} становится вырожденной и представление (13) невозможно. Однако выражение (13) можно заменить жордановой формой. Это ничего, по сути, не изменяет в изложенном алгоритме, только усложняя изложение. Алгоритм при $\Lambda = 1$ строиться полностью по аналогии с [3] при учете изменения правой части (19) в соответствии с (15). Не представляет труда обобщение метода решения на стратифицированный слой: слой разбивается на совокупность однородных слоев, для каждого из которых формируется общая система, аналогичная описанной.

Литература

1. Владимиров В. С. Уравнения математической физики.— М.: Наука, 1971.— 512 с.
2. Будак В. П., Козельский А. В., Савицкий Е. Н. Улучшение сходимости метода сферических гармоник при сильно анизотропном рассеянии // Оптика атмосферы и океана.— 2004.— Т.17.— С.36–41.

3. *Karp A. H., Greenstadt J., Fillmore J. A.* Radiative transfer through an arbitrary thick, scattering atmosphere // JQSRT.— 1980.— V.24.— P.391–406.
4. *Karp A. H., Petrack S.* On the spherical harmonics and discrete ordinate methods for azimuth-dependent intensity calculations // JQSRT.— 1983.— V.30.— P.351–356.
5. *Muldashev T. Z., Sultangazin A. I., Lyapustin A. I.* Spherical harmonics method in the problem of radiative transfer in the atmosphere-surface system // JQSRT.— 1999.— V.61.— P.393–404.
6. *Goudsmit S., Saunderson J. L.* Multiple Scattering of Electrons // Phys. Rev.— Part I.: 1940.— V.57.— P.24–29; Part II.: V.58.— P.36–42.
7. *Marshak R. E.* Note on the spherical harmonics method as applied to the Milne problem for a sphere // Phys.Rev.— 1947.— V.71.— P.443–446.
8. *Dave J. V.* A direct solution of the spherical harmonics approximation to the radiative transfer equation for an arbitrary solar elevation // Journ. Atmosph. Sci.— 1975.— V.32.— P.790–798.
9. *de Rooij W. A., van der Star C. C. A. H.* Expansion of Mie scattering matrices in generalized functions // Astron. Astrophys.— 1984.— V.131.— P.237–248.

УДК 536.46:519.8

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ РЕЖИМОВ КАТАЛИТИЧЕСКОГО ОКИСЛЕНИЯ НА ОСНОВЕ СИСТЕМЫ MATLAB

Вержбицкая И. С.

Казахский национальный университет им. аль-Фараби, Алматы, Казахстан
e-mail: i_verb@mail.ru

Теоретически исследуется окисление этана в химическом проточном реакторе с неподвижным слоем сферических частиц катализатора. Предполагается, что катализатор и газ имеют одинаковую температуру, диффузия в частицах катализатора отсутствует, тепло-и массоперенос характеризуется эффективными коэффициентами диффузии и теплопроводности. Рассматривается необратимая экзотермическая химическая реакция в газовой фазе типа $A \rightarrow B$. Процесс описывается двумерной квазигомогенной моделью, учитывающей диффузию тепла, этана и водяного пара [1] и в безразмерных переменных имеет вид:

$$\tau > 0: \quad 0 < S_1 < 1, \quad 0 < S_2 < 1,$$

$$F(\varepsilon) \frac{\partial \theta}{\partial \tau} + Pe \frac{\partial \theta}{\partial S_1} = n \frac{\lambda_r^{\text{ef}}}{\lambda_G} \frac{1}{S_2} \frac{\partial}{\partial S_2} \left(S_2 \frac{\partial \theta}{\partial S_2} \right) + \frac{\lambda_x^{\text{ef}}}{\lambda_G} \frac{\partial^2 \theta}{\partial S_1^2} + q \cdot Da \cdot C_1 \cdot \Gamma(\theta); \quad (1)$$

$$\frac{\partial C_1}{\partial \tau} + Pe \frac{\partial C_1}{\partial S_1} = n \frac{1}{S_2} \frac{\partial}{\partial S_2} \left(S_2 \cdot Le_r \frac{\partial C_1}{\partial S_2} \right) + Le_x \frac{\partial^2 C_1}{\partial S_1^2} - Da \cdot C_1 \cdot \Gamma(\theta); \quad (2)$$

$$\frac{\partial C_2}{\partial \tau} + Pe \frac{\partial C_2}{\partial S_1} = n \frac{1}{S_2} \frac{\partial}{\partial S_2} \left(S_2 \cdot Le_r \frac{\partial C_2}{\partial S_2} \right) + Le_x \frac{\partial^2 C_2}{\partial S_1^2} + 3Da \cdot C_2 \cdot \Gamma(\theta), \quad (3)$$

с начальными и граничными условиями:

$$\tau = 0: \quad 0 \leq S_1 \leq 1, \quad 0 \leq S_2 \leq 1, \quad \theta = \theta_H, \quad C_1 = C_{1H}, \quad C_2 = C_{2H}; \quad (4)$$

$$\tau > 0: \quad 0 \leq S_2 \leq 1, \quad S_1 = 0, \quad \frac{\lambda_x^{\text{ef}}}{\lambda_G} \frac{\partial \theta}{\partial S_1} = Pe(\theta - \theta_{\text{BX}}); \quad C_1 = C_{1\text{BX}}; \quad C_2 = C_{2\text{BX}}; \quad (5)$$

$$S_1 = 1, \quad \frac{\partial \theta}{\partial S_1} = 0, \quad \frac{\partial C_1}{\partial S_1} = 0; \quad \frac{\partial C_2}{\partial S_1} = 0; \quad (6)$$

$$0 \leq S_1 \leq 1, \quad S_2 = 0, \quad \frac{\partial u}{\partial S_2} = 0, \quad \frac{\partial C_1}{\partial S_2} = 0; \quad \frac{\partial C_2}{\partial S_2} = 0; \quad (7)$$

$$S_2 = 1, \quad -\frac{\partial \theta}{\partial S_2} = Bi_T(\theta - \theta_{\infty}); \quad -\frac{\partial C_1}{\partial S_2} = Bi_D(C_1 - C_{1\infty}); \quad -\frac{\partial C_2}{\partial S_2} = Bi_D(C_2 - C_{2\infty}). \quad (8)$$

где

$$\begin{aligned}\theta &= \frac{E}{RT_{\text{BX}}^2} (T - T_{\text{BX}}); S_1 = \frac{x}{l}; S_2 = \frac{r}{r_t}; \tau = \frac{a_h t}{l^2}; q = \frac{\tilde{c}_G \cdot Q \cdot E}{\dot{r}_G \cdot \dot{E} \cdot R \cdot T_{\text{BX}}^2}; Da = \frac{l^2 k_0}{a_h \tilde{c}_G}; Le_x = \frac{D_x}{a_h}; \\ Le_r(r) &= \frac{D_r}{a_h}; Pe = \frac{u l}{a_h}; a_h = \frac{\lambda_G}{\rho_G \cdot c_{p_G}}; Bi_T = \frac{\alpha_h r_t}{\lambda_r^{ef}}; Bi_D = \frac{\alpha_D r_t}{D_r}; n = \frac{l^2}{r_t^2}; \\ F(\varepsilon) &= \varepsilon(r) + (1 - \varepsilon(r)) \cdot \gamma; \vartheta_{\text{BX}} = \frac{RT_{\text{BX}}}{E}; \\ \Gamma(\theta) &= f(\theta) \cdot \exp(-1/\vartheta_{\text{BX}}) \cdot \exp\left(\frac{\theta}{1 + \vartheta_{\text{BX}} \theta}\right) f_1; f(\theta) = \frac{1 - \varepsilon(r)}{1 - \bar{\varepsilon}} \left(1 + \frac{|\theta_{\infty} - \theta|}{\theta_{\infty} + \frac{1}{\vartheta_{\text{BX}}}}\right).\end{aligned}$$

Здесь t — время; x — продольная координата; r — радиальная координата; r_b, l — радиус и длина реактора; T — температура; C_1, C_2 — концентрация реагента и водяного пара; \tilde{c}_G — концентрация реагента на входе в реактор; ρ_G — плотность газа; ρ_K — плотность катализатора; C_{p_G}, c_K — теплоемкость газа и катализатора; ε — пористость слоя катализатора; $F(\varepsilon)$ — функция пористости; u — скорость течения смеси; λ — коэффициент теплопроводности; D — коэффициент диффузии; Q, k_0, E — тепловой эффект, предэкспоненциальный множитель, энергия активации химической реакции; R — универсальная газовая постоянная; α_h, α_D — коэффициенты тепло- и массообмена; a_h — коэффициент температуропроводности; $\Gamma(\theta)$ — скорость реакции.

Индексы «G» и «K» относятся к газовой и твердой фазам, «r» и «x» — к радиальной и продольной составляющим, «ef» — эффективному значению, «вх», «н» и «∞» — к входу в реактор, началу процесса и окружающей среде.

Обезразмеривание позволило выделить следующие параметры, определяющие ход процесса: q — тепло реакции; $Le_x, Le_r, Pe, Da, Bi_T, Bi_D$ — числа Льюиса, Пекле, Дамкелера, тепловое и диффузионное Био; $\vartheta_{\text{вх}}$ — температура на входе в реактор; $\frac{\lambda_x^{ef}}{\lambda_G}, \frac{\lambda_r^{ef}}{\lambda_G}$ — отношения эффективных акси-

ального и радиального коэффициентов теплопроводности к коэффициенту теплопроводности газа; γ — отношение объемных теплоемкостей катализатора и газа; n — геометрический размер реактора; θ_n, C_{1n}, C_{2n} — начальная температура смеси и концентрации реагента и водяного пара; $\theta_{\text{вх}}, C_{1\text{вх}}, C_{2\text{вх}}, \theta_{\infty}, C_{1\infty}, C_{2\infty}$ — температура смеси и концентрации реагента и водяного пара на входе в реактор и в окружающей среде.

При решении задачи (1)–(8) использовалась явная конечно-разностная схема. Расчеты выполнялись при фиксированных шагах по пространственным координатам, а шаг по времени «автоматически» уменьшался с ростом скорости реакции, что обеспечивало устойчивость вычисления на каждом временном слое. Выбор явной конечно-разностной схемы связан со спецификой исследуемой задачи. Уловить колебания, имеющие физическую природу, можно, когда шаг по времени хотя бы на порядок меньше характерного времени физического процесса (период колебаний). В противном случае колебания останутся незамеченными. Поэтому допускаемое при использовании неявной схемы увеличение шага по времени могло бы привести к потере истинной картины физического явления.

Алгоритм решения задачи реализован в операционной среде MATLAB. При этом использовались основные отличительные свойства системы MATLAB, позволившие значительно сократить время вычислений, а именно векторизация циклов, т. е. замена циклических вычислений матричными операциями; широкое применение встроенных функций системы, использование глобальных переменных, графических средств и др.

Ранее, в [2] методами качественной теории дифференциальных уравнений на нуль-мерном аналоге исходной двумерной задачи (1)–(8) была исследована устойчивость стационарных состояний и выявлены параметрические области возможных режимов окисления. В данной работе рассчитывались и анализировались пространственно-временные распределения температуры и концентрации газовой смеси для значений параметров из работы [2].

На рис. 1, а, б показан численно реализованный режим затухающих колебаний из области моностационарности. Как видно из рис. 1, а рассчитанная численно кривая изменения температуры со временем располагается вблизи стационарного состояния, полученного из бифуркационной диаграммы [2]. На рис. 1, б показано изменение со временем концентрации этана (кривая 1) и водяного пара (кривая 2) для этого режима.

Рис. 2, а, б иллюстрирует типичные для экзотермической каталитической реакции радиальные распределения температуры и концентрации реагента, полученные в результате вычислений в различных сечениях по длине реактора: на его входе (кривые 1), выходе (кривые 3) и сечении $S_I=0,3$, соответствующему области с максимальной температурой (кривые 2). Как видно из рис. 2, б, конверсия вдоль оси реактора выше, чем в какой-либо другой радиальной позиции (см. кривые 2, 3), т. к. температура здесь максимальна (см. кривые 2, 3 рис. 2, а, и рис. 2, с). Экзотермическая природа реакции вначале ведет к возрастанию температуры, но из-за того, что на последних стадиях реакции радиальный перенос тепла к стенке и окружающей среде превышает тепло, выделяемое химической реакцией, температура устойчиво уменьшается. Картина изотерм для этого режима

показано на рис. 2, с (цифры на кривых соответствуют значениям температуры θ). Видно, что область с максимальной температурой располагается в центре реактора, вблизи его входа. Минимальная температура наблюдается на входе в реактор и на его стенке.

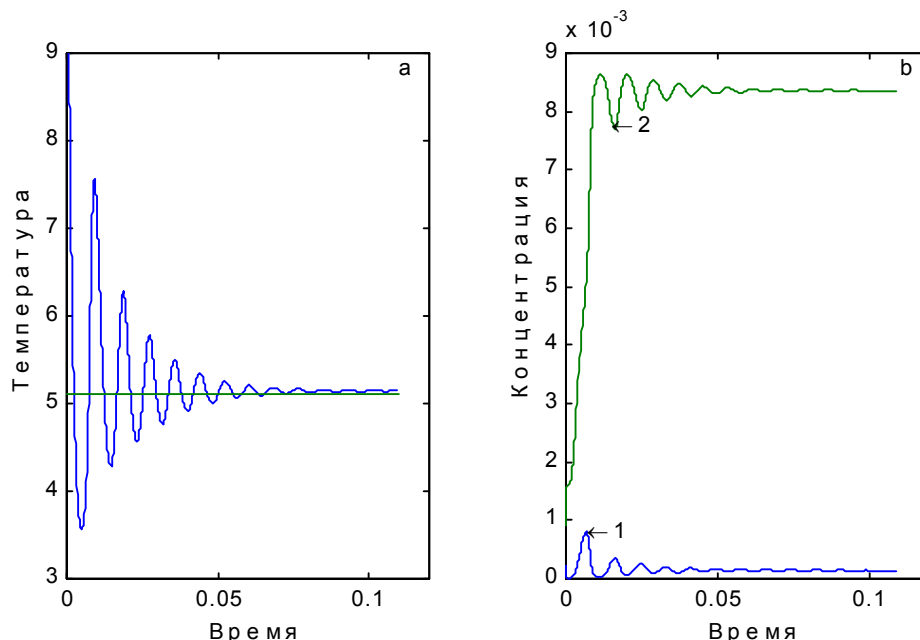


Рис. 1. Изменение со временем температуры газовой смеси (а), концентрации этана (кривая 1) и концентрации водяного пара (кривая 2) (b): $Pe=121,48$; $Bi_I=2,97$; $Bi_D=0$; $Da=3,12 \cdot 10^6$; $Le_x=3,58$; $\theta_H=8,85$; $\theta_{BX}=0$; $C_{IH}=0,001$; $C_{2H}=0,001$; $C_{1BX}=0,0004$; $C_{2BX}=0,0004$; $\varepsilon=0,4$; $n=8,29$.

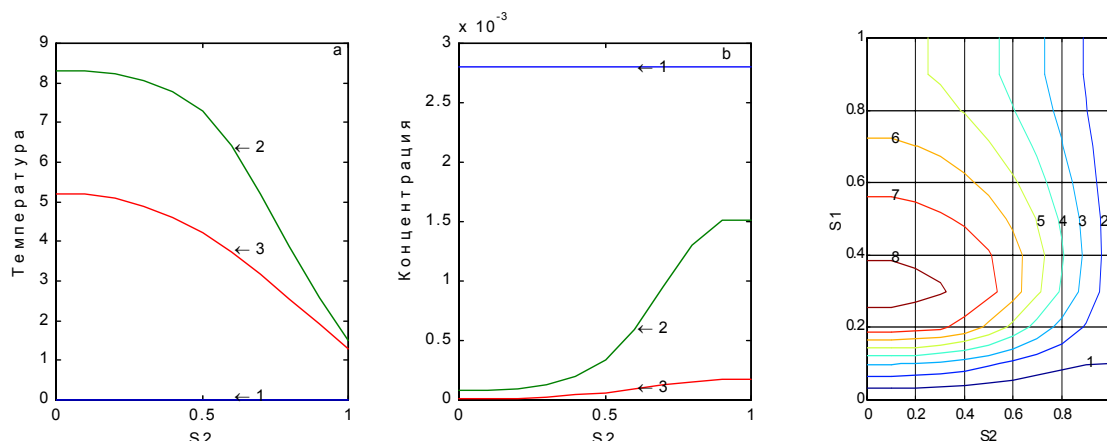


Рис.2. Стационарные радиальные профили температуры (а), концентрации этана (b) и распределение изотерм в реакторе (с): $\tau=0,1$; $S_I=0$ (1); 0,3 (2); 1 (3) (а,b). Параметры расчетов соответствуют режиму затухающих колебаний на рис.1.

Как известно, экзотермические химические реакции в неизотермическом реакторе чувствительны к изменению температуры стенок. Небольшое изменение температуры хладагента может при определенных обстоя-

тельствах быть причиной возникновения нежелательных горячих пятен или областей высокой температуры внутри реактора. Рассмотрим влияние изменения температуры стенок реактора (θ_w) на режим затухающих колебаний (рис.1). В результате численных экспериментов было получено, что, как и следовало ожидать, увеличение температуры хладагента приводит к установлению стационарного режима с более высокой максимальной температурой. Кроме того, изменение температуры хладагента влияет на характер режима окисления. Проводя расчеты с различным параметром θ_w , можно наблюдать эволюцию фазового портрета системы.

Рассмотрим случай, когда температура смеси на входе в реактор ниже начальной температуры смеси $\theta_{вх} < \theta_n$. В этом случае, при температуре стенок реактора, превышающей начальную температуру смеси ($\theta_w > \theta_n$) в системе наблюдаются асимптотически устойчивые режимы. Например, при $\theta_w = 10$ тип состояния равновесия на фазовом портрете системы в пространстве «Температура» θ — «Концентрация этана» C_1 — «Концентрация водяного пара» C_2 — устойчивый узел, к которому стремятся из различных начальных условий все траектории (рис. 3, а).

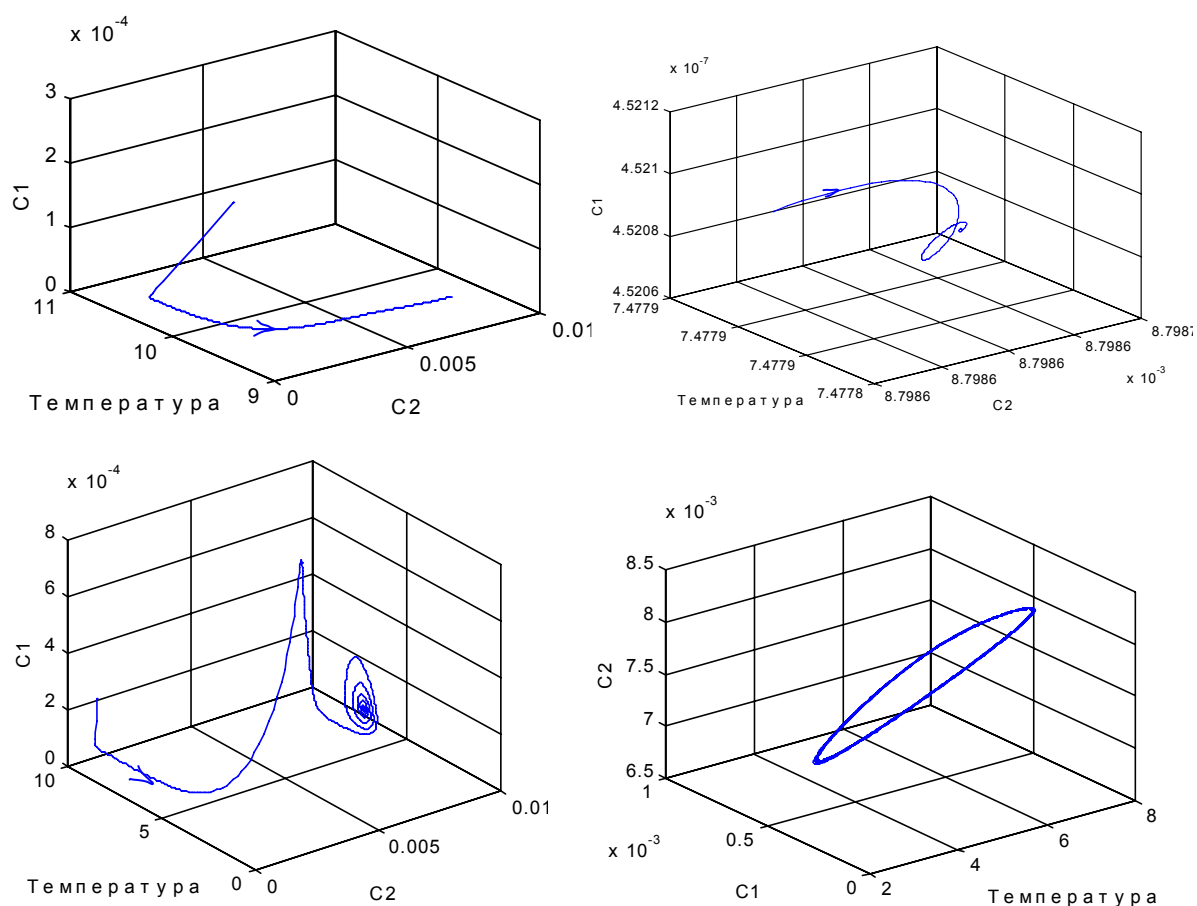


Рис.3. Фазовые портреты системы при различных значениях температуры хладагента: $S_1=S_2=0,8$; $\theta_{вх}=0$; $\theta_n=8,85$: $\theta_w=10$ (а); 6,5 (б); 1 (в); 0,0541 (г).

С уменьшением температуры стенок реактора $\theta_w < \theta_n$ ($\theta_w \approx 0,5 \div 3$) в системе возникают затухающие колебания, фазовый портрет системы соответствует устойчивому состоянию равновесия - устойчивому фокусу, и имеет вид закручивающихся спиралей. Из рис. 3, б, в, иллюстрирующих режимы затухающих колебаний, видно, что чем выше температура стенок реактора, тем быстрее затухают колебания. При дальнейшем уменьшении θ_w фокус постепенно смещается, пока не достигнет бифуркационного значения θ_w . В этой точке происходит качественная перестройка портрета, выражающаяся в рождении предельного цикла, то есть вхождению в режим автоколебаний. Чем ближе θ_w к бифуркационному значению параметра, тем меньше амплитуда возникающих колебаний (мягкое возбуждение колебаний).

При наложении пульсаций температуры стенок реактора на режим затухающих колебаний (рис. 1) по закону:

$$\theta_w(\tau) = \theta_{w1}(\tau) \cdot [1 + A_1 \cdot \sin(\omega_1 \cdot \tau)] \quad (9)$$

при $A_1=1$; $\theta_w=1$ в системе устанавливаются незатухающие колебания с частотой воздействия (рис. 4, а, б). Амплитудно-частотная характеристика (кривая 1, рис. 6, б) отражает параметрический резонанс вблизи частоты воздействия, близкой к собственной. Высокочастотные воздействия не оказывают влияние на режимы окисления.

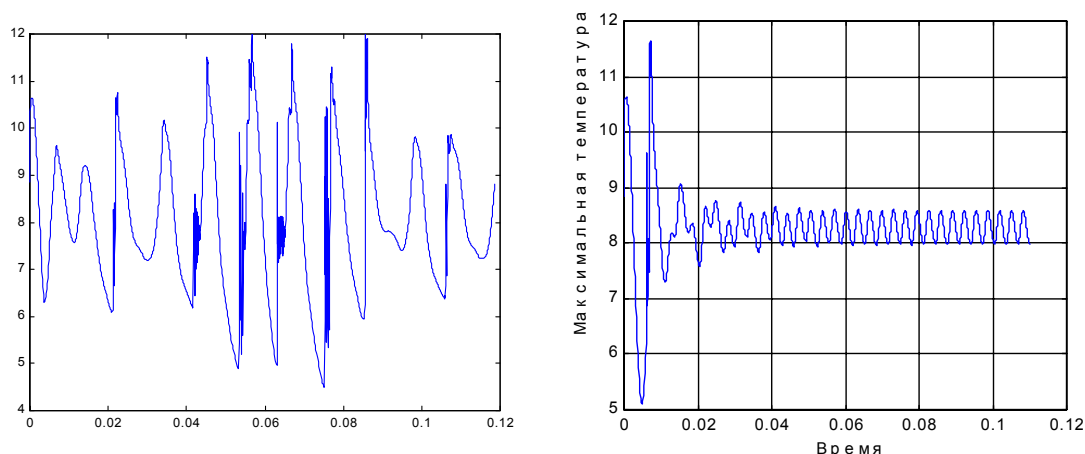


Рис.4. Влияние частоты пульсаций температуры стенки на температуру в реакторе при $S_1=S_2=0,8$: (а). $\omega/\omega_0=0,75$ (а); 2,5(б).

Для режима со сложными квазипериодическими колебаниями, показанного на рис. 4, а характерно возникновение в отдельные моменты времени локальных температурных неоднородностей (рис. 5, а), которые затем сменяются типичным, при протекании экзотермической химической реакции распределением температуры в реакторе (рис. 5, б) (цифры на рис. 5, а, б соответствуют значениям температуры θ). То есть, профиль температуры по длине трубки имеет ярко выраженный максимум, так на-

зываемую «горячую точку», которая находится на оси реактора. Как правило, в этом месте температура близка к максимально допустимой для данного процесса, ее превышение может привести к спеканию катализатора, воспламенению реакционной смеси и т. д. Поэтому значение и положение «горячей точки» — важный показатель работы трубчатых реакторов.

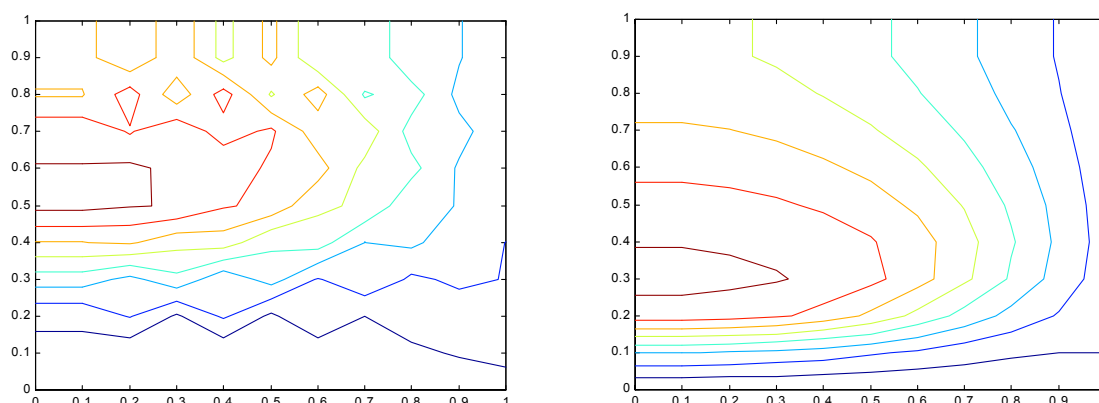


Рис.5. Картины изотерм в реакторе для режима на рис.4,а в различные моменты времени: $\tau=0,053$ (а); $0,12$ (б).

При воздействии пульсациями скорости подачи смеси по закону:

$$Pe(\tau) = Pe_1(\tau) \cdot [1 + A_2 \cdot \sin(\omega_2 \cdot \tau)] \quad (10)$$

при $A_2=1$, $Pe_1=121,25$ на высокотемпературный, устойчивый режим из области неединственных стационарных состояний (рис. 6, а), в системе возникают незатухающие колебания с частотой воздействия (за характерную частоту ω_0 принималась $\omega_0=\pi/\tau_{уст}$, где $\tau_{уст}$ — время установления стационарного режима). При низкочастотных воздействиях имеют место колебания между высокотемпературным и низкотемпературным стационарными состояниями, при высокочастотных — колебания в высокотемпературном режиме. Резонанс наблюдается на частоте $\omega_2 \approx 1,2 \cdot \omega_0$. Таким образом, амплитудно-частотная характеристика (кривая 2, рис. 6, б) носит резонансный характер, даже при воздействии на первоначально устойчивый режим.

Амплитудно-частотные характеристики, полученные при воздействиях вида (13) и (14) носят непрерывный характер. В обоих случаях имеет место смещение резонансной частоты относительно собственной, что связано с существенной нелинейностью процесса.

Таким образом, в результате численных экспериментов на математической модели (1)–(8) получено, что экзотермическая химическая реакция чувствительна к изменению температуры стенок реактора. Изменение температуры хладагента влияет не только на температуру в реакторе, но и на характер режима окисления и может приводить к появлению локальных температурных неоднородностей, ухудшающих работу реактора. При исследовании периодических пульсаций скорости подачи реагентов и темпе-

ратуры стенок реактора на выбранные режимы окисления установлены резонансные эффекты, свойственные нелинейным системам.

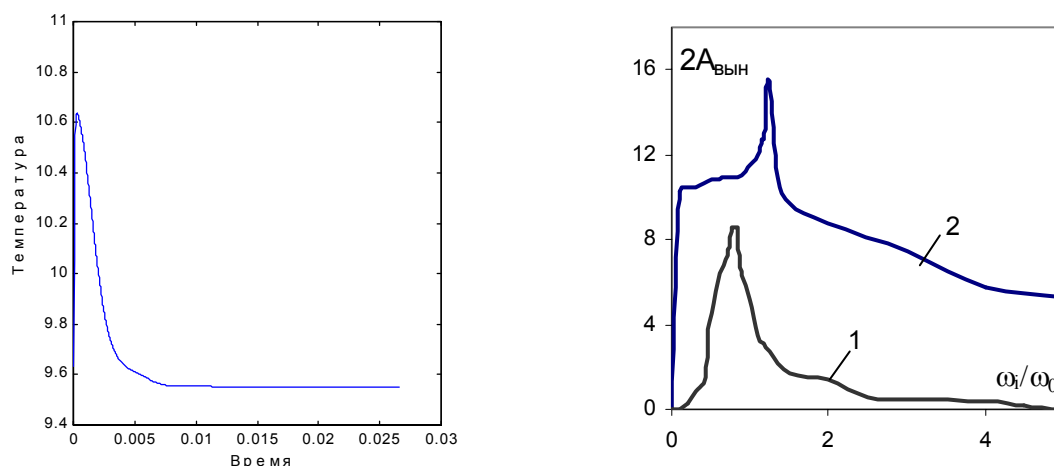


Рис.6. Изменение температуры со временем (кривая 1, рис.6,а) (а) и амплитудно-частотные характеристики системы для воздействий вида (9) (кривая 1) и (10) (кривая 2) (б): а) $S_1=S_2=0,8$; $\omega/\omega_0=0$ (1); 0,1(2); 0,4(3); 1,15(4); 1,5(5); 5(6).

Применение системы MATLAB позволило существенно сократить время вычислений и увеличить эффективность решения задачи. Полученные в результате вычислений огромные массивы данных были без труда проанализированы с помощью мощных и разнообразных графических возможностей системы MATLAB.

Литература

1. Vortmeyer D. and Winter R.P. Impact of porosity and velocity - distribution on theoretical prediction of fixed-bed chemical reactor performance // Chem.React. Eng.— V.6.— 1981.— №2.— P.49–54.
2. Вержбицкая И. С., Ицкова П. Г., Лукьянов А. Т. Математическое моделирование устойчивости стационарных режимов окисления этана в проточном реакторе // Горение и плазмохимия. Алматы: Казак университеті.— 2003.— № 1.— С.47–54.

УДК 004

ПРИМЕНЕНИЕ ПАКЕТА MATLAB ДЛЯ РЕШЕНИЯ КРАЕВЫХ ЗАДАЧ МЕХАНИКИ ЖИДКОСТИ И ГАЗА

Волков К. Н., Емельянов В. Н.

Балтийский государственный технический университет

«ВОЕНМЕХ» им. Д. Ф. Устинова, Санкт-Петербург,

e-mail: root@kv7340.spb.edu, root@ev.spb.su

Введение

Систематическое использование возможностей, предоставляемых новыми информационными технологиями, средствами объектно-ориентированного программирования и современными операционными системами, требует рассмотрения особенностей решения краевых задач механики жидкости и газа, описываемых дифференциальными уравнениями, при помощи таких оболочек.

Применение системы MATLAB открывает новые возможности по формализации и реализации конечно-разностных и конечно-объемных методов численного решения краевых задач механики жидкости и газа.

В состав MATLAB входит пакет PDE Toolbox, который содержит средства для решения нестационарных дифференциальных уравнений в частных производных второго порядка при помощи метода конечных элементов. Команды и графический интерфейс пакета PDE Toolbox могут быть использованы для математического моделирования широкого класса инженерных и научных приложений (сопротивление материалов, электротехника, тепломассоперенос, механика жидкости и газа). Однако при решении задач механики жидкости и газа конечно-элементный подход оказывается менее точным, чем метод конечных разностей или метод конечного объема. Во многих случаях требуются подходы, учитывающие структуру исходной системы уравнений, особенности формулировки граничных условий, сложную пространственно-временную зависимость правой части уравнения и вид расчетной области.

Тип дискретизации и используемый сеточный шаблон в значительной степени определяют свойства и особенности разностной схемы и вычислительной процедуры в целом (устойчивость, точность, время счета и другие показатели). Определенное влияние на показатели численного решения оказывают метод решения системы разностных уравнений, разностная реализация граничных условий и ряд дополнительных факторов, характеризующих выбранную разностную схему (например, распределение узлов, методы ускорения сходимости).

В работе рассматриваются возможности построения векторизованных алгоритмов решения краевых задач, описываемых дифференциальными уравнениями в частных производных второго порядка при помощи метода конечных разностей, а также особенности их программной реализации в пакете MATLAB. Разработанные алгоритмы, с одной стороны, широко используют функции MATLAB, предназначенные для обработки векторов и разреженных матриц, а с другой, отличаются высокой эффективностью и скоростью счета, сравнимыми с показателями программ, написанных на языке C/C++. Несмотря на то, что разработанные алгоритмы иллюстрируются на примере решения достаточно простых задач (решение уравнения Пуассона и уравнения нестационарной теплопроводности), они допускают сравнительно простое обобщение на более сложные задачи, например, решение уравнений Эйлера на основе TVD-схемы Хартена и интегрирование уравнений Навье–Стокса, записанных в координатах функции тока — завихренности (в том числе, в криволинейных координатах). При этом распределение функции тока описывается уравнением Пуассона, а распределение завихренности — уравнением, совпадающим по форме с уравнением нестационарной теплопроводности.

Векторизованные сеточные структуры

Разностная стека, покрывающая расчетную область, имеет структуру, схожую со структурой двумерного массива, что позволяет производить адресацию к ее ячейкам или узлам через систему двух индексов i и j . Узел сетки с индексами (i, j) имеет соседей, к которым легко адресоваться, давая соответствующие приращения индексов. Указанный принцип адресации работает как для обозначения узлов сетки в методе конечных разностей, так и для обозначения ячеек в методе контрольного объема, причем такая адресация возможна и в физическом, и в вычислительном пространстве.

При программировании вычислительных задач работа с сеточными структурами обычно осуществляется при помощи переборного алгоритма, реализованного на основе вложенных циклов. Данный подход подразумевает вычисление индексного выражения и занесение результатов вычислений в двумерный массив U .

Векторизация позволяет работать со множеством данных как с единой вычислительной структурой, что не только делает компактной запись вычислительного алгоритма, избегая вложенных циклов, но и повышает эффективность вычислений. Организовать векторизацию вычислений можно при помощи средств современных объектно-ориентированных языков программирования, в частности, при помощи пакета MATLAB.

Адресация к внутренним ячейкам

Рассмотрим принципы построения векторизованных алгоритмов для решения двумерных задач на структурированных сетках. Векторизация в таких задачах, исключая вложенные циклы, обеспечивается за счет соответствующей организации данных и использования векторизованных операций.

В задачах механики жидкости и газа двумерным структурам данных соответствуют поля гидродинамических величин и координаты сеточных узлов при решении задачи в области прямоугольной формы или в криволинейной области, которую можно отобразить на вычислительный прямоугольник при помощи преобразования координат [1].

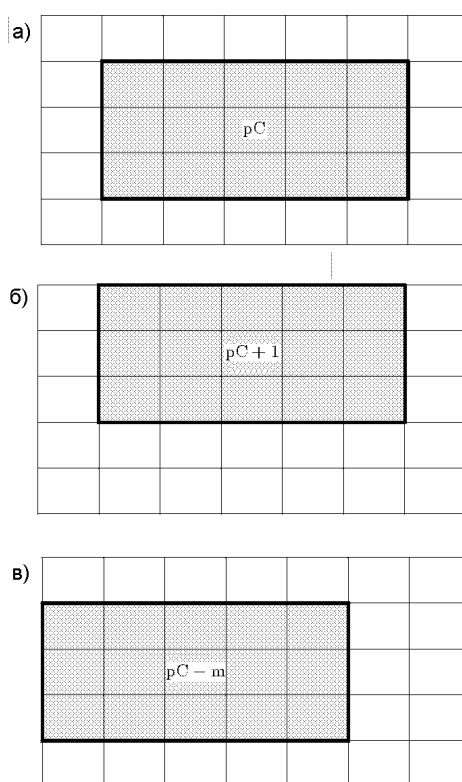


Рис. 1. Адресация к подмножествам ячеек.

При организации вычислительного алгоритма возникает необходимость конечно-разностного представления дифференциальных операторов, фигурирующих в уравнениях или граничных условиях. В этом случае, кроме текущего избираемого данного, необходимо обеспечить адресацию и к данным в соседних узлах, входящих в вычислительный шаблон. Традиционный способ адресации к таким данным состоит в указании двух индексов, определяющих столбец и строку выбираемого данного.

Пронумеруем ячейки расчетной сетки сквозным образом (например, по столбцам). Тогда при помощи одномерного массива индексов возможно

указать определенное подмножество ячеек, покрывающих расчетную область (рис. 1).

Например, обозначив через \mathbf{pC} вектор (множество) индексов всех внутренних ячеек расчетной области прямоугольной формы (рис. 1, а), можно указать подмножества ячеек, которые лежат выше или ниже заданного слоя ячеек (рис. 1, б), прибавив или отняв единицу от всех компонент вектора \mathbf{pC} и образовав для обозначения этих ячеек вектора

$$\mathbf{pU}=\mathbf{pC}+1;$$

$$\mathbf{pD}=\mathbf{pC}-1.$$

Таким же способом можно определить индексные вектора для всех левых и правых ячеек расчетной области

$$\mathbf{pL}=\mathbf{pC}-m;$$

$$\mathbf{pR}=\mathbf{pC}+m.$$

Величина m определяет количество ячеек (число элементов индексного вектора) в столбце расчетной области (рис. 1, в).

Адресация к граничным ячейкам

В вычислительных алгоритмах, реализованных при помощи метода конечных разностей на разнесенной шахматной сетке или при помощи метода контрольного объема, обычно выделяются фиктивные граничные ячейки, образующие вместе с внутренними ячейками расширенную расчетную область. Внутренние ячейки выделены заливкой на рис. 2, а адресация к ним обеспечивается с помощью индексного вектора \mathbf{pC} .

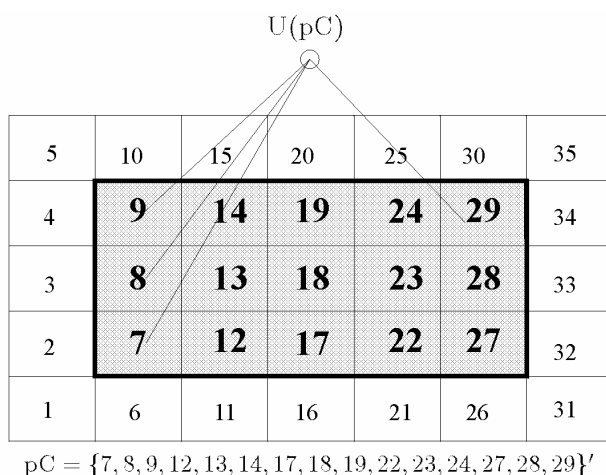


Рис. 2. Расширенная область ячеек.

Способ адресации к фиктивным граничным ячейкам, вводимым для постановки граничных условий, поясняет рис. 3. Для этого строятся индексные вектора \mathbf{pBL} , \mathbf{pBU} , \mathbf{pBR} , \mathbf{pBD} , соответствующие левой, верхней, правой и нижней граням расчетной области. Способ формирования этих

векторов такой же, как и индексных векторов внутренних ячеек (угловые ячейки при этом не используются).

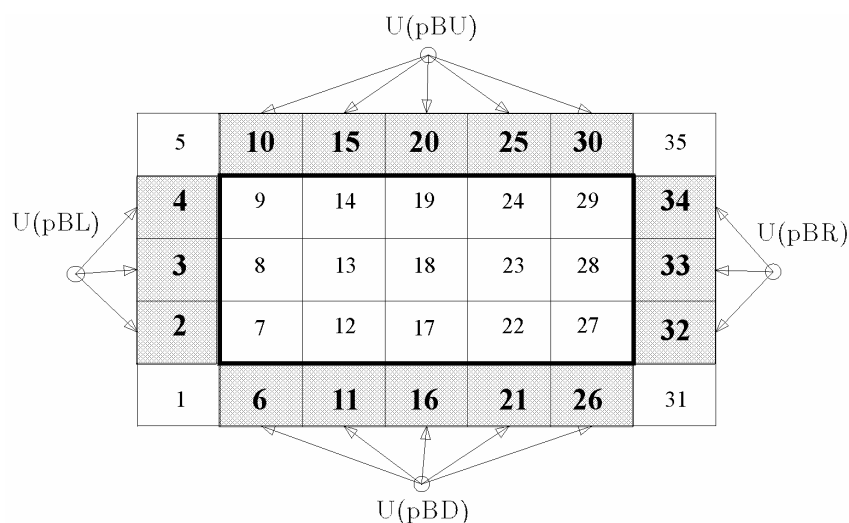


Рис. 3. Индексные вектора, соответствующие граничным ячейкам.

Вычисление потоков

Рассмотрим вопрос о векторизованном вычислении потоков на границах расчетных ячеек для двумерных задач, реализованных на структурированных сетках (например, на прямоугольной сетке, ориентированной по осям декартовой системы координат). В качестве объекта, которому соотносится поток, выбирается грань, разделяющая две соседние ячейки. Привязка потока к грани позволяет избежать его повторного вычисления при реализации законов сохранения для контрольного объема. Одна и та же величина потока участвует в построении балансовых соотношений для двух соседних ячеек (входящий в ячейку поток является выходящим для соседней ячейки), что обеспечивает консервативность разностной схемы. При введении дополнительных фиктивных ячеек на границах области все ячейки, расположенные внутри области, имеют соседей и вычисление потоков для всех граней проводится по единому алгоритму.

Для вычисления потоков формируется два множества индексных векторов, адресуемых к ячейкам. Один из таких векторов **pCFL** определяет множество всех ячеек, расположенных слева от боковых граней (рис. 4а). Другой индексный вектор **pCFR** обеспечивает операцию подстановочной адресации для множества всех правых ячеек (рис. 4б). Индексный вектор множества правых ячеек определяется как $\mathbf{pCFR} = \mathbf{pCFL} + m$. Векторная операция вычисления потоков задается соотношением

$$\mathbf{F}(\mathbf{pC}) = \mathbf{F}(\mathbf{U}(\mathbf{pCFR}), \mathbf{U}(\mathbf{pCFL})). \quad (1)$$

Последовательность вычислений, записанная в обычных индексных выражениях, имеет вид

$$\mathbf{F}_{i+1/2,j} = \mathbf{F}(\mathbf{U}_{i+1,j}, \mathbf{U}_{i,j}). \quad (2)$$

Индексы i и j относятся к центру ячейки, а индекс $i+1/2$ соответствует грани, разделяющей ячейки с индексами i и $i+1$.

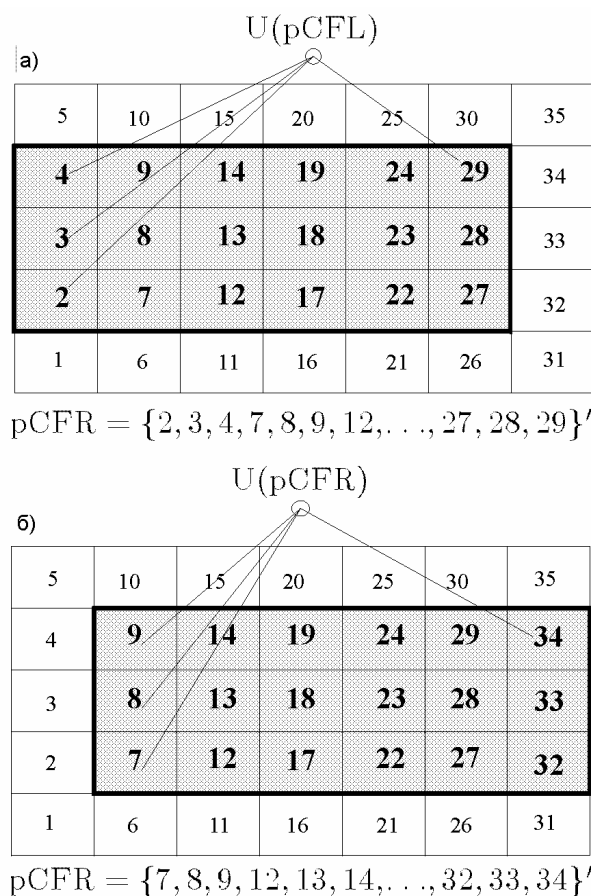


Рис. 4. Множества левых и правых ячеек расчетной области.

Отметим принципиальное различие между выражениями (1) и (2), используемыми для расчета потоков. В то время как соотношение (1) определяет все множество потоков, соотношение (2) требует двойного цикла вычислений по индексам i и j . Преимущества подхода, основанного на выражении (1), проявляются при применении вычислительных сред, допускающих выполнение векторизованных операций, в частности, пакета MATLAB. Современные языки программирования (например, C++) также позволяют создавать векторизованные алгоритмы, эффективность которых проявляется при соответствующем аппаратном обеспечении.

Поскольку поток привязывается к грани ячейки, а параметры газа — к ее центру (рис. 5), то необходимо определить соответствие индексных выражений для вычисления потоков с индексами, определяющими ячейки расчетной области. Несмотря на то, что поток содержит меньшее количество элементов по сравнению с расширенным множеством ячеек, имеет смысл хранить его в структуре той же размерности (хотя ряд элементов

при этом и не используется, но удобства адресации и наглядность преобладают над соображениями экономии памяти).

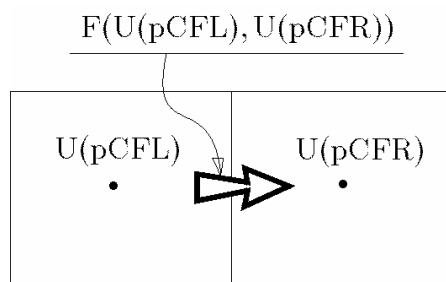


Рис. 5. Векторизованное вычисление потока.

Сформируем индексный вектор для определения множества потоков через боковые грани расчетных ячеек исходя из схемы размещения потоков, приведенной на рис. 6.

5	10	15	20	25	30	35
4	9	14	19	24	29	34
3	8	13	18	23	28	33
2	7	12	17	22	27	32
1	6	11	16	21	26	31

$pC = \{2, 3, 4, 7, 8, 9, 12, \dots, 27, 28, 29\}'$

Рис. 6. Размещение потоков и структура вектора ссылок.

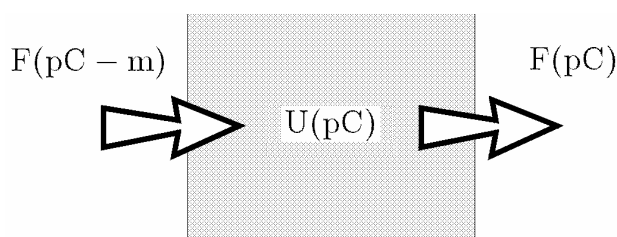


Рис. 7. Вычисление потоков через грани ячейки.

При такой организации данных легко записываются векторизованные балансовые соотношения для каждой ячейки расчетной области. Адресация к потокам через боковые грани множества ячеек показана на рис. 7. Для всех расчетных ячеек, определяемых индексным вектором pC , потоки через их левые грани определяются индексным вектором $pC - m$, а через правые грани — индексным вектором pC . Величина m представляет собой количество ячеек в столбце расширенной области ячеек и определяет сдвиг при переходе от двумерной матричной структуры к линейному размещению элементов.

Вычисление производных

Для вычисления производной в пакете MATLAB используется функция $y = \text{diff}(x, n)$, которая вычисляет конечные разности порядка n , удовлетворяющие рекуррентному соотношению $\text{diff}(x, n) = \text{diff}(\text{diff}(x, n-1))$. Если x — одномерный массив $x = [x(1) \ x(2) \ \dots \ x(n)]$, то $\text{diff}(x)$ представляет собой вектор разностей соседних элементов

$$\text{diff}(x) = [x(2) - x(1) \ x(3) - x(2) \ \dots \ x(n) - x(n-1)].$$

Количество элементов вектора x на единицу меньше количества элементов вектора $\text{diff}(x)$. Приближением производной порядка n является отношение $\text{diff}(y, n) / \text{diff}(x, n)$.

Вычислить производную от сеточной функции U , определенной на последовательном наборе равноотстоящих точек, можно и другим способом. Разобьем промежуток интегрирования $x_0 \leq x \leq x_D$ на N равных отрезков, а длину каждого из них обозначим через $\Delta x = x_i - x_{i-1}$ (рис. 8). Пронумеруем все промежутки разностного шаблона от 1 до $N-1$, где N — число узлов.

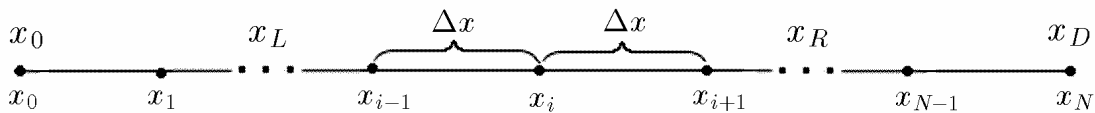


Рис. 8. Разбиение промежутка интегрирования по координате x .

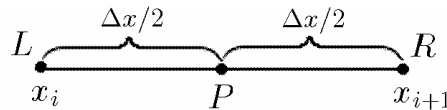


Рис. 9. Разностный шаблон.

Выберем разностный шаблон так, как показано на рис. 9. Серединную точку каждого отрезка обозначим через P , а левую и правую границы каждого интервала — через L и R соответственно (всего имеется $N-1$ точек P , L и R).

Введем массив индексов I для всех элементов данной структуры, в котором последовательное возрастание индексов соответствует порядку следования его элементов

$$I = \{1, 2, \dots, P-1, P, P+1, \dots, N-1, N\}' = 1:N.$$

Дополнительно введем массивы индексов для левых и правых элементов сеточной структуры

$$L = \{1, 2, \dots, P-1, P, P+1, \dots, N-2, N-1\}' = 1:N-1;$$

$$R = \{2, 3, \dots, P-1, P, P+1, \dots, N-1, N\}' = 2:N.$$

Если соответствующим программно-математическим обеспечением поддерживается векторная операция вычитания

$$\Delta U = U(R) - U(L),$$

понимаемая как

$$\begin{pmatrix} \Delta U(1) \\ \Delta U(2) \\ \vdots \\ \Delta U(N-1) \end{pmatrix} = \begin{pmatrix} U(R(1)) - U(L(1)) \\ U(R(2)) - U(L(2)) \\ \vdots \\ U(R(N-1)) - U(L(N-1)) \end{pmatrix},$$

то формула векторизованного вычисления производной во всех точках разностного шаблона имеет вид

$$\frac{dU(x)}{dx} = \frac{U(R) - U(L)}{\Delta x^2} + O(\Delta x^2).$$

Занесем номера узлов в массив **iEq**. Номера узлов левой и правой границы каждого промежутка занесем в массивы **iL** и **iR** соответственно. Тогда алгоритм вычисления производной можно представить в виде следующей последовательности шагов

```
% индексы промежутков и их граничных узлов
iEq=1:N-1; iL=iEq; iR=iEq+1;
% длина промежутка
dx=1/(N-1);
% производная y'(x)
ур=(y(iR)-y(iL))/dx;
```

Именно данный подход лежит в основе разработки векторизованных алгоритмов решения краевых задач.

Решение уравнения Пуассона

Рассмотрим решение краевой задачи Дирихле для уравнения Пуассона в области Ω , ограниченной поверхностью Γ , с правой частью, зависящей от пространственных координат

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = R(x, y). \quad (3)$$

Граничные условия для уравнения (3) имеют вид

$$U(x, y) = U_\Gamma \text{ при } x, y \in \Gamma. \quad (4)$$

Уравнение (3) с граничными условиями (4) описывает, в частности, ламинарное течение вязкой несжимаемой жидкости в канале с произвольной формой поперечного сечения в плане [2] (течение Пуазейля). В этом случае правая часть имеет вид

$$R(x, y) = \frac{1}{\mu} \frac{dp}{dx} = \frac{1}{\mu} \frac{\Delta p}{\Delta x},$$

где μ — динамическая вязкость, а градиент давления dp/dx считается заданной величиной.

Для решения уравнения (3) используется метод конечных разностей [3]. Шаги сетки полагаются равными

$$\Delta x = \frac{1}{n_x - 1}, \quad \Delta y = \frac{1}{n_y - 1}.$$

Здесь n_x , n_y — число узлов разностной сетки по координатным направлениям x и y соответственно.

При использовании центральных конечно-разностных формул второго порядка на пятиточечном сеточном шаблоне (рис. 10) уравнение Пуассона имеет следующий дискретный вид

$$\frac{U_E - 2U_P + U_W}{\Delta x^2} + \frac{U_N - 2U_P + U_S}{\Delta y^2} = R(x_P, y_P).$$

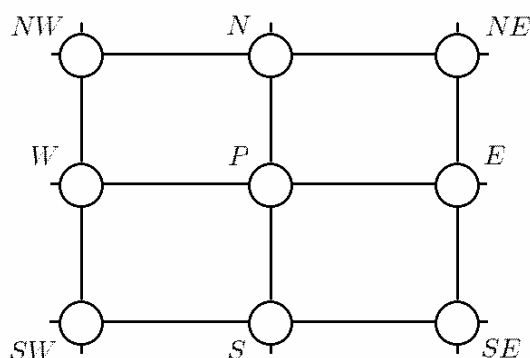


Рис. 10. Шаблон для дискретизации уравнения Пуассона.

Совокупность конечно-разностных уравнений и граничных условий, записанных в дискретной форме, представляет собой связанную систему линейных алгебраических уравнений для определения сеточных значений искомой функции в узлах расчетной области $\mathbf{AU}=\mathbf{B}$, где \mathbf{A} — матрица линейной системы, \mathbf{U} — вектор-столбец неизвестных, \mathbf{B} — вектор-столбец правых частей. Разреженная структура матрицы \mathbf{A} позволяет реализовать компактные методы хранения ее компонент и использовать эффективные в вычислительном плане методы обращения.

Приведем некоторые моменты программной реализации численного решения краевой задачи Дирихле для уравнения Пуассона в области Т-образной формы (рис. 11).

Сначала рассматриваются способы формирования расчетной области и индексации внутренних и граничных узлов, а затем — разностная дискретизация уравнения Пуассона.

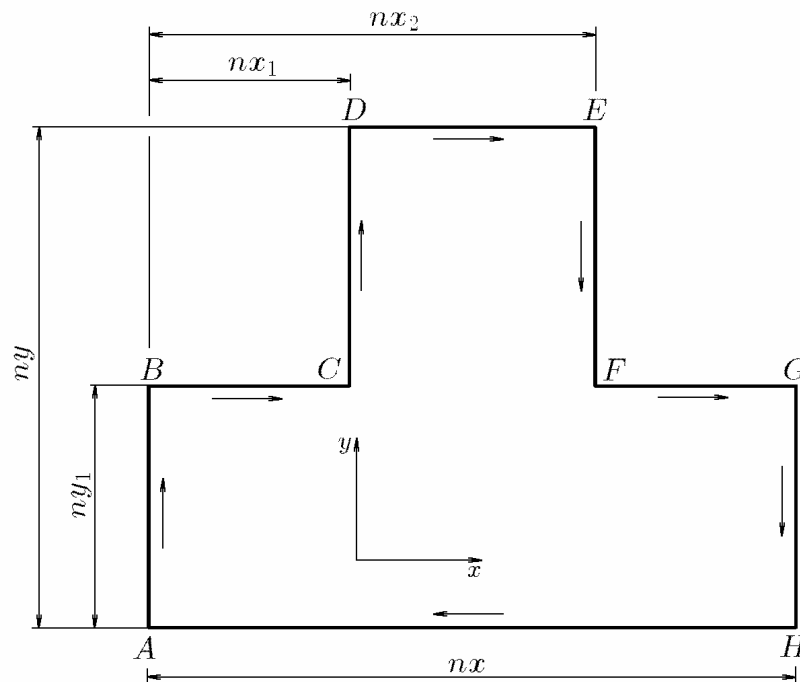


Рис. 11. Структура расчетной области для решения краевой задачи Дирихле для уравнения Пуассона.

На подготовительном этапе производится задание входных данных, таких как размерности сетки, правой части уравнения, а также ряда других. Для упрощения используются целочисленные единицы, а правая часть уравнения Пуассона полагается равной единице.

```
% размер сетки по x и y
nx=7;
ny=6;
% вспомогательные размерности сетки
nx1=3;
nx2=5;
ny1=4;
% шаги сетки по x и y
h1=1/(nx-1);
h2=1/(ny-1);
% правая часть уравнения Пуассона (после преобразований)
RHS=-2*h1^2;
```

Для индексации внутренних и граничных точек области прямоугольной формы $[1, n_x] \times [1, n_y]$, внутри которого лежит расчетная область, формируются две матрицы **NX** и **NY**, одна из которых содержит порядковые номера узлов по горизонтали (номера столбцов), а другая — по вертикали (номера строк). Для этого используется функция `ones(m, n)`, которая формирует массив единиц размерности $m \times n$.

```
% матрицы, содержащие индексы узлов расчетной области
% по горизонтали (номера столбцов, одинаковые строки)
NX=ones(ny,1)*(1:nx);
```

% по вертикали (номера строк, одинаковые столбцы)

NY=(1:ny)'*ones(nx,1)';

Для сеточной области размером 7×6 такие матрицы имеют вид

$$\mathbf{NX} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}, \quad \mathbf{NY} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 & 6 \end{pmatrix}.$$

В соответствии с заданной формой расчетной области, из рассмотрения исключаются точки, находящиеся за ее пределами. Для этого, например, удобно сопоставить единицы точкам, входящим в расчетную область, а нули — точкам, не входящим в ее пределы. Используя матрицы **NX** и **NY**, строится матрица **G** соответствующего вида. Программно построение матрицы **G** производится при помощи операторов отношения и оператора логического отрицания.

% матрица, содержащая единицы в узлах, соответствующих

% узлам расчетной области, и нули в остальных точках

G=(~(NX<nx1&NY>ny1))&(~(NX>nx2&NY>ny1));

Для заданной расчетной области Т-образного вида имеем

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Последовательно пронумеруем все точки расчетной области и занесям их номера в вектор-столбец **pG**. Точки, не входящие в расчетную область, тоже нумеруются, но их номера не заносятся в вектор **pG**. Для формирования вектора подстановок используется функция **k=find(X)**, которая возвращает индексы ненулевых элементов матрицы **X** (если таких элементов нет, то возвращается пустой вектор). При этом матрица **X** рассматривается как вектор-столбец, образованный объединением столбцов исходной матрицы.

% вектор-столбец, содержащий глобальные номера всех узлов

pG=find(G);

Для рассматриваемого примера вектор **pG** имеет вид

pG={1, ..., 4, 7, ..., 10, 13, ..., 34, 37, ..., 40}'.

В вектор **pG** не входят точки с номерами 5, 6, 11, 12, 35, 36, 41, 42, находящиеся за пределами области Т-образной формы.

Используя вектор подстановок **pG**, в матрицу **G** заносятся порядковые номера всех точек расчетной области.

```
% матрица, содержащая порядковые номера всех узлов
G(pG)=(1:length(pG))';
```

После этого матрица **G** принимает следующий вид

$$\mathbf{G} = \begin{pmatrix} 1 & 5 & 9 & 15 & 21 & 27 & 31 \\ 2 & 6 & 10 & 16 & 22 & 28 & 32 \\ 3 & 7 & 11 & 17 & 23 & 29 & 33 \\ 4 & 8 & 12 & 18 & 24 & 30 & 34 \\ 0 & 0 & 13 & 19 & 25 & 0 & 0 \\ 0 & 0 & 14 & 20 & 26 & 0 & 0 \end{pmatrix}.$$

Нумерация производится сверху вниз (по столбцам). Элементы матрицы **G**, которые соответствуют точкам прямоугольника $[1, n_x] \times [1, n_y]$, не входящим в пределы области заданной формы, имеют, по-прежнему, нулевые значения.

Применяя тот же самый прием, формируются вектора, содержащие номера граничных узлов. Вектора, соответствующие отдельным участкам Т-образной границы, обозначим через **pAB**, **pBC**, **pCD**, **pDE**, **pEF**, **pFG**, **pGH**, **pHA**. Приведем последовательность операторов, при помощи которых реализуется построение индексных векторов. При этом функция **k=find(«условие»)** возвращает индексы элементов, удовлетворяющих заданному условию.

```
% вектора, содержащие глобальные номера точек на границах
pAB=find((G>0)&(NX==1));
pBC=find((G>0)&(NY==ny1)&(NX<=nx1));
pCD=find((G>0)&(NX==nx1)&(NY>=ny1));
pDE=find((G>0)&(NY==ny));
pEF=find((G>0)&(NX==nx2)&(NY>=ny1));
pFG=find((G>0)&(NY==ny1)&(NX>=nx2));
pGH=find((G>0)&(NX==nx));
pHA=find((G>0)&(NY==1));
```

В данном случае они имеют вид (номера граничных узлов упорядочиваются по возрастанию)

```
pAB={1, 2, 3, 4}',    pBC={4, 10, 16}',
pCD={16, 17, 18}',    pDE={18, 24, 30}',
pEF={30, 29, 28}',    pFG={28, 34, 40}',
pGH={37, 38, 39, 40}',    pHA={1, 7, 13, 19, 25, 31, 37}'.
```

Используется сквозная нумерация граничных узлов, включая точки, находящиеся за пределами расчетной области.

В результате, при помощи простого слияния формируется вектор подстановок **pB**, содержащий номера всех граничных точек (двойной учет угловых точек исключается).

```
% вектор-столбец, содержащий глобальные номера граничных точек
pB=[pAB(2:length(pAB)-1);pBC;pCD(2:length(pCD)-1);pDE;...
```

pEF(2:length(pEF)-1);pFG;pGH(2:length(pGH)-1);pHA];

Теперь можно построить матрицу **Gp**, содержащую номера только внутренних узлов. Матрица **Gp** строится в два приема. Сначала, используя матрицу **G** и вектор подстановок **pB**, формируется промежуточная матрица, содержащая номера только граничных узлов, остальные ее элементы полагаются равными нулю. Функция `zeros(m, n)` возвращает массив нулей размерности $m \times n$.

% шаг 1

% промежуточная матрица, содержащая номера всех граничных точек
% (точки за пределами и внутри области не нумеруются)

Gp=zeros(ny,nx);

Gp(pB)=G(pB);

Промежуточная матрица имеет вид

$$\mathbf{Gp} = \begin{pmatrix} 1 & 5 & 11 & 15 & 21 & 27 & 31 \\ 2 & 0 & 0 & 0 & 0 & 0 & 32 \\ 3 & 0 & 0 & 0 & 0 & 0 & 33 \\ 4 & 8 & 12 & 0 & 24 & 30 & 34 \\ 0 & 0 & 13 & 0 & 25 & 0 & 0 \\ 0 & 0 & 14 & 20 & 26 & 0 & 0 \end{pmatrix}.$$

Окончательная структура матрицы **Gp** получается путем простого вычитания и выполнения оператора присваивания.

% шаг 2

% матрица, содержащая номера всех внутренних точек

% (точки за пределами и на границе области не нумеруются)

Gp=G-Gp;

В результате, для рассматриваемого примера имеем

$$\mathbf{Gp} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 10 & 16 & 22 & 28 & 0 \\ 0 & 7 & 11 & 17 & 23 & 29 & 0 \\ 0 & 0 & 0 & 18 & 0 & 0 & 0 \\ 0 & 0 & 0 & 19 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Формируется вектор подстановок **pP**, содержащий номера внутренних узлов расчетной области. Как и ранее, для его построения используется функция `k=find(X)`.

% вектор, содержащий глобальные номера внутренних узлов

pP=find(Gp);

Для данного примера вектор **pP** имеет следующую

pP={8, 9, 14, 15, 20, 21, 22, 23, 26, 27, 32, 33}'.

Теперь можно перейти к построению разностного шаблона для дискретизации уравнения Пуассона. Следуя векторному принципу построения

вычислительного алгоритма, формируется четыре массива **pN**, **pE**, **pS**, **pW**, содержащих соответственно номера узлов, используемых для построения разностного шаблона типа «крест» (рис. 10).

% индексные массивы для точек шаблона

% (внутренние точки)

pN=pP+1; pE=pP+ny; pS=pP-1; pW=pP-ny;

Указанные массивы имеют вид

pN={9, 10, 15, 16, 21, 22, 23, 24, 27, 28, 33, 34}';

pE={14, 15, 20, 21, 26, 27, 28, 29, 32, 33, 38, 39}';

pS={7, 8, 13, 14, 19, 20, 21, 22, 25, 26, 31, 32}';

pW={2, 3, 8, 9, 14, 15, 16, 17, 20, 21, 26, 27}';

Перед тем как сформировать матрицу коэффициентов системы разностных уравнений, определяется ее структура. Для этого вводится два целочисленных вектора **indI** и **indJ**, задающие позиции элементов матрицы коэффициентов, а также вещественный вектор **valS**, определяющий числовые значения элементов этой матрицы.

Сначала строятся массивы **indI**, **indJ** и **valS** для граничных узлов, индексы которых хранятся в векторе подстановок **pB**. Поскольку для краевой задачи Дирихле значения искомой функции на границе области заданы, то соответствующие элементы вектора **valS** равны единице.

Используя операцию слияния векторов и алгоритм векторных подстановок, формируются массивы **indI**, **indJ** и **valS** для внутренних узлов при соблюдении последовательности обхода узлов разностного шаблона P, N, E, S, W (по часовой стрелке, как указано на рис. 11). Элементы вектора **valS** находятся в соответствии с используемой разностной схемой.

% вспомогательные коэффициенты разностной схемы

Tmp1=(h1/h2)^2;

Tmp2=2*(1+Tmp1);

% индексная запись коэффициентов системы уравнений

% граничные узлы

indI=G(pB);

indJ=G(pB);

valS=ones(length(pB),1);

% внутренние узлы

indI=[indI;G(pP);G(pP);G(pP);G(pP);G(pP)];

indJ=[indJ;G(pP);G(pN);G(pE);G(pS);G(pW)];

valS=[valS;-Tmp2*ones(length(pP),1);...

Tmp1*ones(length(pP),1);...

ones(length(pP),1);...

Tmp1*ones(length(pP),1);...

ones(length(pP),1)];

Вектор правых частей **B** строится отдельно для внутренних и граничных узлов. Элементы **B(pP)**, связанные с внутренними узлами, равняются правой части решаемого уравнения, которая для простоты полагается равной единице. Для задания значений элементам **B(pB)**, относящихся к граничным узлам, используются соответствующие граничные условия. В

рассматриваемом примере значения искомой функции на границе Т-образной области равны нулю (используется условие прилипания для скорости).

```
% правая часть
B=zeros(length(pG),1);
% внутренние точки
B(G(pP)+0)=ones(length(pP),1)*RHS;
% граничные точки
B(G(pB)+0)=zeros(length(pB),1);
```

Сформированные данные используются для построения матрицы системы линейных алгебраических уравнений и ее решения прямым методом. Функция `sparse(i, j, s)` формирует матрицу в соответствии с правилами записи разреженных матриц, принятыми в системе MATLAB. При вызове этой функции строки массива `[i j s]` используются для формирования разреженной матрицы размерности $m \times n$, где $m = \max(i)$ и $n = \max(j)$. Вектора `i` и `j` задают позиции элементов и являются целочисленными, а вектор `s` определяет числовое значение элемента матрицы. При формировании разреженной матрицы все строки вида `[i j 0]` из описания удаляются. Длина вектора `s` совпадает с количеством ненулевых элементов разреженной матрицы. Решение системы линейных уравнений реализуется в MATLAB с помощью специального монитора, который использует различные алгоритмы в зависимости от структуры матрицы. Эти алгоритмы реализованы в MATLAB на основе функций из пакета LINPACK [4].

```
% формирование разреженной матрицы
A=sparse(indI,indJ,valS);
% решение системы линейных уравнений
U=A\B;
```

Структура матрицы `A` для сетки 6×7 показана на рис. 12.

Визуализация численного решения задачи производится отдельно для каждого прямоугольного фрагмента расчетной области. Линии уровня решения приведены на рис. 13 для сетки 28×24 .

```
% количество линий уровня
NLevel=21;
% подготовка окна
figure(2), clf, hold on, axis([0 nx+1 0 ny+1]), axis('image')
% границы расчетной области
XTmp=[1 1  nx1 nx1 nx2 nx2 nx  nx 1];
YTmp=[1 ny1 ny1 ny  ny  ny1 ny1 1 1];
plot(XTmp,YTmp,'-')
% минимальное и максимальное значения искомой функции
minVX=min(U); maxVX=max(U);
% шаг рисования изолиний
VV=minVX:(maxVX-minVX)/NLevel:maxVX;
vx=G; vx(G>0)=full(U(G(G>0)));
%
vx1=vx(1:ny1,1:nx);
contour(1:nx,1:ny1,vx1,VV);
```

```
%
vx1=vx(ny1:ny,nx1:nx2);
contour(nx1:nx2,ny1:ny,vx1,VV);
```

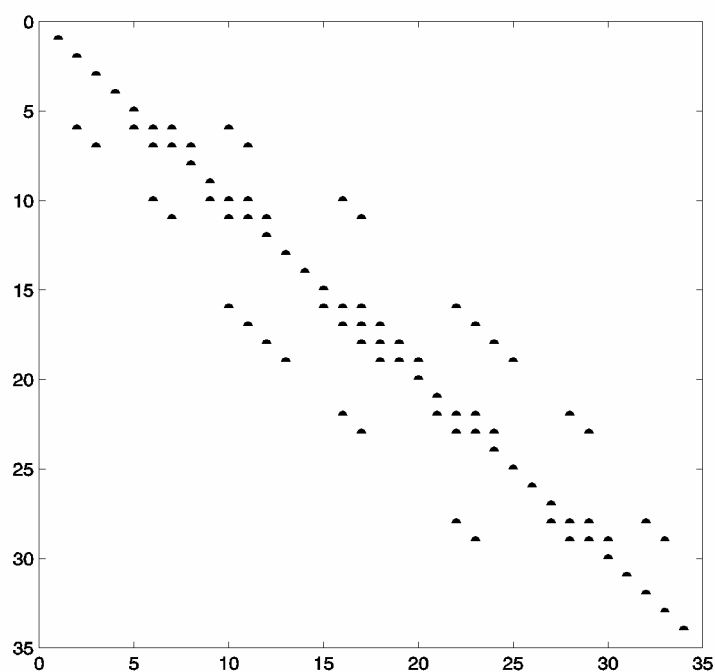


Рис. 12. Структура матрицы коэффициентов для сетки 6×7 . Матрица A имеет размер 34×34 и 82 ненулевых элементов.

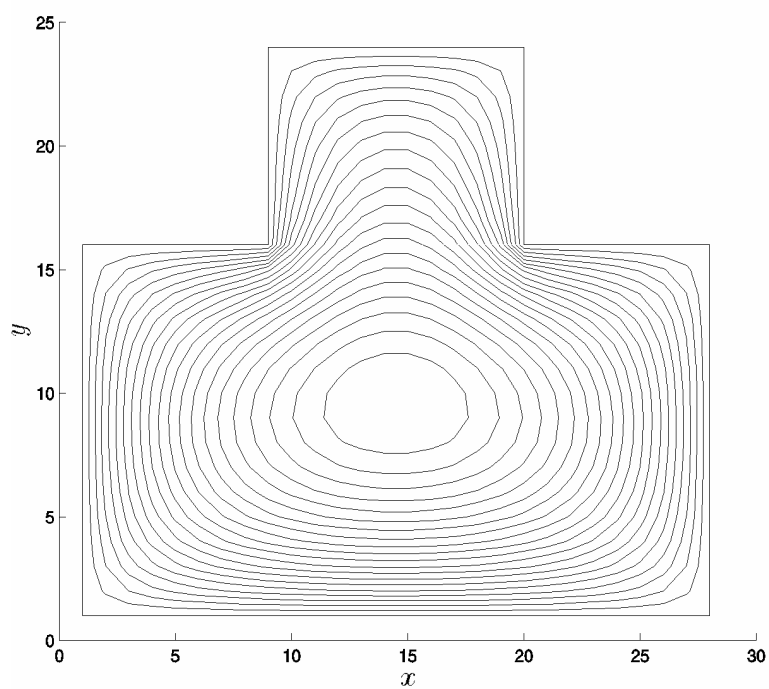


Рис. 13. Линии уровня решения уравнения Пуассона на сетке 28×24 .

Решение уравнения нестационарной теплопроводности

Рассмотрим нагрев пластины конечной толщины лазерным излучением, интенсивность которого зависит от времени (рис. 14). Для описания температурного поля в пластине и в прилегающем к ней пространстве используется нестационарное уравнение теплопроводности. Теплофизические характеристики пластины и окружающей газовой среды считаются постоянными. Модель лазерного импульса подробно изложена в работе [5].

В консервативной форме уравнение теплопроводности с учетом внутренних объемных источников тепла записывается в следующем виде

– для материала пластины

$$\rho_p c_p \frac{\partial T_p}{\partial t} = \lambda_p \frac{\partial^2 T_p}{\partial x^2} + Q(t, x); \quad (5)$$

– для газовой среды

$$\rho_g c_g \frac{\partial T_g}{\partial t} = \lambda_g \frac{\partial^2 T_g}{\partial x^2}; \quad (6)$$

Здесь ρ – плотность, c – теплоемкость, λ – теплопроводность. Индексы g и p относятся к газовой фазе и пластине соответственно.

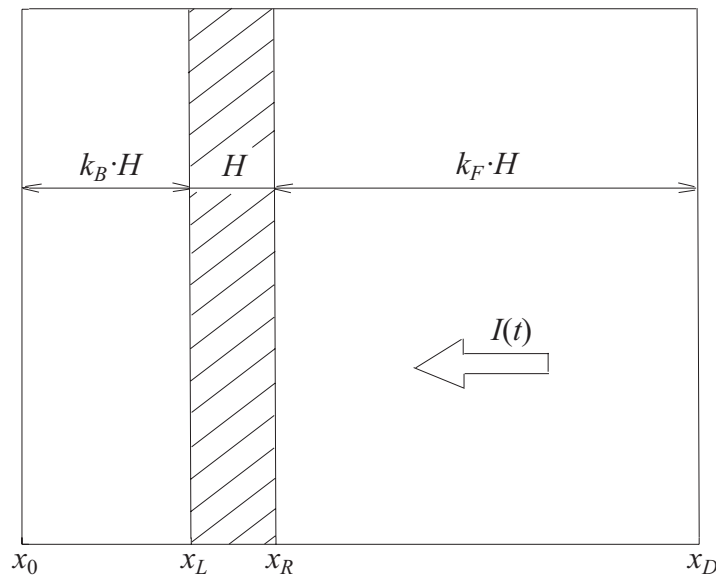


Рис. 14. Структура расчетной области для решения уравнения теплопроводности.

Слагаемое Q в уравнении (6) учитывает объемное тепловыделение, обусловленное поглощением излучения пластиной. Интенсивность энергоподвода, приходящаяся на единицу объема, находится из соотношения $Q=IK/H$, где H — толщина пластины, K — фактор эффективности поглощения излучения.

Предполагается, что в начальный момент времени $t=0$ пластина и газ имеют одинаковую температуру T_0 .

Условие теплового сопряжения на границе пластины с окружающей газовой средой (при отсутствии или несущественности процессов фазового перехода на поверхности пластины) записывается в виде условия непрерывности тепловых потоков

$$-\lambda_p \left(\frac{\partial T}{\partial x} \right)_{w=0} = -\lambda_g \left(\frac{\partial T}{\partial x} \right)_{w+0} + \alpha I(t),$$

которое дополняется условием непрерывности температуры

$$T_p = T_g.$$

Зависимость интенсивности падающего излучения от времени описывается функцией $I(t)$. Ослабление излучения при прохождении через среду моделируется при помощи закона Бугера–Ламберта–Бера с показателем поглощения α .

Поскольку размеры расчетной области не изменяются во времени, то на каждом временном слое используется один и тот же сеточный шаблон, показанный на рис. 8 и рис. 9.

Для дискретизации по времени используется явная схема Эйлера. Тогда на промежутке $[x_i, x_{i+1}]$ уравнение теплопроводности для пластины и газовой среды можно записать в следующем разностном виде

$$\rho c \frac{T_P^{n+1} - T_P^n}{\Delta t} = \lambda \frac{T_P^n - 2T_P^n + T_L^n}{\Delta x^2} + Q(t^n, x_P).$$

Для разностного представления градиентов используются односторонние формулы численного дифференцирования

$$\left(\frac{\partial T}{\partial x} \right)_{w=0} = \frac{T_{w-2} - 4T_{w-1} + 3T_w}{2\Delta x}, \quad \left(\frac{\partial T}{\partial x} \right)_{w+0} = \frac{-3T_w + 4T_{w+1} - 3T_{w+2}}{2\Delta x}.$$

Условие теплового сопряжения на границе раздела фаз представляется в виде следующего линейного соотношения

$$-\lambda_p T_{w-2} + 4\lambda_p T_{w-1} - 3(\lambda_p + \lambda_g) T_w + 4\lambda_g T_{w+1} - \lambda_g T_{w+2} = 0.$$

Приведем особенности построения векторизованного алгоритма численного решения сопряженной краевой задачи для нестационарного уравнения теплопроводности и особенности его программной реализации. В области $[x_L, x_R]$ используется уравнение (5), а в области $[x_0, x_L]$ и $(x_R, x_D]$ — уравнение (8). Размер расчетной области выбирается исходя из условия сопряжения температурного поля газовой среды с граничным условием на бесконечности.

На подготовительном этапе производится задание входных данных, таких как размера расчетной области, координат характерных точек, числа узлов, а также теплофизических свойств материала пластины и газовой среды.

```
% толщина пластинки
H=1.0;
% число узлов разностной сетки
NP=50;
% шаг разностной сетки
hx=H/NP;

% относительный размер области слева и справа от пластины
kBack=2; kFwrд=5;

% координата левой границы области
X0=0;
% координата левой границы пластины и число узлов
XL=kBack*H;      NL=XL/hx+1;
% координата правой границы пластины и число узлов
XR=(1+kBack)*H;  NR=XR/hx+1;
% координата правой границы области и число узлов
XD=(1+kBack+kFwrд)*H; ND=XD/hx+1;
```

Следующий фрагмент программы связан с формированием индексных векторов для области слева от пластины $[x_0, x_L]$, для области, занятой пластиной $[x_L, x_R]$, и для области справа от пластины $(x_R, x_D]$. Для удобства также находятся индексы узлов, соответствующих границам раздела фаз (точки x_L и x_R).

```
% координаты узлов
xAll=(0:hx:XD)';
% индексные вектора для области слева от пластины
pCBack=(2:NL-1)';
pLBack=pCBack-1;
pRBack=pCBack+1;
% индексные вектора для области, занятой пластиной
pCFlat=(NL+1:NR-1)';
pLFlat=pCFlat-1;
pRFlat=pCFlat+1;
% индексные вектора для области справа от пластины
pCFwrд=(NR+1:ND-1)';
pLFwrд=pCFwrд-1;
pRFwrд=pCFwrд+1;
% индекс левой границы пластины
pB_L=NL; pLB_L=pB_L-1; pRB_L=pB_L+1;
pLLB_L=pB_L-2; pRRB_L=pB_L+2;
% индекс правой границы пластины
pB_R=NR; pLB_R=pB_R-1; pRB_R=pB_R+1;
pLLB_R=pB_R-2; pRRB_R=pB_R+2;
```

Основную часть программы составляет цикл интегрирования уравнения теплопроводности по времени (фрагмент программы, связанный с вычислением мощности лазерного импульса, не приводится из-за его громоздкости и несущественности для изложения). Расчет производится, на-

чина с момента времени $t=0$, до окончания действия лазерного импульса $t=t_p$ с шагом Δt .

```
% вектор решения на слое n и слое n+1 по времени
TOld=T0*ones(ND,1); TNew=TOld;
% вектор правых частей
B=zeros(ND,1);
% интегрирование по времени
while 1
    % время
    t=t+dt;
    % тепловыделение в единице объема
    % (расчетное соотношение для мощности импульса не приводится)
    Q=(t<=tImpuls ? расчет : 0);
    % условие выхода
    if (t>=tImpuls)
        break
    end
    % обновление массива решения
    TOld=TNew;
end
```

Внутри цикла производится построение целочисленных векторов **indI** и **indJ**, задающих позиции элементов матрицы коэффициентов, а также вещественного вектора **valS**, определяющего числовые значения элементов этой матрицы. Компоненты вектора **valS** находятся в соответствии с используемой разностной схемой.

```
% объявление индексных векторов
indI=[]; indJ=[]; valS=[];
% граничное условие на левой границе области
% индексные вектора
indI=[indI;1;1];
indJ=[indJ;1;2];
valS=[valS;1;-1];
% правая часть
B(1)=0;
% газовая область слева от пластины
CCBack=(1+2*aGas*dt/(hx^2))*ones(length(pCBack),1);
CLBack=-(aGas*dt/(hx^2))*ones(length(pCBack),1);
CRBack=-(aGas*dt/(hx^2))*ones(length(pCBack),1);
% индексные вектора
indI=[indI;pCBack;pCBack;pCBack];
indJ=[indJ;pLBack;pCBack;pRBack];
valS=[valS;CLBack;CCBack;CRBack];
% правая часть
B(pCBack)=TOld(pCBack);
% граничное условие на левой границе пластины
CLB_L=lGas;
CCB_L=-(lGas+lFlat);
CRB_L=lFlat;
% индексные вектора
```

```

indI=[indI;pB_L;pB_L;pB_L];
indJ=[indJ;pLB_L;pB_L;pRB_L];
valS=[valS;CLB_L;CCB_L;CRB_L];
% правая часть
B(pB_L)=0;
% пластина
CCFlat=(1+2*aFlat*dt/(hx^2))*ones(length(pCFlat),1);
CLFlat=-(aFlat*dt/(hx^2))*ones(length(pCFlat),1);
CRFlat=-(aFlat*dt/(hx^2))*ones(length(pCFlat),1);
% индексные вектора
indI=[indI;pCFlat;pCFlat;pCFlat];
indJ=[indJ;pLFlat;pCFlat;pRFlat];
valS=[valS;CLFlat;CCFlat;CRFlat];
% правая часть
B(pCFlat)=TOld(pCFlat);
% область справа от пластины
CCFwrd=(1+2*aGas*dt/(hX^2))*ones(length(pCFwrd),1);
CLFwrd=-(aGas*dt/(hX^2))*ones(length(pCFwrd),1);
CRFwrd=-(aGas*dt/(hX^2))*ones(length(pCFwrd),1);
% индексные вектора
indI=[indI;pCFwrd;pCFwrd;pCFwrd];
indJ=[indJ;pLFwrd;pCFwrd;pRFwrd];
valS=[valS;CLFwrd;CCFwrd;CRFwrd];
% правая часть
B(pCFwrd)=TOld(pCFwrd);
% граничное условие на правой границе пластины
CLB_R=IFlat;
CCB_R=-(IGas+IFlat);
CRB_R=IGas;
% индексные вектора
indI=[indI;pB_R;pB_R;pB_R];
indJ=[indJ;pLB_R;pB_R;pRB_R];
valS=[valS;CLB_R;CCB_R;CRB_R];
% правая часть
B(pB_R)=-Q*hx;
% граничное условие на правой границе области
% индексные вектора
indI=[indI;ND;ND];
indJ=[indJ;ND-1;ND];
valS=[valS;-1;1];
% правая часть
B(ND)=0;

```

Сформированные данные используются для построения матрицы системы линейных алгебраических уравнений и ее решения методом прямого обращения [4].

```

% формирование разреженной матрицы
A=sparse(indI,indJ,valS);
% решение системы
TNew=A\B;

```

Пространственное распределение температуры внутри пластины графита и в окружающей пластину воздушной среде, приводится на рис. 15. Различные кривые отвечают разным моментам времени с начала воздействия на пластину лазерного импульса. Суммарная мощность лазерного импульса $Q_{\Sigma}=2.5$ Дж, а его длительность составляет $t_p=2.6$ мкс.

В силу малости температуропроводности пластины внутреннее поле существенно неоднородно. На границе пластины имеется локальное повышение температуры, связанное с падающим потоком энергии. По мере развития процесса происходит резкое увеличение температуры поверхности пластины на той ее стороне, которая обращена к источнику излучения (поток излучения падает на пластину справа налево). За время действия импульса температура поверхности пластины возрастает (кривые 1 и 2), достигает своего максимального значения к концу импульса (кривая 3), а после окончания действия импульса происходит ее падение за счет передачи тепла от пластины в окружающую среду (кривые 4, 5 и 6).

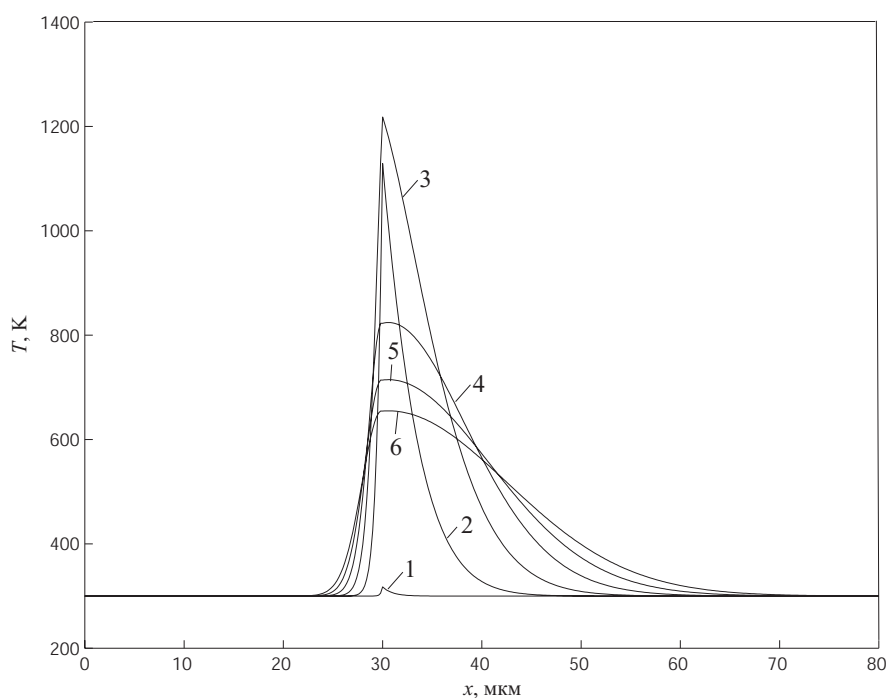


Рис. 15. Температурное поле около пластины графита при $Q_{\Sigma}=2.5$ Дж. Кривые соответствуют различным моментам времени $t=0.05$ (1); 1.09 (2); 2.13 (3); 3.17 (4); 4.21 (5); 5.25 мкс (6)

Заключение

Разработан подход к построению и реализации векторизованных алгоритмов решения краевых задач механики жидкости и газа в пакете MATLAB. В качестве конкретных примеров рассматриваются моделирование ползущего течения вязкой несжимаемой жидкости в канале с Т-

образной формой поперечного сечения и моделирование нагрева пластины лазерным излучением.

Разработанные алгоритмы применимы для решения более сложных задач механики жидкости и газа. К числу таких задач относятся задачи построения криволинейных сеток для областей сложной геометрической конфигурации; задачи о течении жидкости около тел произвольной геометрической формы при использовании переменных функция тока — вихрь скорости и криволинейных сеточных структур; задачи нестационарного нагрева пластины, цилиндра и шара лазерным импульсом; задачи расчета течений в каналах со вдувом при использовании переменных скорость — вихрь скорости; а также ряд других.

Созданные программные средства могут быть привлечены для поддержки научных исследований в различных областях, где применяются методы математического моделирования.

Программные разработки использованы на кафедре плазмогазодинамики и теплотехники Балтийского государственного технического университета «ВОЕНМЕХ» им. Д. Ф. Устинова при подготовке лабораторных практикумов по курсам «Численное моделирование в механике жидкости и газа», «Динамика вязкой жидкости», «Моделирование высокоинтенсивных процессов», «Математическое моделирование процессов в аэрокосмической технике», «Двухфазные течения».

Литература

1. Численное решение многомерных задач газовой динамики / Годунов С. К., Забродин А. В., Иванов М. Я., Крайко А. Н., Прокопов Г. П. / Под ред. С. К. Годунова.— М.: Наука, 1976.— 400 с.
2. Шлихтинг Г. Теория пограничного слоя.— М.: Наука, 1974.— 712 с.
3. Самарский А. А., Гулин А. В. Численные методы математической физики.— М.: Мир, 2000.— 316 с.
4. Dongarra J. J., Bunch J. R., Moler C. B., Stewart G. W. LINPACK User's Guide.— Philadelphia, 1979.
5. Волков К.Н., Емельянов В.Н. Взаимодействие интенсивных потоков излучения с газодисперсными системами // Математическое моделирование.— 2003.— Т.15.— №6.— С.35-40.

УДК 519.272

ОЦЕНИВАНИЕ ПАРАМЕТРОВ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ МЕТОДОМ МАКСИМАЛЬНОГО ПРАВДОПОДОБИЯ

Диаб Т. О.

Владимирский государственный университет, Владимир
e-mail:abo_nouran@hotmail.com

В работе рассматривается метод оценивания параметров логистической регрессии путем минимизации функции максимального правдоподобия. Для оценивания параметров использована система компьютерной математики (СКМ) MATLAB.

1. Построение логистической регрессии

В бинарную модель входит зависимая переменная S , принимающая два значения (обычно 0 и 1), а также непрерывная независимая переменная x , определяющая выбор. Обычная линейная регрессионная модель не подходит для описания этой ситуации, так как зависимая переменная имеет дискретное распределение. Логистическая модель (см. [1–3]) связывает S с x следующим образом:

$$\text{logit}(P(S = 1 | x)) = \ln \left\{ \frac{P(S = 1 | x)}{P(S = 0 | x)} \right\} = \sum_{k=0}^m a_k x^k.$$

Обозначим условную вероятность $P(S = 1 | x) = \pi(x)$, где $\pi(x)$ имеет вид:

$$\pi(x_i) = \frac{\exp(a_0 + a_1 * x_i + a_2 * x_i^2 + \dots + a_m * x_i^m)}{1 + \exp(a_0 + a_1 * x_i + a_2 * x_i^2 + \dots + a_m * x_i^m)}.$$

Задача построения логистической регрессии заключается в оценивании вектора параметров $A = [a_0, a_1, \dots, a_m]$. Как известно, оценкой максимального правдоподобия является значение оцениваемых параметров, которое максимизирует вероятность события, при котором наблюдения, сгенерированные с подстановкой оцениваемых параметров, совпадают с действительными значениями наблюдений.

Для оценивания параметров логистической регрессии известны оценки максимального правдоподобия, которые максимизируют функцию правдоподобия:

$$L(A) = \prod_{i=1}^n \exp \left(s_i \sum_{k=0}^m x_i^k a_k \right) \left(1 + \exp \left(\sum_{k=0}^m x_i^k a_k \right) \right)^{-1}.$$

Оценка \hat{A} максимального правдоподобия вектора параметров A находится путем решения уравнений максимального правдоподобия, полученных дифференцированием $L(A)$ по A .

Для упрощения вычислений переходят от функции правдоподобия к ее логарифму $\ln L(A)$. Такое преобразование допустимо, так как функция правдоподобия — положительная функция, и она достигает максимума в той же точке, что и ее логарифм. Оценки максимального правдоподобия находят из системы уравнений.

$$f_k = \frac{\partial \ln L(A)}{a_k} = \sum_{i=1}^n s_i x_i^k - \pi(x_i) x_i^k = 0 \quad k = \overline{0, m}. \quad (1)$$

Для проверки того, что точка оптимума соответствует максимуму функции правдоподобия, необходимо найти вторую производную от этой функции. И если вторая производная в точке оптимума отрицательна, то найденные значения параметров максимизируют функцию.

Задачи (1) может решаться численно с использованием какого-либо итерационного процесса. Соответствующая система нелинейных уравнений решается методом Ньютона-Рафсона (см. [4]). В методе Ньютона-Рафсона очередное приближение ОМП определяется соотношением

$$\hat{A}^{(k+1)} = \hat{A}^{(k)} - J^{-1}(\hat{A}^{(k)}) \nabla \ln L(\hat{A}^{(k)}),$$

где вектор $\nabla \ln L(\hat{A}^{(k)})$ — градиент логарифма функции правдоподобия с компонентами $\frac{\partial \ln L(A)}{\partial a_k} = 0$, $k = \overline{0, m}$. В методе Ньютона-Рафсона матрица

$J(A)$ представляет собой матрицу вторых частных производных логарифма функции правдоподобия с элементами

$$J_{kl}(A) = \frac{\partial^2 \ln L(A)}{\partial A_k \partial A_l} = - \sum_{i=1}^n x_i^{(k+l)} \pi(x_i) (1 - \pi(x_i)) \quad k, l = \overline{0, m}.$$

Итак, нахождение оценок максимального правдоподобия включает следующие этапы: построение функции правдоподобия (ее натурального логарифма); дифференцирование функции по искомым параметрам и составление системы уравнений; решение системы уравнений для нахождения оценок; определение второй производной функции, проверку ее знака в точке оптимума первой производной и формирование выводов.

Оценивание параметров логистической регрессии можно реализовать с помощью следующего программы для система MATLAB.

```
function xnew=NewRaph(x,s,A)
% Эта функция оценивает параметры логистической регрессии
Flag=1;
xold=A;
while Flag
    [F,J]=FDF(xold,x,s);
    xnew=xold-(inv(J)*F');
```

```

if mean((xnew-xold).^2)<=(10E-7)
    Flag=0;
else
    xold=xnew;
end
end
function [F,J]=FDF(A,x,s)
% Эта функция оценивает  $F = [f_0 \ f_1 \ \dots \ f_m]^T$ 
% и Jacobian матрица  $J$ 
if size(A,2)>size(A,1)
    A=A';
end
if size(x,2)>size(x,1)
    x=x';
end
if size(s,2)>size(s,1)
    s=s';
end
n=length(x);
m=length(A)-1;
Af=zeros(n,m+1);
for i=0:m
    Af(:,i+1)=(x.^i);
end
Fi=Af*A;
Fexp=exp(Fi)./(1+exp(Fi));
for i=1:(m+1)
    xs(i)=sum((x.^(i-1)).*s);
end
for i=1:(m+1)
    F(i)=Fexp'*x.^(i-1)-xs(i);
end
for i=1:(m+1)
    for j=1:(m+1)
        J(i,j)=(Fexp-Fexp.^2)*x.^(i+j-2);
    end
end
end

```

2. Результаты машинного эксперимента

В машинном эксперименте объемы модельных выборок были выбраны 100, 200, 400, 800 и 1000 точек, которые имеют равномерное распределение на интервале (0;1). Графики истинных функций условной вероятности и аппроксимирующие функции показаны на рис. 1.

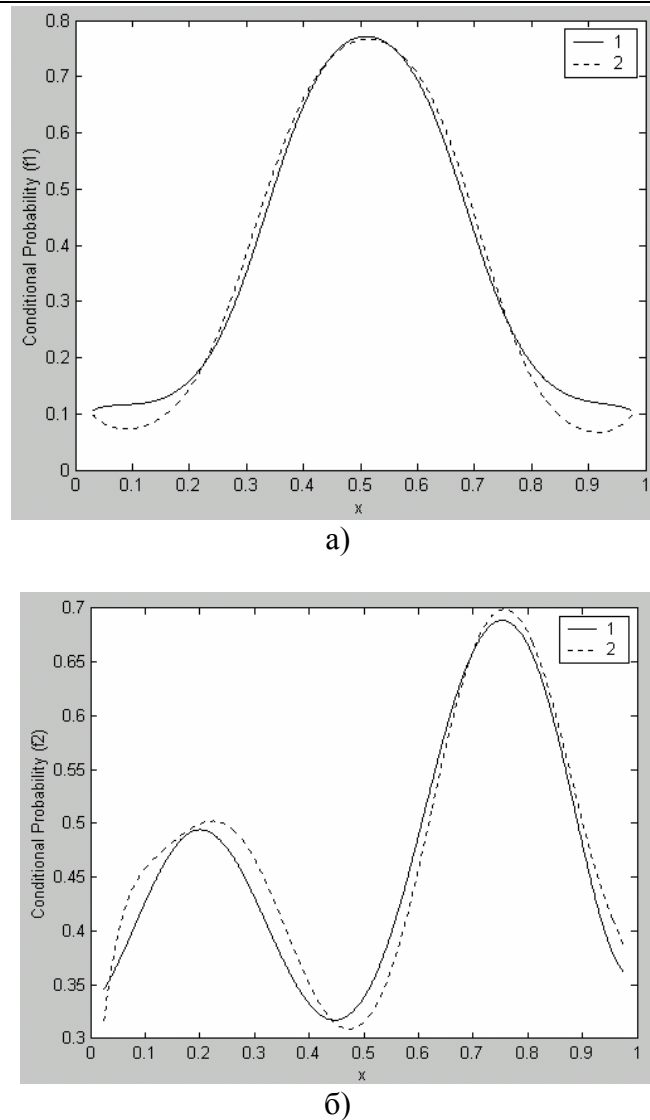


Рис. 1. Результаты аппроксимации условной вероятности: 1 — истинная функция условной вероятности; 2 — аппроксимирующая функция (ММП). а) модельная функция f_1 имеет колоколообразную форму, б) модельная функция f_2 имеет два экстремума,

Для определения расстояния между аппроксимирующей и модельной функциями использовались следующие критерии:

$$R = \frac{1}{n} \sum_{i=1}^n [P(S=1|x) - P^*(S=1|x)]^2.$$

Средние значения для каждого способа вычисления расстояния между модельной и аппроксимирующей функциями определялись по серии из 500 экспериментов. Результаты приведены в табл. 1. Их анализ показывает, что метод максимального правдоподобия дает меньшую погрешность для больших объемов модельных выборок.

Таблица 1.
Результаты вычисления среднего расстояния

Модель	n	\bar{R}
f_1	100	0.0059
	200	0.0041
	400	0.0026
	800	0.0016
	1000	0.0016
f_2	100	0.0170
	200	0.0147
	400	0.0077
	800	0.0043
	1000	0.0032

Литература

1. Кокс Д., Хинкли Д. Теоретическая статистика.— М.: Мир, 1978.— 560 с.
2. Cox D. R., Snell E. J. Analysis of binary data.— London: Chapman and hall, 1989.
3. Hosmer D. W., Lemshow S. Applied logistic regression.— New York: Wiley 1989.
4. Teukolsky S. A., Vetterling W. T. Numerical Recipes in C.— Cambridge: Cambridge university press, 1994.— 965 с.

УДК 515.2

МЕТОДЫ ИНТЕРАКТИВНОГО КОНСТРУИРОВАНИЯ ПАРАМЕТРИЧЕСКИХ СОСТАВНЫХ ПОВЕРХНОСТЕЙ

*Дубанов А. А.**Восточно-Сибирский государственный технологический университет,
e-mail: dubanov@esstu.ru*

1. Введение

Рассмотрим пространственный объект, который является результатом последовательных множественных операций над более простыми объектами, называемых базовыми элементами формы (БЭФ). Базовыми элементами формы могут быть многогранники, эллипсоиды, цилиндры, конусы, плоскости и т. д. Будем считать, что поверхность каждого БЭФ известна и задана в параметрическом виде. Представляется целесообразным разработать методику, позволяющую найти аналитическое представление поверхности такого составного параметрического объекта в векторном параметрическом виде. Это может быть востребовано в системах технического зрения, предназначенных для сравнения распознаваемых объектов с номенклатурными изделиями. В системах автоматизированного проектирования, когда является необходимым моделирование и анализ специальных линий на поверхности составного объекта. Это могут быть линии уровня, геодезические, линии тока, кривизны и т. д. Предполагается, что предложенные в данной статье методы и алгоритмы конструирования составных поверхностей, могут быть использованы в автоматизированной программной среде.

Будем считать известными параметрические уравнения поверхностей $\vec{P}(u_1, u_2)$ и $\vec{G}(t_1, t_2)$ двух пространственных объектов. Мы будем рассматривать замкнутые поверхности, определенные на прямоугольной области определения. Пусть существует линия пересечения наших поверхностей (иначе нет смысла их рассматривать), определенная системой уравнений $\vec{P}(u_1, u_2) = \vec{G}(t_1, t_2)$. Фактически, данная система уравнений определяет линии P , принадлежащую плоскости параметров (u_1, u_2) , и G , принадлежащую плоскости параметров (t_1, t_2) . Над объектами, поверхности которых заданы параметрическими функциями $\vec{P}(u_1, u_2)$ и $\vec{G}(t_1, t_2)$, нам необходимо совершить какую-либо из операций над множествами: объединение, пересечение, вычитание. Конечным результатом является параметрическое представление поверхности получившегося объекта.

Пусть на плоскости (u_1, u_2) существует область Ω , точки $\vec{P}(\Omega)$ будем считать удаленными из объекта $\vec{P}(u_1, u_2)$. А на плоскости (t_1, t_2) существует область Φ , точки $\vec{G}(\Phi)$ будем подставлять в объект $\vec{P}(u_1, u_2)$ вместо точек $\vec{P}(\Omega)$. Отметим, что границы областей Ω и Φ полностью определяются системой уравнений $\vec{P}(u_1, u_2) = \vec{G}(t_1, t_2)$. Тогда необходимо выполнить взаимно-однозначное преобразование областей Ω и Φ для того, чтобы позволить описание составного объекта по формуле [1]:

$$\vec{R}(u_1, u_2) = f_1(\Omega) \cdot \vec{P}(u_1, u_2) + f_2(\Omega) \cdot \vec{G}(t_1(u_1, u_2), t_2(u_1, u_2)), \quad (1.1)$$

где $f_1(\Omega) = \begin{cases} 1, & (u_1, u_2) \notin \Omega \\ 0, & (u_1, u_2) \in \Omega \end{cases}$, $f_2(\Omega) = \begin{cases} 0, & (u_1, u_2) \notin \Omega \\ 1, & (u_1, u_2) \in \Omega \end{cases}$.

В статье предложены алгоритмы конструирования составных параметрических поверхностей с использованием формулы (1.1). Предполагается, что будет создана интерактивная среда конструирования пространственных объектов, по образу и подобию CAD — систем (AutoCAD, ADEM), но в результате моделирования будет получена векторная функция поверхности пространственного объекта.

2. Разрезы плоскостями и параметризация поверхности объекта

Сутью метода является решение системы уравнений $\vec{P}(u_1, u_2) = \vec{G}(t_1, t_2)$, для определения линий P и G , после чего будут определены области Ω и Φ , чтобы стало возможным использование формулы (1.1). Математический аппарат несложен, но необходимо сделать так, чтобы все вышеперечисленные этапы выполнялись в автоматизированном режиме, с минимальным участием конструктора-разработчика.

Вспомним, к примеру, конструирование объектов в среде AutoCAD. Конструктор может расставлять объекты на рабочем поле при помощи мыши (опираясь на визуальную информацию), выполнить ввод опорных точек непосредственно из командной строки, а также может использовать механизм объектной привязки [2].

В данном параграфе приводится методика, позволяющая выполнить параметризацию поверхности пространственного объекта, после выполнения простого разреза [3]. Подразумевается, что поверхность объекта уже хранится в имеющейся базе данных. К вопросу организации базы данных и хранения поверхности пространственного объекта в ней мы вернемся чуть позднее.

Пусть в нашем распоряжении имеется уравнение поверхности некоторого объекта $\vec{P}(u_1, u_2)$. Если плоскость задана точкой и двумя векторами $\vec{G}(t_1, t_2) = \vec{C} + t_1 \cdot \vec{l}_1 + t_2 \cdot \vec{l}_2$, то линия пересечения поверхностей $\vec{P}(u_1, u_2)$ и $\vec{G}(t_1, t_2)$ удовлетворяет уравнению:

$$(\vec{P}(u_1, u_2) - \vec{C}) \cdot [\vec{l}_1 \times \vec{l}_2] = 0. \quad (2.1)$$

При использовании формулы (2.1) очень удобно производить определение области Ω (см. § 1). Практически, это сводится к анализу на знак формулы (2.1) в точке плоскости (u_1, u_2) .

2.1 Взаимно-однозначное соответствие сегментов плоскостей определения

При помощи уравнения (2.1) возможно определение областей Ω и Φ . Нужны дополнительные условия для того, чтобы выполнить биективное преобразование области Ω плоскости (u_1, u_2) в область Φ плоскости (t_1, t_2) . Это необходимо для определения зависимостей $t_1(u_1, u_2)$ и $t_2(u_1, u_2)$, используемых в (1.1). Таких преобразований можно предложить множество, но в данной статье для получения объекта, изображенного на рис. 1, использовалась центральная проекция точек отсекаемой части параллелепипеда на сегмент плоскости на соответствующую часть плоскости.

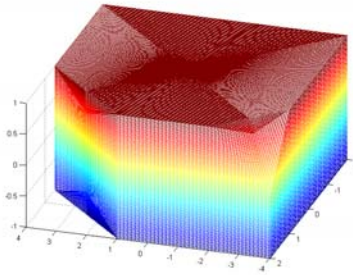


Рис. 1. Разрезы параллелепипеда.

Пусть \vec{S} является центром проецирования. Построение центральной проекции точек параллелепипеда на плоскость приводит к системе уравнений (2.1.1):

$$[(\vec{P}(u_1, u_2) - \vec{S}) \times (\vec{G}(t_1, t_2) - \vec{S})] = \vec{0}. \quad (2.1.1)$$

Система (2.1.1) разрешима относительно переменных t_1, t_2 :

$$\begin{aligned} t_1 &= \frac{\vec{l}_2 \cdot \left([\vec{P}(u_1, u_2) \times \vec{S}] - [(\vec{P}(u_1, u_2) - \vec{S}) \times \vec{C}] \right)}{\vec{l}_2 \cdot \left((\vec{P}(u_1, u_2) - \vec{S}) \times \vec{l}_1 \right)}, \\ t_2 &= \frac{\vec{l}_1 \cdot \left([\vec{P}(u_1, u_2) \times \vec{S}] - [(\vec{P}(u_1, u_2) - \vec{S}) \times \vec{C}] \right)}{\vec{l}_1 \cdot \left((\vec{P}(u_1, u_2) - \vec{S}) \times \vec{l}_2 \right)}. \end{aligned} \quad (2.1.2)$$

Для построения фигуры, изображенной на рис.1, центр проецирования помещался в начало координат, $\vec{S} = \vec{0}$. Тогда формулы (2.1.2) приобретают более простой вид:

$$t_1 = -\frac{\vec{l}_2 \cdot [\vec{P}(u_1, u_2) \times \vec{C}]}{\vec{l}_2 \cdot [\vec{P}(u_1, u_2) \times \vec{l}_1]}, t_2 = -\frac{\vec{l}_1 \cdot [\vec{P}(u_1, u_2) \times \vec{C}]}{\vec{l}_1 \cdot [\vec{P}(u_1, u_2) \times \vec{l}_2]}. \quad (2.1.3)$$

Напомним, что плоскость разреза задана уравнением: $\vec{G}(t_1, t_2) = \vec{C} + t_1 \cdot \vec{l}_1 + t_2 \cdot \vec{l}_2$.

Является характерным то, что параметры t_1, t_2 можно выразить через параметры u_1, u_2 не решая численно никаких уравнений. Существуют ограничения для использования аппарата центральной проекции при построении соответствий между сегментами поверхностей. Например, данный метод неприменим в том случае, когда центр проецирования находится в начале координат, а секущая плоскость проходит через него.

При построении взаимно-однозначных соответствий можно использовать также и аппарат ортогональной проекции. В этом случае необходимо решить систему уравнений (2.1.4) относительно переменных t_1, t_2 :

$$[(\vec{P}(u_1, u_2) - \vec{G}(t_1, t_2)) \times [\vec{l}_1 \times \vec{l}_2]] = \vec{0}. \quad (2.1.4)$$

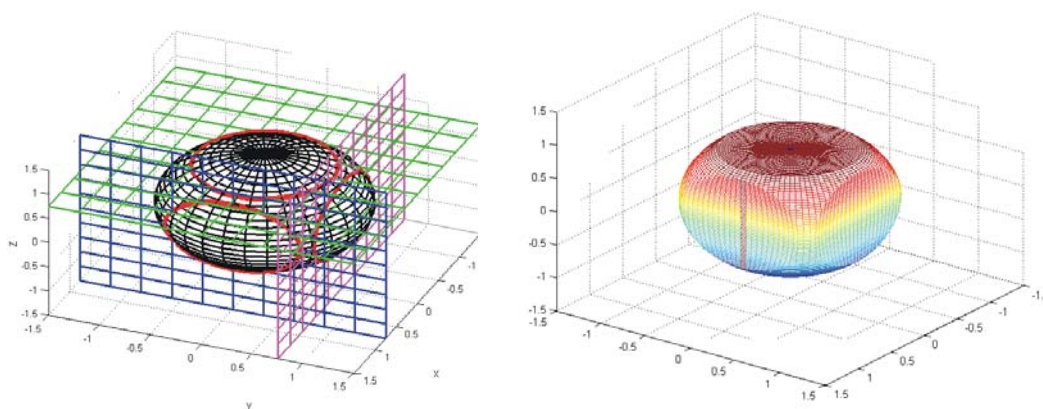


Рис. 2. Сфера и плоскости.

Поверхность фигуры, изображенной на рис. 2, получена именно таким способом. В этом случае вектор \vec{N} проецирования является нормалью секущей плоскости.

3. Формирование векторной параметрической функции

Зададимся вопросом о формировании в автоматизированном режиме векторной параметрической функции поверхности пространственного составного объекта. Топология конструируемых изделий может быть очень сложной. Помимо внешней оболочки объекта, в нем могут присутствовать различные полости. Сложно предложить универсальный метод, позволяющий выполнить параметризацию поверхности объекта любой топологии. В данной статье рассматриваются поверхности, каждой точке которой можно поставить в соответствие точку на сфере. Возможно поэтому, что

мы рассматриваем объекты, отображаемые на сферу, одно семейство параметрических линий напоминает «меридианы», а другое — «параллели». Тогда одну, отдельно взятую параметрическую линию можно трактовать замкнутую периодическую параметрическую функцию. И если искать приближенные аналитические выражения для координатных функций, то, на наш взгляд, решение искать следует среди тригонометрических функций, а именно — использовать разложение в ряд Фурье.

Если использовать разложение координатных функций в двумерный ряд Фурье по формулам (3.2), желательно чтобы координатные функции были периодическими по обоим параметрам. Но нами выбрана такая модель, когда точке поверхности сопоставляется точка сферы. А в семействе параметрических линий периодическими являются линии, похожие на параллели.

Рассмотрим это на следующем примере. Рассмотрим векторную параметрическую функцию $\vec{R}(u, v)$ от прямоугольной области определения. Периодическими являются только линии u , ($v = \text{const}$). Пусть функция $\vec{R}(u, v)$ является определенной на прямоугольнике где $u \in [0; 2\pi]$, а $v \in [0; \pi]$. Расширим область определения по параметру $v \in [0; 2\pi]$. После этого определим поверхность $\tilde{\vec{R}}(u, v)$, заданную формулами (3.1).

$$\tilde{\vec{R}}(u, v) = g_1(v) \cdot \vec{R}(u, v) + g_2(v) \cdot \vec{R}(f(u), 2\pi - v), \quad (3.1)$$

где

$$g_1(v) = \begin{cases} 1, & v \leq \pi \\ 0, & v > \pi \end{cases}, \quad g_2(v) = \begin{cases} 0, & v \leq \pi \\ 1, & v > \pi \end{cases}, \quad f(u) = \begin{cases} u + \pi, & u \leq \pi \\ u - \pi, & u > \pi \end{cases}.$$

С геометрической точки зрения это означает следующее: мы сформировали замкнутую кривую («меридиан») из двух частей, $L_1 = \vec{R}(u_{\text{const}}, v), v \leq \pi$ и $L_2 = \vec{R}(u_{\text{const}} + \pi, 2\pi - v)$, если $u_{\text{const}} \leq \pi$. Если $u_{\text{const}} > \pi$, то $L_1 = \vec{R}(u_{\text{const}}, v), v \leq \pi$ и $L_2 = \vec{R}(u_{\text{const}} - \pi, 2\pi - v)$.

Если окружность $x = \cos(t)$, $y = \sin(t)$ определить на отрезке $t \in [0; 4\pi]$, то график окружности несколько не изменится. Только мы будем знать, что окружность покрывается два раза при прохождении отрезка $t \in [0; 4\pi]$. Также и в случае с поверхностью, определенной формулами (3.1). Поверхность объекта будет покрыта два раза.

На рис. 3 изображено поведение координатной функции Y на области определения $u_1 \in [0; U_1], u_2 \in [0; 2 \cdot U_2]$, где $U_1 = 4$, $U_2 = 3$, фигуры, изображенной на рис. 1. А на рис. 4 изображено поведение функции Z на той же области определения. Изображение координатной функции X мы не приводим потому, что ее поведение очень похоже на поведение функции Y . Координатные функции были переопределены при помощи формул (3.1). И если произвести вывод на экран поверхности, определенной этими

функциями, то поверхность ничем не будет отличаться от поверхности, изображенной на рис. 1.

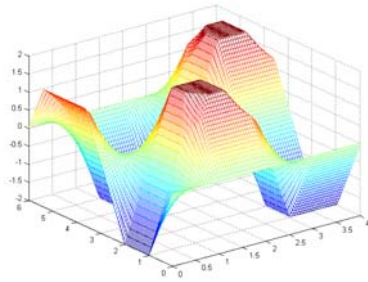


Рис. 3.

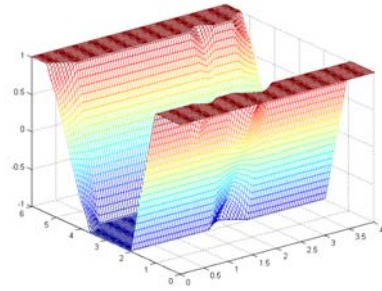


Рис. 4.

Приведем разложение координатной функции $R_x(u, v)$ в ряд Фурье :

$$\hat{R}_x(u, v) = \sum_{h_u, h_v} \alpha_x(h_u, h_v) \cdot e^{i(h_u \cdot u + h_v \cdot v)}, \quad (3.2)$$

$$\text{где } \alpha_x(h_u, h_v) = \frac{1}{4\pi^2} \cdot \int_0^{2\pi} \int_0^{2\pi} \tilde{R}_x(u, v) \cdot e^{-i(h_u \cdot u + h_v \cdot v)} du dv.$$

Напомним, что $\tilde{R}_x(u, v)$ является функцией, дополнительной к $R_x(u, v)$ согласно формулам (3.1). Напомним, что функция $R_x(u, v)$ определена на области $u \in [0; 2\pi]$, а $v \in [0; \pi]$, а функция $\tilde{R}_x(u, v)$ — на области $u \in [0; 2\pi]$, а $v \in [0; 2\pi]$.

При организации базы данных хранения поверхностей, является удобным хранить три матрицы комплексных коэффициентов $\alpha_x(h_u, h_v)$, $\alpha_y(h_u, h_v)$ и $\alpha_z(h_u, h_v)$, которые характеризуют поверхность.

4. Об определении линии пересечения поверхностей

Рассмотрим систему уравнений $\vec{P}(u_1, u_2) = \vec{G}(t_1, t_2)$, которую необходимо решить, чтобы найти линию пересечения двух поверхностей. Если для поверхности, заданной уравнением $\vec{G}(t_1, t_2)$, существует неявное аналитическое представление $\mathbf{G}(G_x(t_1, t_2), G_y(t_1, t_2), G_z(t_1, t_2)) = 0$, тогда для формирования области Ω (п.1) на плоскости параметров (u_1, u_2) , необходимо анализировать на знак выражение $\mathbf{G}(P_x(u_1, u_2), P_y(u_1, u_2), P_z(u_1, u_2))$. А для моделирования биективных соответствий между сегментами поверхностей уже требуется параметрическое представление. Случай центральной или ортогональной проекции является наиболее простым. Если же при такой модели соответствия точек сегментов поверхностей невозможно достичь при помощи проецирующих прямых, то следует использовать криволинейные проецирующие. На наш взгляд решение следует искать в теории полей,

решении уравнений математической физики численными методами с использованием метода конечных элементов (4).

Иметь и неявное представление, и параметрическое возможно лишь в очень частных случаях. Но с другой стороны, если взять систему автоматизированного проектирования AutoCAD, и рассмотреть инструментальный твердотельного проектирования, там библиотека базовых элементов состоит из сферы, параллелепипеда, конуса, пирамиды, тора, цилиндра и призмы. Мы думаем, что множество поверхностей, представляемых и параметрическом виде, и в неявном, больше на несколько порядков, чем перечисленные выше базовые элементы. Хотя неизвестно, каким образом можно записать уравнение поверхности параллелепипеда в неявном виде, или конуса или цилиндра вместе с основаниями. Но нет сложностей с написанием отдельных программных модулей для каждого из перечисленных базовых элементов.

Мы хотели бы обратить внимание читателя на следующий момент. Рассмотрим поверхность, полученную в результате объединения шаров одинакового радиуса (рис. 5–8).

Линия пересечения двух сфер есть окружность (если они пересекаются), как бы они в пространстве не располагались. Поведение линий P и G на плоскостях параметров (u_1, u_2) и (t_1, t_2) уже зависит от взаимного расположения сфер. На рис. 5 линия пересечения не пересекает «нулевых меридианов» обеих сфер, характерное поведение линий P и G изображено на рис. 6.

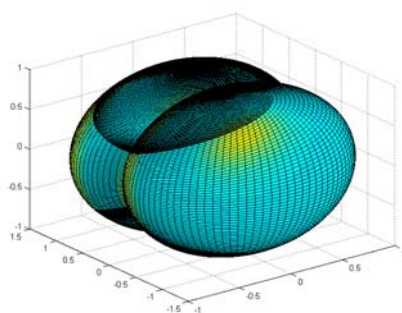


Рис. 5 Две сферы.

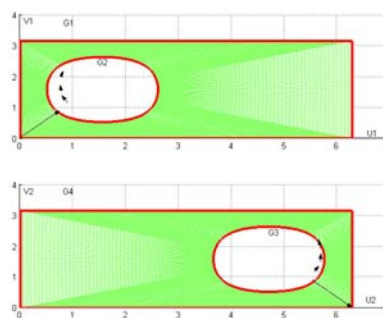


Рис. 6. К комбинации двух сфер.

Сфера допускает и неявное и параметрическое представление, но мы не будем использовать этот момент. Нашей целью является продемонстрировать поведение линий P и G на соответствующих плоскостях параметров в зависимости от взаимного расположения объектов. Для этого мы взяли такой симметричный объект как сфера.

Рассмотрим рис. 5, решение системы уравнений для определения линий пересечения дает следующую картину (рис. 6). Отметим следующее, что линия пересечения не пересекает «нулевого меридиана».

Рассмотрим рис. 7, в данном случае сферы пересекаются по параллелям. У составной поверхности, изображенной на рис. 8, есть такая часть сферы, у которой пересекается «нулевой меридиан».

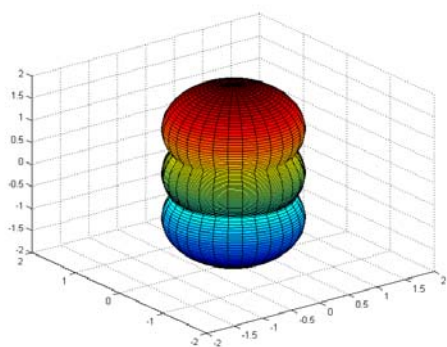


Рис. 7 Пересечение по линии $v=Const$ (параллель).

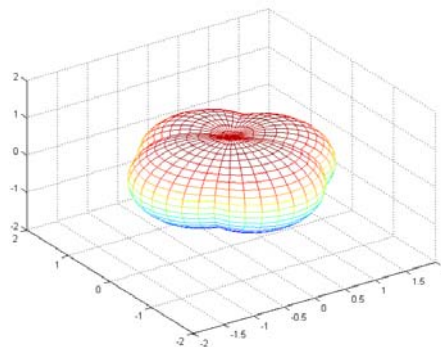


Рис. 8 Четыре сферы.

В формуле (1.1) рассчитываются функции $f_1(\Omega)$ и $f_2(\Omega)$. Интересно посмотреть на их поведение в зависимости от расположения пересекающихся объектов. На рис.9 функции f_{11} и f_{12} получаются такими, когда не пересекается «нулевой меридиан».

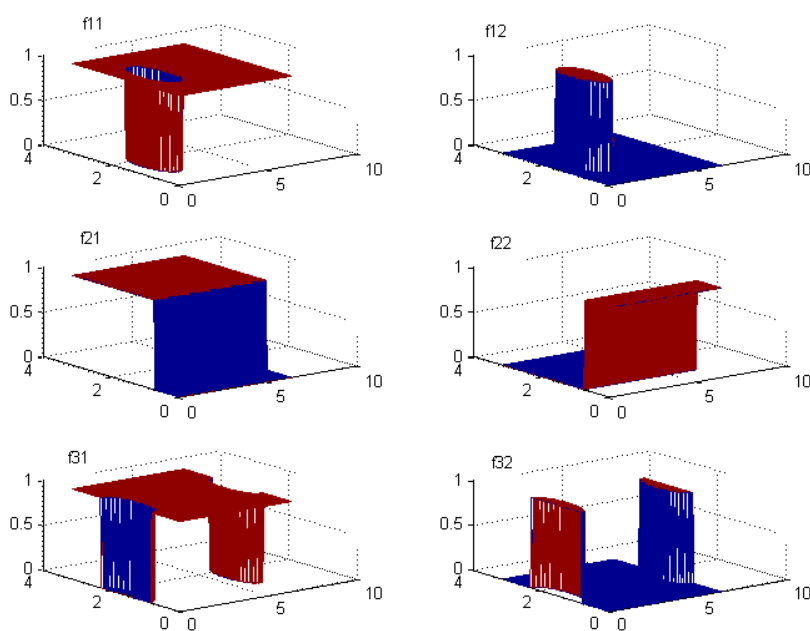


Рис. 9 Характерные прообразы линии пересечения на плоскости определения.

Функции f_{21} и f_{22} получаются тогда, когда точка пересечения «меридианов» («полюс») находится внутри линии пересечения. Типичный случай приведен на рис.7 - линией пересечения является «параллель».

Функции f_{31} и f_{32} получаются тогда, когда линия пересечения пересекает «нулевой меридиан».

4.1 Моделирование параметрической сетки

Рассмотрим следующий случай, который мы приводим для того, чтобы показать, что конструктор сможет в интерактивном режиме нанести на объект желаемую параметрическую сеть. Рассмотрим параметризацию объекта, который получается в результате объединения двух параллелепипедов.

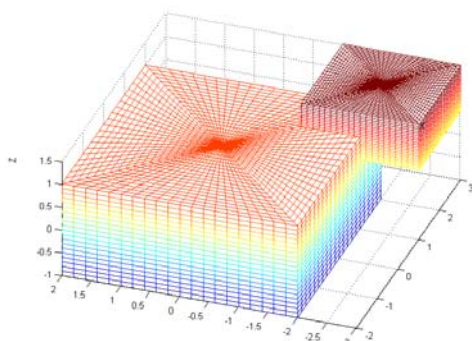


Рис. 10 Начальные условия.

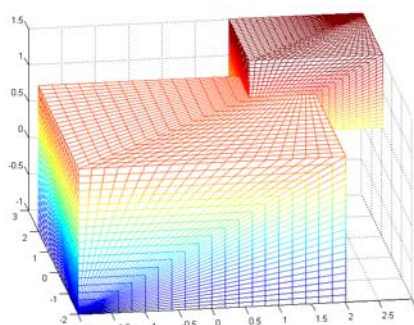


Рис. 11 Объединение параллелепипедов.

Рассмотрим рис. 10, необходимо получить векторную параметрическую функцию, задающую поверхность объединения двух параллелепипедов. Конечно, можно решить систему уравнений равенства координатных функций поверхностей обоих объектов, найти линию пересечения параллелепипедов и ее прообразы на плоскостях определения, а также установить биективное соответствие между точками прообразов.

На этом примере мы покажем, что в некоторых частных случаях легче построить другую сеть параметрических линий на поверхностях (см. рис. 11). Первое, что было сделано при такой параметризации (рис. 11), — это совмещение «нулевых меридианов», на рис. 12 он помечен стрелкой. Тогда при использовании формулы (1.1) поведение функций $f_1(\Omega)$ и $f_2(\Omega)$ будет «похоже» на поведение функций f_{21} и f_{22} (см. рис. 9).

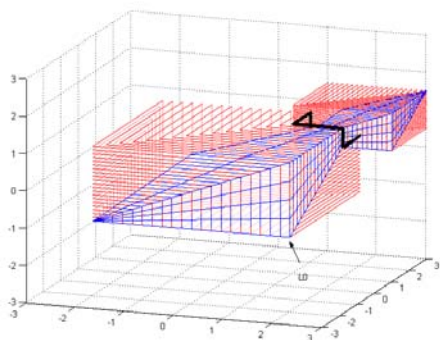


Рис. 12. Нулевой меридиан и линия пересечения.

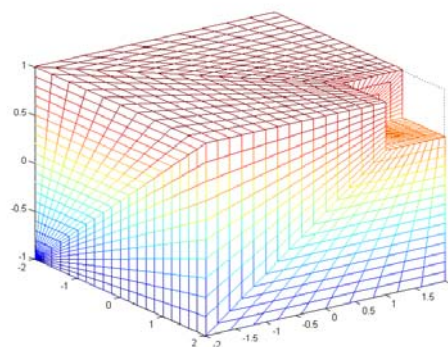


Рис. 13. Вычитание параллелепипедов.

При такой сети параметрических линий не только удобно выполнять различные множественные операции, но и управлять плотностью параметрических линий, сравните рис. 14 и 15.

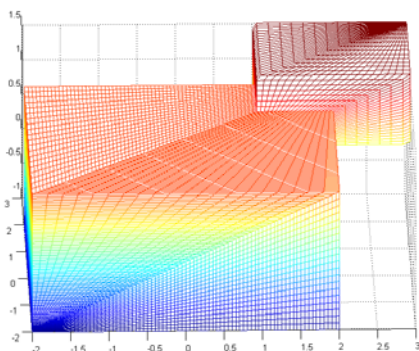


Рис. 14. Плотность линий мала на малом объекте.

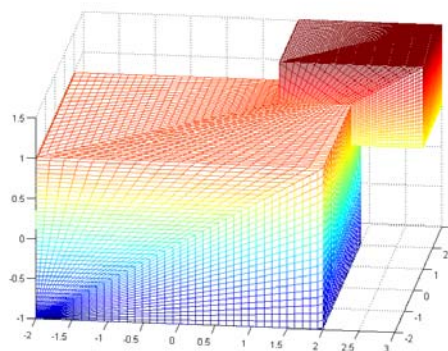


Рис. 15. Плотность линий на малом объекте увеличена.

Литература

1. Дубанов А. А. Конструирование базовых элементов формы и параметризация составных поверхностей // Информационные технологии.— 2003.— №10.
2. Полещук Н., Савельева В. Самоучитель AutoCAD 2004.— СПб.: BHV-СПб, 2003.— 640 с.
3. Попова Г. Н., Алексеева С. Ю. Машиностроительное черчение. Справочник.— Л.: Машиностроение, Ленингр. отд., 1986.— 447 с.
4. Шмелев В. Е. FEMLAB 2.3. Замечания по версии, (<http://www.MATLAB.ru/femlab/book3/default.asp>).

УДК 535.341:681.3.06

ПРИМЕНЕНИЕ MATLAB В СПЕКТРОСКОПИИ

Дьяченко Г. Г.

Физический институт им. П. Н. Лебедева РАН, Москва,
e-mail: dyatch@sci/lebedev.ru

Система компьютерной математики MATLAB идеально приспособлена для решения задач как теоретической, так и экспериментальной молекулярной спектроскопии. Так задача расчета частот и форм колебаний молекулы формулируется следующим образом. Необходимо сформировать две матрицы: матрицу кинематических коэффициентов T и матрицу силовых постоянных U , и решить уравнение Шредингера с гамильтонианом

$$\hat{H} = 1/2(\hat{p}'T\hat{p} + q'Uq),$$

где q — вектор-столбец обобщенных координат, а p — вектор-столбец сопряженных импульсов. В гармоническом приближении в данном уравнении можно произвести разделение переменных и привести гамильтониан к виду

$$\hat{H} = 1/2(\hat{P}'\hat{P} + Q'\Lambda Q),$$

где Q — нормальные координаты, а P — сопряженные им импульсы, Λ — диагональная матрица. Это можно сделать путем одновременного приведения пары квадратичных форм (T и U) к диагональному виду [1]. Преобразование матриц удобно проводить через декартово представление, обобщенные координаты связаны с декартовыми посредством соотношения

$$\delta q = B \delta r.$$

Матрица кинетической энергии диагональна в этом представлении:

$$T' = E = M^{-1},$$

где M — диагональная матрица атомных масс, и умножением слева и справа на $E^{-1/2}$ приводится к единичной. При этом матрица силовых постоянных в масс-взвешенных декартовых координатах приобретает вид:

$$U^m = E^{1/2} B' U B E^{1/2}.$$

Дальнейшая задача сводится к диагонализации матрицы силовых постоянных.

Удобство использования системы MATLAB на данном этапе обусловлено встроенным механизмом работы с разреженными матрицами. Дело в том, что матрица B , а в валентно-силовом приближении и матрица U , являются сильно разреженными. Основная трудность заключается в формировании самих матриц B и U . Для облегчения работы нами была использованы объектно-ориентированные возможности системы MATLAB. А именно, были разработаны классы **atom**:

```

function a = atom(ain,nm)
if nargin == 0
    a.num = [];
    a.type = [];
    a.xyz = [0 0 0]';
    a.mass = [];
    a.name = '';
    a = class(a, 'atom');
elseif isa(ain, 'atom')
    a = ain;
elseif isa(ain, 'cell')
    a=cell2struct(ain,{'num','type','xyz','mass','name'},1);
    a = class(a, 'atom');
else
    ain=ain(:);
    a.num = ain(1);
    a.type = ain(2);
    a.xyz = ain(3:5);
    a.mass = ain(6);
    a.name = '';
    a = class(a, 'atom');
end
if nargin==2
    a.name=nm;
end

```

Основными характеристиками объектов этого класса являются: атомный номер (num); тип (type), зависящий от окружения; координаты (xyz); массу атома (mass) и название (name). Этот класс снабжен вспомогательными функциями, основными из которых являются move и rotate, позволяющие преобразовывать координаты атомов в соответствии с заданными сдвигами и поворотами вокруг начала координат. Другой класс — **molle**:

```

function mo = molle(moin)
if nargin == 0
    mo.atoms = [];
    mo.adj = [];
    mo.q.type = [];
    mo.q.dq = [];
    mo.q.index = [];
    mo.u = cell(0);
    mo = class(mo, 'molle');
elseif isa(moin, 'atom')
    if length(moin) == 1
        mo.atoms = moin;
        mo.adj = sparse(1,1,0,1,1);
        mo.q.type = [];
        mo.q.dq = [];
        mo.q.index = [];
        mo.u = cell(0);
    end
end

```

```

    mo = class(mo, 'molle');
elseif length(moin) == 2
    mo.atoms = moin;
    mo.adj = sparse(1,2,1,2,2);
    mo.q = dq(1,coord(moin));
    mo.q.index = [1; 2];
    mo.u = cell(0);
    mo = class(mo, 'molle');
else
    error('Too many atoms. Can not initialise molle.');
```

```

end
elseif isa(moin, 'molle')
    mo = moin;
else
    error('Unknown object. Can not initialise molle.');
```

```

end
```

позволяет объединять атомы в молекулы. Объекты этого класса содержат: массив атомов (*atoms*), составляющих молекулу; матрицу смежности (*adj*), описывающую связи в молекуле, описание естественных (обобщенных) координат (*q*) и массив ячеек, содержащих производные по параметрам матрица силовых постоянных (*u*). Такое хранение матрицы силовых постоянных подразумевает, что параметры задаются отдельно, и содержит главную информацию об эквивалентности или соотношении параметров, описывающих взаимодействия в разных частях молекулы, удобно при решении обратной задачи, когда по экспериментальному спектру определяются силовые постоянные.

Основными функциями этого класса наряду с функциями, обеспечивающими перемещение молекулы (**move** и **rotate**) являются функции, позволяющие добавлять естественные координаты к модели молекулы (**addq**) и производные силовых постоянных (**addu**). Эти функции используются при первоначальном построении элементарных фрагментов, содержащих все необходимые для построения молекулы данные о геометрии и силовых постоянных. В случае ароматических бензоидных углеводородов таких фрагментов всего два (рис. 1):



Рис. 1. Фрагменты, необходимые для построения моделей ароматических бензоидных углеводородов.

Имея эти фрагменты для построения полной модели молекулы достаточно использовать функцию **glue**, которая позволяет склеивать фрагменты лишь путем указания соответствующих атомов. При этом автоматически формируется матрица смежности объединенного фрагмента, объединенная система естественных координат и силовых постоянных.

В специфическом случае ароматических молекул после окончательного построения молекулы дополнительно используется функция **addpi**, формирующая делокализованную по всей молекуле систему силовых постоянных. Кроме того имеются блоки, позволяющие учесть природу электронного состояния и в соответствии с ним модифицировать геометрию молекулы. Подробно методика расчетов изложена в работе [2], а алгоритм расчета интенсивностей вибронных переходов с использованием Symbolic Toolbox в работе [3].

Далее в нашей работе на примере тонкоструктурного спектра флуорена рассмотрены преимущества сравнительно недавно появившегося метода подавления шумовой составляющей на основе дискретного одномерного вейвлет-преобразования [4].

Тонкоструктурный спектр возбуждения флуорена (рис. 2) получался путем возбуждения лазерным излучением с перестраиваемой длиной волны молекул флуорена, охлажденных до единиц градусов Кельвина, и измерения интенсивности возникающей люминесценции и опорного сигнала при 20 отсчетах на каждой длине волны, окончательное значение интенсивности получалось путем обработки данных по методу наименьших квадратов. Охлаждение осуществлялось при адиабатическом истечении паров флуорена в вакуум сквозь маленькое отверстие, схема установки описана в работе [5]. Вследствие маленькой концентрации молекул в истекающей струе сигналы с фотоприемника необходимо значительно усиливать, поэтому спектры получаются зашумленные, и возникает проблема очистки от шумов и выделения слабых спектральных линий. Спектр шумов измерялся в тех же условиях при отсутствии струи.

До недавнего времени мы в основном пользовались методами сглаживания кривых, встроенными в систему Origin [6]. На рис. 3 приведен пример такого сглаживания участка спектра методами скользящего среднего (**S5AA**) и с применением Фурье фильтра (**S5FT**) встроенными в систему Origin. В глаза бросаются недостатки такого сглаживания: пропадает тонкая структура спектра, изменяется высота интенсивных линий, а применение Фурье фильтра приводит к дополнительным осцилляциям вблизи интенсивных линий (эффект Гиббса).

Упомянем также, что с помощью локальной полиномиальной регрессии, также встроенной в Origin, тонкая структура спектра сохраняется, однако заметного сглаживания не происходит.

Применение дискретного одномерного вейвлет-преобразования для сглаживания кривых позволяет в значительной мере избежать перечисленных выше недостатков.

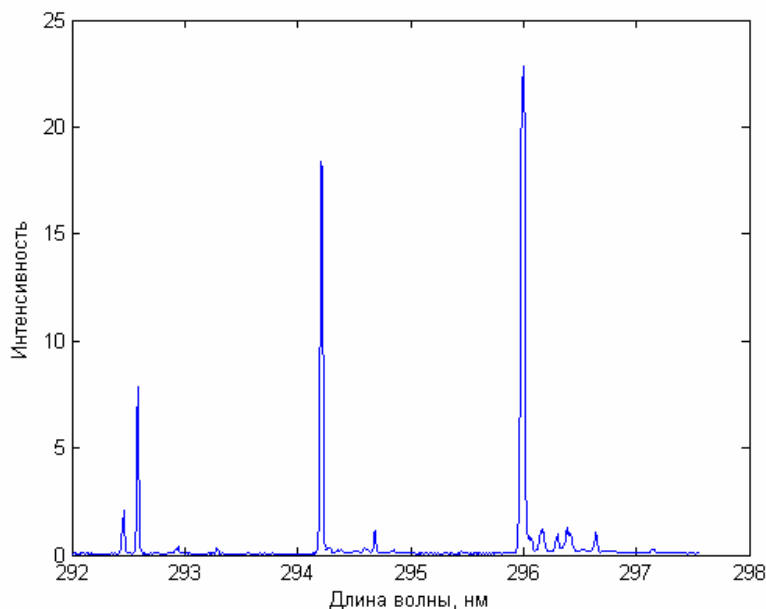


Рис.2. Спектр возбуждения флуорена

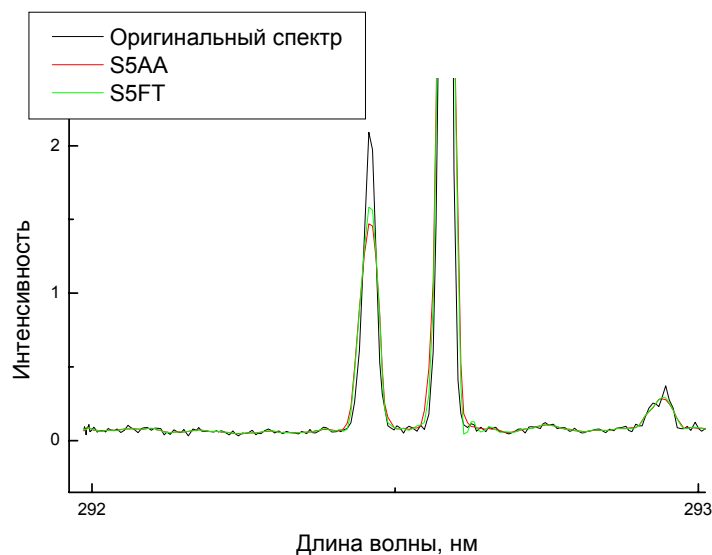


Рис.3. Спектр возбуждения флуорена и его сглаживание в системе Origin.

Дело в том, что техника вейвлет-преобразований как раз приспособлена для выявления локальных особенностей спектров, одновременно с этим позволяет убирать шумовую составляющую без заметного искажения интенсивных линий. Эта техника обладает большой гибкостью в связи с

тем, что существует большое количество вейвлетов, поэтому необходим предварительный анализ для выяснения, какой тип в наших условиях дает наилучшие результаты.

С нашей точки зрения такой анализ удобно проводить на основе измеренного спектра шумовой составляющей. В качестве инструмента исследования нами выбрана система компьютерной математики (СКМ) MATLAB 6.5 с пакетом расширения Wavelet Toolbox. По обилию типов вейвлетов и функций для обработки сигналов этот пакет является лучшим среди пакетов расширения для СКМ в этой области [4]. Очистка спектров проводится с помощью «штрафного» метода Бирге-Массарта, с подробным описанием всех параметров можно ознакомиться по книге [4], немного переделанный текст примера из этой работы представлен ниже:

```
clear all
fname='tst17.dat' %Исходные данные
[xx,yy,sh,xxi,s2y]=spexmnk(fname); %Обработка спектра по методу наименьших квадратов
yd=[]; %Массив, накапливающий очищенные от шумов спектры
wname='sym5'; %Имя вейвлета
for i=4:6
    lev=i; %Уровень вейвлет-разложения
    [c,l]=wavedec(yy,lev,wname); %Структура вейвлет разложения
    sigma=wnoisest(c,l,1); %Стандартное отклонение белого шума Гаусса
    alpha=1; %Параметр настройки штрафного метода
    thr=wbmpen(c,l,sigma,alpha); %Глобальный порог для удаления шумов
    sorh='s'; %Установка жесткого порога
    keepapp=1; %Коэффициенты аппроксимации не могут ограничиваться
    yd=[yd;wdencmp('gbl',yy,wname,lev,thr,sorh,keepapp)]; %Очищенные от шума векторы
    % Отрисовка результатов
    subplot(410+i-3), plot(sh,yd(i-3,:)),
    title(['Очищенный спектр (wname=',wname,' lev=',int2str(lev),')']);
    ylabel('Инт. ');ylim([.1 .4]);
    set(gca,'XTickLabel',[]);
end
subplot(414), plot(sh,yy),title('Исходный спектр шумов');
xlabel('шаг');ylabel('Инт. ');ylim([.1 .4]);
```

В результате исследования всех типов ортогональных вейвлетов (Добеши и симлет) до уровня разложения $lev=6$ было выделено два типа вейвлетов **sym5** и **sym7**, использование которых дает наилучшие результаты, представленные на рис. 4 и 5.

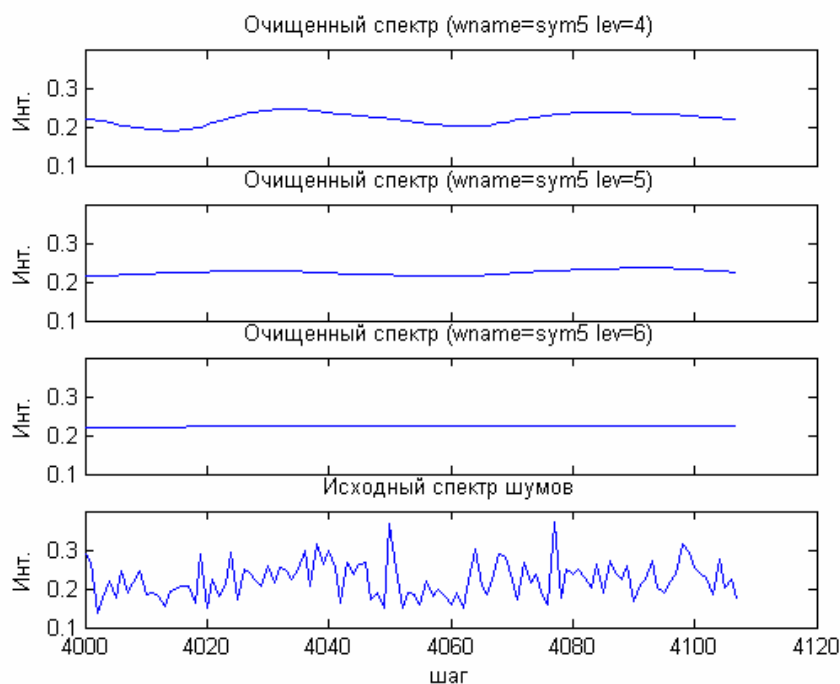


Рис. 4. Очистка спектра от шума с помощью вейвлета **sym5** для разных уровней разложения.

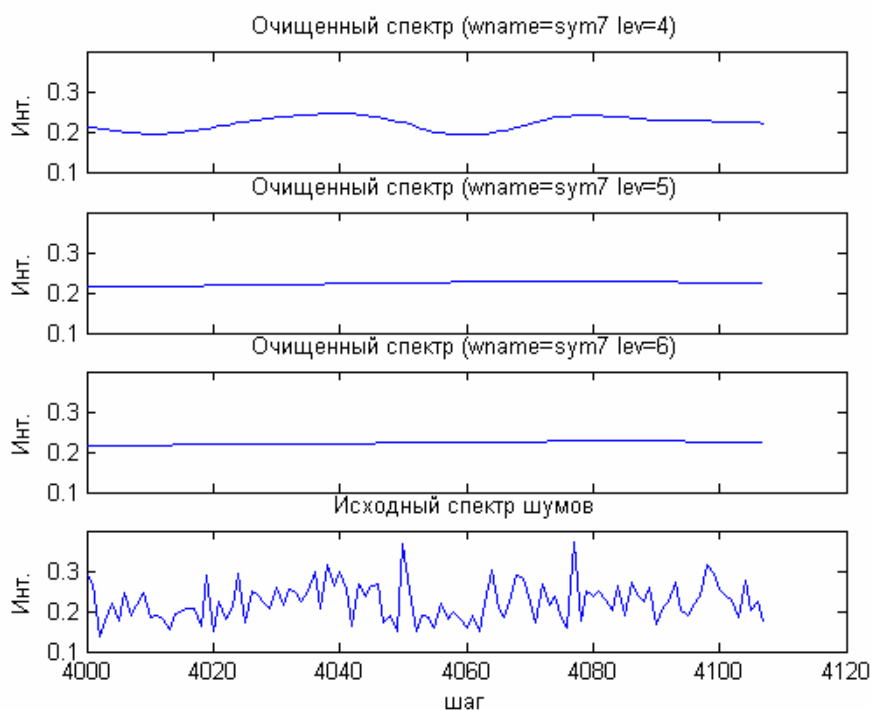


Рис. 5. Очистка спектра от шума с помощью вейвлета **sym7** для разных уровней разложения.

Наилучший результат среди приведенных дает вейвлет **sym7** на уровне разложения 6. Этот вариант мы использовали для обработки спектров возбуждения флуорена. Поскольку при измерении длины волны в общем случае сетка получается квазиравномерной из-за особенностей ме-

ханической части, мы предварительно перешли к равномерной сетке с помощью кусочной Эрмитовой кубической интерполяции, которая сохраняет монотонность и форму данных. Это связано с тем, что быстрое дискретное вейвлет-преобразование работает только на равномерной сетке. Далее мы использовали ту же технику, как при очистке спектра шумов. Текст программы представлен ниже.

```
clear all
fname='fl05.dat'
[xx,yy,sh,xxi,s2y]=spexmnk(fname);
m=length(yy);
xxi=xxi/2000;
linx=xxi(1):(xxi(end)-xxi(1))/(m-1):xxi(end);
liny=interp1(xxi,yy,linx,'cubic');
wname='sym7';
lev=6;
[c,l]=wavedec(liny,lev,wname);
sigma=wnoisest(c,l,1);
alpha=1;
thr=wbmpen(c,l,sigma,alpha);
sorh='s';
keepapp=1;
x=liny;
xd=wdencmp('gbl',x,wname,lev,thr,sorh,keepapp);
indx=1:length(x);
xmin=292;
xmax=293;
xyaxis=[xmin xmax 0 .2];
subplot(211), plot(linx,x),
title('Исходный спектр');xlabel('Длина волны, нм');
ylabel('Интенсивность');axis(xyaxis)
subplot(212), plot(linx,xd),
title(['Очищенный спектр (wname=',wname,' lev=',int2str(lev),')']);
xlabel('Длина волны, нм');ylabel('Интенсивность');axis(xyaxis)
```

Результаты вычислений представлены на рис. 6 и 7. Можно заметить, что наряду с заметным уменьшением уровня шумов изменения форма интенсивных линий не изменяется, а это очень важно при исследовании тонкоструктурных спектров.

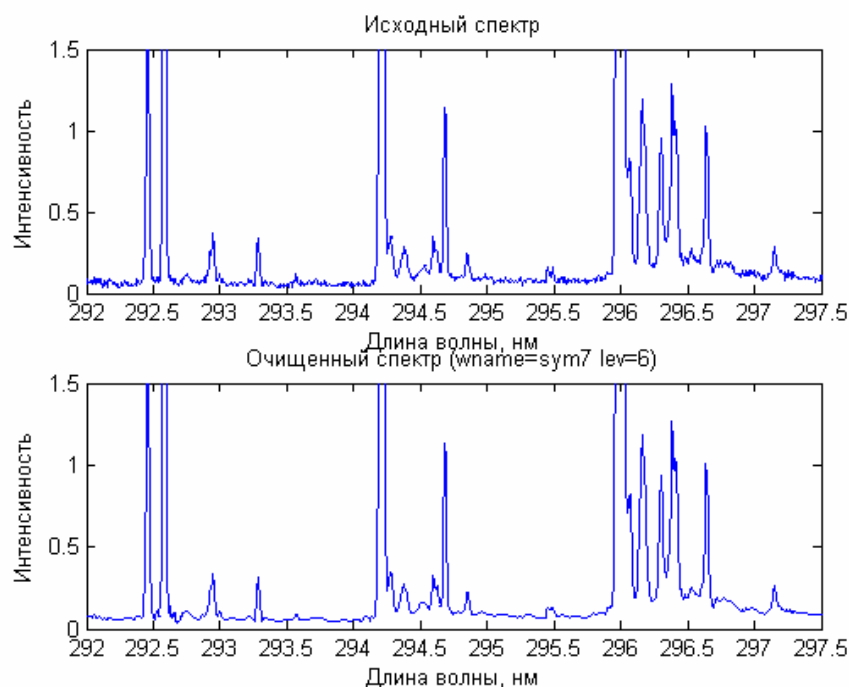


Рис. 6. Сравнение исходного спектра флуорена с очищенным.

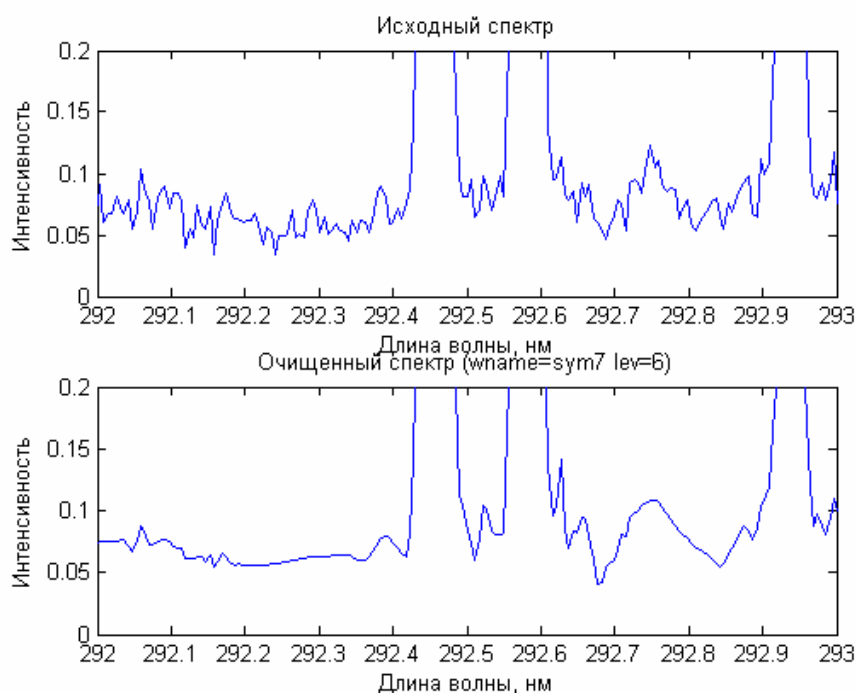


Рис. 7. Сравнение исходного и очищенного спектров флуорена в увеличенном масштабе.

Подавление шумовой компоненты на основе дискретного одномерного вейвлет-преобразования может в дальнейшем оказать неоценимую помощь при изучении спектров флуоресценции, в которых шумовая составляющая намного больше, чем в спектрах возбуждения.

Литература

1. Грибов Л. А., Павлючко А. И. Анггармонические методы решения задач в теории колебательных спектров молекул.— М.: Наука, 1998.— 334 с.
2. Борисевич Н. А., Водоватов Л. Б., Дьяченко Г. Г., Петухов В. А., Семенов М. А. Экспериментальное определение и расчет интенсивностей линий в электронно-колебательных спектрах струйно охлажденного антрацена // Оптика и спектроскопия.— 2000.— Т.89.— №2.— С.249–256.
3. Дьяченко Г. Г., Петухов В. А. Применение символьных вычислений при расчете электронно-колебательных спектров молекул // Журнал прикладной спектроскопии.— 2001.— Т.68.— №1.— С.142–144.
4. Дьяконов В. П. Вейвлеты. От теории к практике.— М.: СОЛОН-Р, 2002.
5. Борисевич Н. А., Водоватов Л. Б., Дьяченко Г. Г., Петухов В. А., Семенов М. А. Спектры возбуждения и флуоресценции пирена, охлажденного в сверхзвуковой струе // Опт. и спектроск.— 1995.— Т.78.— №2.— С.241–248.
6. <http://www.OriginLab.com>.

УДК 519.6

К ПРОБЛЕМЕ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ В СРЕДЕ MATLAB

Дьячук А. К., Медынский М. М.

Московский авиационный институт

(государственный технический университет), Москва,

e-mail: diatchouk@hotmail.ru, medmm@hotmail.com

Анализ содержания пакета прикладных программ для решения оптимизационных задач в среде MATLAB 6.5 (Optimization Toolbox for Use with MATLAB) и проведенные тестовые расчеты, показывают, что все используемые в данном пакете методы многомерной оптимизации являются «локальными», поскольку позволяют отыскать лишь локально оптимальные решения. Применение таких методов в случае мультимодальных целевых функций приводит, как правило, к неудовлетворительным результатам. В этой связи задача поиска глобального экстремума функции многих переменных является одной из самых актуальных задач численной оптимизации. Анализ известной литературы [1–9, 14, 17–22] подтверждает этот вывод.

В работе рассматриваются и предлагаются к применению в среде MATLAB 6.5 малоизвестные методы глобальной оптимизации, основанные на использовании так называемых равномерно распределенных последовательностей точек [10–14, 15, 16]. Метод ЛП-поиска, базирующийся на последовательностях, полученных в работах Соболя И. М. [11], — это «remake» метода, успешно применявшегося в инженерной практике в начале 80-х годов прошлого столетия, а равномерные последовательности точек, полученные в работах [15, 16], для решения задачи оптимизации используются впервые, и полученный метод назван авторами методом Холтона.

Оба алгоритма глобального поиска, рассмотренные в данной работе, реализованы в виде программ в среде MATLAB 6.5, скомпилированы в соответствующие исполняемые модули (MEX-файлы) и библиотеки dll, которые могут быть в дальнейшем использованы при написании прикладных программ на языках программирования высокого уровня, в частности на C++.

Результаты проверки работоспособности предлагаемых методов на примере типовых тестовых задач подтвердили их высокую эффективность.

Введение

Известно [1–6, 13, 14, 21], что многие задачи инженерной практики, связанные с поиском наилучшего решения из множества допустимых, сводятся к задаче нелинейного программирования следующего вида:
найти

$$\min_{\bar{x} \in R^n} f(\bar{x}),$$

при ограничениях в виде равенств и неравенств G :

$$\begin{cases} h_k(\bar{x}) = 0, & k = 1, 2, \dots, m, \\ g_k(\bar{x}) \leq 0, & k = m+1, m+2, \dots, p. \end{cases}$$

Целевая функция $f(\bar{x}) = f(x_1, x_2, \dots, x_n)$ определяет критерий для выбора наилучшего решения из множества альтернатив. Будем говорить о минимизации, поскольку максимизация $f(\bar{x})$ очевидно эквивалентна минимизации функции $(-f(\bar{x}))$. Ограничения задачи определяют допустимую область G , образующую так называемую область допустимых решений. Любой вектор \bar{x} , удовлетворяющий ограничениям, называется допустимым вектором (или допустимой точкой).

Допустимый вектор $\bar{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$, сообщающий минимум целевой функции $f(\bar{x})$, называется оптимальной точкой, а соответствующее значение функции $f(\bar{x}^*)$ — оптимальным значением целевой функции. Пара \bar{x}^* и $f(\bar{x}^*)$ составляет оптимальное решение. Различают локальное и глобальное оптимальное решение. В обоих случаях $f(\bar{x}^*) \leq f(\bar{x})$, но для глобального оптимального решения это соотношение выполняется для всех $\bar{x} \in G$; для локального оптимального решения соотношение имеет место только для малой окрестности точки \bar{x}^* .

Глобальный минимум — это оптимальное решение для всей области допустимых решений. Оно лучше всех других решений, соответствующих локальным минимумам, и именно его, как правило, требуется найти. Однако все известные и эффективно работающие в настоящее время численные методы оптимизации, решающие указанную выше задачу нелинейного программирования, являются локальными, т. е. определяющими лишь локально оптимальные решения [1–9]. Этот факт подтверждается и проведенными численными экспериментами (в том числе и с алгоритмами, используемыми в математическом пакете MATLAB 6.5).

1. Методы Холтона и ЛП-поиска

Методы Холтона и ЛП-поиска являются детерминированными аналогами глобального случайного поиска. Известно, что простейший глобаль-

ный поиск точки минимума \bar{x}^* целевой функции $f(\bar{x})$ состоит в том, что в допустимой области G выбирают некоторые пробные точки $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^N$, в каждой из них вычисляют значение целевой функции и отбирают точку, в которой целевая функция принимает наименьшее значение. Эта точка служит приближением к искомой точке \bar{x}^* глобального минимума целевой функции $f(\bar{x})$. Считается, что метод поиска сходится, если хотя бы одна из точек $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^N$ при достаточно большом N попадет в некоторую наперед заданную малую окрестность U точки глобального минимума целевой функции $f(\bar{x})$.

Пусть допустимая область G представляет собой единичный n -мерный куб $K^n = \{0 \leq x_j \leq 1, j = 1, 2, 3, \dots, n\}$. При случайном поиске в качестве пробных точек $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^N$ выбирают независимые случайные точки, равномерно распределенные в K^n . Вероятность того, что хотя бы одна точка попадет в малую окрестность точки минимума U , равна $P = 1 - (1 - U)^N$ и стремится к 1, когда $N \rightarrow \infty$, т. е. метод сходится.

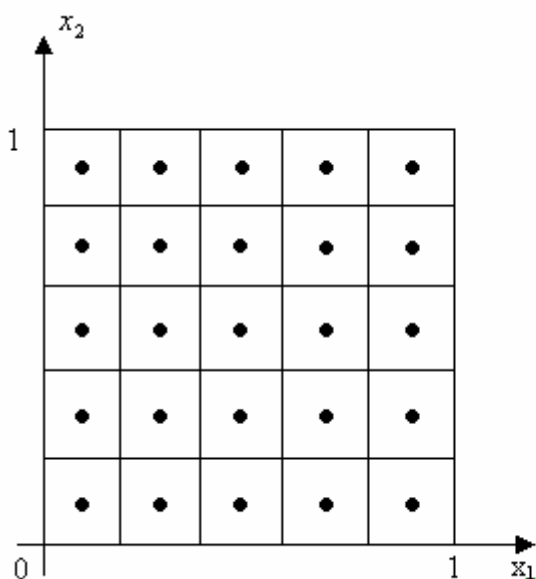


Рис. 1. «Равномерная» сетка при $n = 2$
($N = 25$).

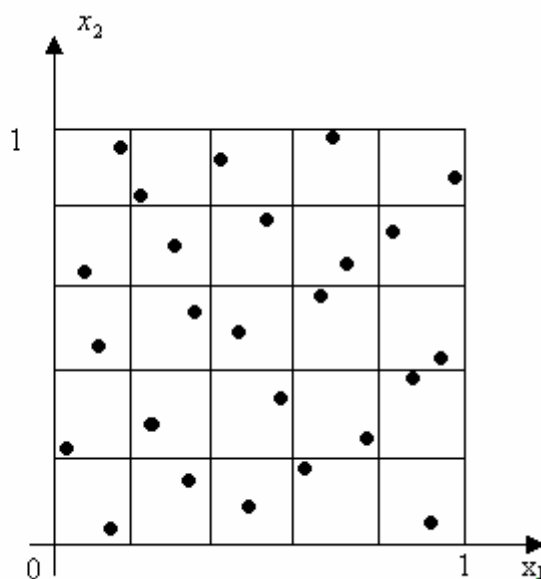


Рис. 2. Улучшенная сетка при $n = 2$
($N = 25$).

При неслучайном поиске в качестве пробных точек выбирают неслучайные точки $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^N$. На первый взгляд кажется, что точки надо выбирать равномерно по области K^n (рис. 1). Однако это не так [13, 14].

Предположим, что функция $f(\bar{x})$ зависит только от одного аргумента $f(\bar{x}) = f(x_1)$. Вычисляя ее значения в точках такой «равномерной» сетки, состоящей из $N = M^n$ точек (рис. 1), получаем всего M различных зна-

чений, среди которых будем выбирать наименьшее, т. е. $N - M$ точек просчитывались напрасно. Так, например, в случае исследования функции двух переменных $f(x_1, x_2)$, определенной в K^2 , которая сильно зависит только от x_1 , мы получим лишь пять различных значений, каждое повторенное пять раз (рис. 1). Если же $f(\bar{x})$ «существенно» зависит от $m < n$ аргументов, то напрасно будут считаться $N - M^m$ точек. Так, например, если $n = 4$, $m = 2$ и $M = 10$ количество «лишних» значений функции будет равно $N - M^m = M^n - M^m = 10^4 - 10^2 = 9900$ из общего числа $N = 10000$ вычисленных значений! Впервые такая особенность была отмечена в 1957 году в работе [10].

Случайные сетки лишены указанного недостатка, так как вероятность появления одинаковых абсцисс у нескольких случайных точек равна нулю. Но может быть эффективнее назначить пробные точки, а не отдавать их выбор на волю случая? Например, при $n = 2$ точки, изображенные на рис. 2, проецируются на каждую из координатных осей в N различных точек и, вычисляя значения функции в точках такой сетки, мы получим N различных значений, дающих гораздо лучшее представление о функции и диапазоне ее изменения.

Идея методов Холтона и ЛП-поиска заключается в использовании в качестве пробных точек точки так называемых равномерно распределенных последовательностей точек в K^n [11, 14, 15], а именно: последовательность Холтона $P_1, P_2, \dots, P_N, \dots$, построенной в [15, 16] и ЛП $_{\tau}$ -последовательность $Q_1, Q_2, \dots, Q_N, \dots$, построенной в работе [11]. Последовательность точек Холтона и ЛП $_{\tau}$ -последовательность являются детерминированными аналогами последовательности независимых случайных точек, равномерно распределенных в K^n . Глобальный поиск по точкам этих последовательностей сходится: доказано, что если $S_N(U)$ — число точек, попавших в малую окрестность U точки минимума целевой функции $f(\bar{x})$, то $S_N(U) \geq 1$, при достаточно большом N . Причем, чем более равномерно распределены точки, тем быстрее сходится процесс поиска. Считается, что точки ЛП $_{\tau}$ -последовательности являются наиболее равномерно распределенными среди всех известных в настоящее время последовательностей [11, 14]. (Вместе с тем, тестовые расчеты, результаты которых представлены ниже, не показали какого-либо существенного преимущества точек ЛП $_{\tau}$ -последовательности перед точками последовательности Холтона).

1.1. Последовательность Холтона

Если r_1, r_2, \dots, r_n — попарно взаимно простые числа, то последовательностью Холтона называется последовательность точек $\{P_i\}$ в K^n с декартовыми координатами $P_i = (p_{r_1}(i), p_{r_2}(i), \dots, p_{r_n}(i))$, $i = 1, 2, \dots$, где $p_r(i)$ — числовая последовательность, определяемая следующим образом.

Если в r -ичной системе счисления $i = a_m a_{m-1} \dots a_2 a_1$, то в той же r -ичной системе: $p_r(i) = 0, a_1 a_2 \dots a_{m-1} a_m$ (здесь все a_s — целые r -ичные цифры, т. е. равны одному из значений $0, 1, \dots, r-1$). В десятичной системе соответственно получаем:

$$i = \sum_{s=1}^m a_s r^{s-1}; \quad p_r(i) = \sum_{s=1}^m a_s r^{-s}.$$

Например, первые 9 значений $p_2(i), p_3(i), p_5(i), p_7(i), p_{11}(i)$ равны:

I	1	2	3	4	5	6	7	8	9
$p_2(i)$	1/2	1/4	3/4	1/8	5/8	3/8	7/8	1/16	9/16
$p_3(i)$	1/3	2/3	1/9	4/9	7/9	2/9	5/9	8/9	1/27
$p_5(i)$	1/5	2/5	3/5	4/5	1/25	6/25	11/25	16/25	21/25
$p_7(i)$	1/7	2/7	3/7	4/7	5/7	6/7	1/49	8/49	15/49
$p_{11}(i)$	1/11	2/11	3/11	4/11	5/11	6/11	7/11	8/11	9/11

Все такие последовательности равномерно распределены в K^n .

На практике, как правило, в качестве r_1, r_2, \dots, r_n выбирают первые n простых чисел: $r_1 = 2, r_2 = 3, r_3 = 5, \dots$ и используют n — мерные точки

$$P_i = (p_2(i), p_3(i), \dots, p_{r_n}(i)), i = 1, 2, 3, \dots$$

Например:

1) при $n = 2$, получаем последовательность Холтона вида:

$$P_1 = (1/2, 1/3), P_2 = (1/4, 2/3), P_3 = (3/4, 1/9), \dots, P_{10} = (5/16, 10/27), \dots;$$

2) при $n = 3$, получаем:

$$P_1 = (1/2, 1/3, 1/5), P_2 = (1/4, 2/3, 2/5), \dots, P_{10} = (5/16, 10/27, 2/25), \dots;$$

3) при $n = 4$, получаем:

$$P_1 = (1/2, 1/3, 1/5, 1/7), P_2 = (1/4, 2/3, 2/5, 2/7), \dots, \\ P_{10} = (5/16, 10/27, 2/25, 22/49), \dots$$

1.2. ЛП τ -последовательность

Если в двоичной системе счисления номер точки записывается как: $i = e_m e_{m-1} \dots e_2 e_1$, где $e_m, e_{m-1}, \dots, e_2, e_1$ — двоичные цифры, каждая из которых равна 0 или 1 (соответствующее разложение номера точки в десятичной системе счисления имеет вид):

$$i = 2^{m-1} e_m + 2^{m-2} e_{m-1} + \dots + 2e_2 + e_1,$$

то для всех $1 \leq j \leq n$:

$$q_j^i = e_1 V_j^1 * e_2 V_j^2 * e_3 V_j^3 * \dots * e_m V_j^m,$$

где $q_j^i, j = 1, 2, \dots, n$ — координаты точек Q_i ЛП τ -последовательности.

Векторы $\bar{V}^S = (V_1^S, V_2^S, \dots, V_n^S)$ легко вычисляются.

В работе [11] представлена таблица числителей координат точек \bar{V}^S , которые называются направляющими точками, для $1 \leq n \leq 13, 1 \leq s \leq 20$.

Знаменатели всех координат точки \bar{V}^s равны 2^s . Например, при $n = 7, s = 3$ имеем $\bar{V}^3 = (1/8, 5/8, 7/8, 1/8, 5/8, 7/8, 3/8)$. Таким образом, количество точек $N < 2^{21}$, размерность $n \leq 13$. В работе [14] представлена таблица направляющих точек для $1 \leq n \leq 51, 1 \leq s \leq 20$. К сожалению, достаточно простых формул для расчета направляющих точек при любом n и s нет.

Операция «*» означает поразрядное сложение по модулю 2 в двоичной системе:

$$\text{если } a = \sum_{k=1}^l a_k 2^{-k}; b = \sum_{k=1}^l b_k 2^{-k}, \text{ то } a * b = \sum_{k=1}^l c_k 2^{-k},$$

где $c_k = (a_k + b_k) \bmod 2$.

Более подробно: для того чтобы вычислить «сумму» $a * b$, надо оба эти числа записать в двоичной системе и сложить цифры в соответствующих разрядах по правилам: $0 + 0 = 0, 1 + 0 = 0 + 1 = 1, 1 + 1 = 0$, т. е. без переноса единиц в старшие разряды. Например: $\frac{7}{8} * \frac{11}{16} = 0,111 * 0,1011 =$

$$0,0101 = \frac{5}{16}.$$

Во всех ЭВМ имеется специальная команда для осуществления операции «*». Количество операций, затрачиваемых на ЭВМ для вычисления Q_i , растет с ростом i , но медленно, как $\log_2 i$, см. [11].

Например, вычислим координаты точки Q_{13} в 4-мерном кубе.

Так как в двоичной системе счисления число 13 записывается как 1101, то $e_1 = 1, e_2 = 0, e_3 = 1, e_4 = 1$ и координаты Q_{13} равны:

$$q_j^{13} = V_j^1 * V_j^3 * V_j^4, 1 \leq j \leq 4.$$

Рассчитываем векторы $\bar{V}^1, \bar{V}^3, \bar{V}^4$:

$$\bar{V}^1 = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right); \bar{V}^3 = \left(\frac{1}{8}, \frac{5}{8}, \frac{7}{8}, \frac{1}{8} \right); \bar{V}^4 = \left(\frac{1}{16}, \frac{15}{16}, \frac{11}{16}, \frac{5}{16} \right).$$

Записав их координаты в двоичной системе, получим

$$q_1^{13} = 0.1 * 0.001 * 0.0001 = 0.1011 = \frac{11}{16};$$

$$q_2^{13} = 0.1 * 0.101 * 0.1111 = 0.1101 = \frac{13}{16};$$

$$q_3^{13} = 0.1 * 0.111 * 0.1011 = 0.1101 = \frac{13}{16};$$

$$q_4^{13} = 0.1 * 0.001 * 0.0101 = 0.1111 = \frac{15}{16}.$$

Итак, $Q_{13} = (11/16, 13/16, 13/16, 15/16)$. Используя точки Q_i , можно получить соответствующие пробные точки $\bar{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$ в любом параллелепипеде Π :

$$a_j \leq x_j \leq b_j, \quad 1 \leq j \leq n.$$

Их координаты можно вычислить по формуле :

$$x_j^i = a_j + q_j^i (b_j - a_j), \quad 1 \leq j \leq n.$$

Доказано [14], что если точки $Q_i = (q_1^i, q_2^i, \dots, q_n^i)$ образуют равномерно распределенную последовательность в K^n , то точки $\bar{x}^i = (x_1^i, x_2^i, \dots, x_n^i)$ также образуют равномерно распределенную последовательность в n -мерном параллелепипеде Π . Кроме того доказано, что если $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^i, \dots$ — последовательность точек равномерно распределенная в заданном параллелепипеде Π со сторонами, параллельными координатным граням, а $G \subset \Pi$ — произвольная область с положительным объемом $V_G > 0$, то все отобранные точки $\bar{x}^i \in G$, также образуют равномерно распределенную в G последовательность точек. При этом отношение объемов $\gamma = V_G / V_\Pi$ называют эффективностью отбора, так как в среднем для получения одной точки в G приходится рассмотреть $1/\gamma$ точек в Π .

Естественно возникает вопрос: чем же эти “квазислучайные числа” лучше случайных в методе Монте-Карло [10, 12]?

- 1) На всех классах тестовых функций, на которых проводился численный эксперимент, получалась более быстрая сходимость и, по сравнению со случайным поиском, метод Холтона и ЛП-поиска позволял получить ту же точность при числе пробных точек в 2÷4 раза меньшем.
- 2) расчет точек последовательности Холтона и точек ЛП-последовательности весьма прост и легко программируется.

2. Тестовые задачи и оценка эффективности методов Холтона и ЛП-поиска

На практике оценку эффективности численных методов оптимизации, как правило, проводят с помощью вычислительных экспериментов при решении так называемых специальных тестовых задач [2, 3]. Одним из важнейших вопросов при этом является вопрос о широте применения метода, т. е. можно ли с помощью данного метода решить большинство возникающих на практике задач?

Работоспособность описанных выше методов проверена на следующих общепринятых тестовых функциях:

- 1) “Функция Розенброка”:

$$f(x_1, x_2) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$$

— унимодальная функция “овражного” типа с нелинейным параболического вида “дном” (имеет крутую закругленную впадину вдоль кривой $x_2 = x_1^2$). Минимум функции достигается в точке $\bar{x}^* = (1, 1)$ и $f(\bar{x}^*) = 0$.

2) “Функция Флетчера — Пауэлла”:

$$f(x_1, x_2, x_3) = 100 \cdot \left\{ [x_3 - 5 \cdot \theta(x_1, x_2)]^2 + \left[(x_1^2 + x_2^2)^{1/2} - 1 \right]^2 \right\} + x_3^2,$$

где

$$\theta(x_1, x_2) = \begin{cases} \frac{1}{\pi} \arctg\left(\frac{x_2}{x_1}\right), & \text{если } x_1 > 0, \\ 1 + \frac{1}{\pi} \arctg\left(\frac{x_2}{x_1}\right), & \text{если } x_1 < 0, \end{cases}$$

— функция с «извивающимся оврагом». Ее частные производные 1-го порядка кусочно-непрерывные. Глобальный минимум достигается в точке $\bar{x}^* = (1, 0, 0)$ и $f(\bar{x}^*) = 0$.

3) “Функция Пауэлла”:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5 \cdot (x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10 \cdot (x_1 - x_4)^4$$

имеет «уплощенное дно оврага» (слабовырожденная ситуация). Минимум достигается в точке $\bar{x}^* = (0, 0, 0, 0)$ и $f(\bar{x}^*) = 0$. Гессиан функции вырожден в точке \bar{x}^* .

4) “Функция Вуда”:

$$f(x_1, x_2, x_3, x_4) = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2 + 90 \cdot (x_4 - x_3^2)^2 + (1 - x_3)^2 + 10,1 \cdot ((x_2 - 1)^2 + (x_4 - 1)^2) + 19,8 \cdot (x_2 - 1)(x_4 - 1)$$

имеет несколько локальных минимумов, отличных от глобального. Глобальный минимум достигается в точке $\bar{x}^* = (1, 1, 1, 1)$ и $f(\bar{x}^*) = 0$.

5) “Частный случай функции Растригина” для 2-х переменных:

$$f(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

на $[-\pi, \pi]$ функция имеет 25 локальных минимумов. Глобальный минимум достигается в точке $\bar{x}^* = (0, 0)$ и $f(\bar{x}^*) = -2$.

6) “Функция Растригина” для 3-х переменных ($n = 3$):

$$f(\bar{x}) = n + \sum_{i=1}^n x_i^2 - \cos(2 \cdot \pi \cdot x_i);$$

Особенностью функции является существование большого числа локальных минимумов. Глобальный минимум достигается в точке $\bar{x}^* = (0, 0, 0)$; $f(\bar{x}^*) = 0$.

7) “Функция Гриеванга” для 4-х переменных ($n = 4$):

$$f(\bar{x}) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right);$$

Глобальный минимум достигается в точке $\bar{x}^* = (0, 0, 0, 0)$; $f(\bar{x}^*) = 0$.

8) “Функция Швевеля” для 3-х переменных ($n = 3$):

$$f(\bar{x}) = 418.9828872724338 \cdot n - \sum_{i=1}^n x_i \cdot \sin(\sqrt{|x_i|});$$

Особенностью функции является большое расстояние между глобальным минимумом и «наилучшим» из локальных минимумов с большим числом «худших» локальных минимумов между ними. Глобальный минимум достигается в точке $\bar{x}^* = (420,9687; 420,9687; 420,9687)$; $f(\bar{x}^*) = 0$.

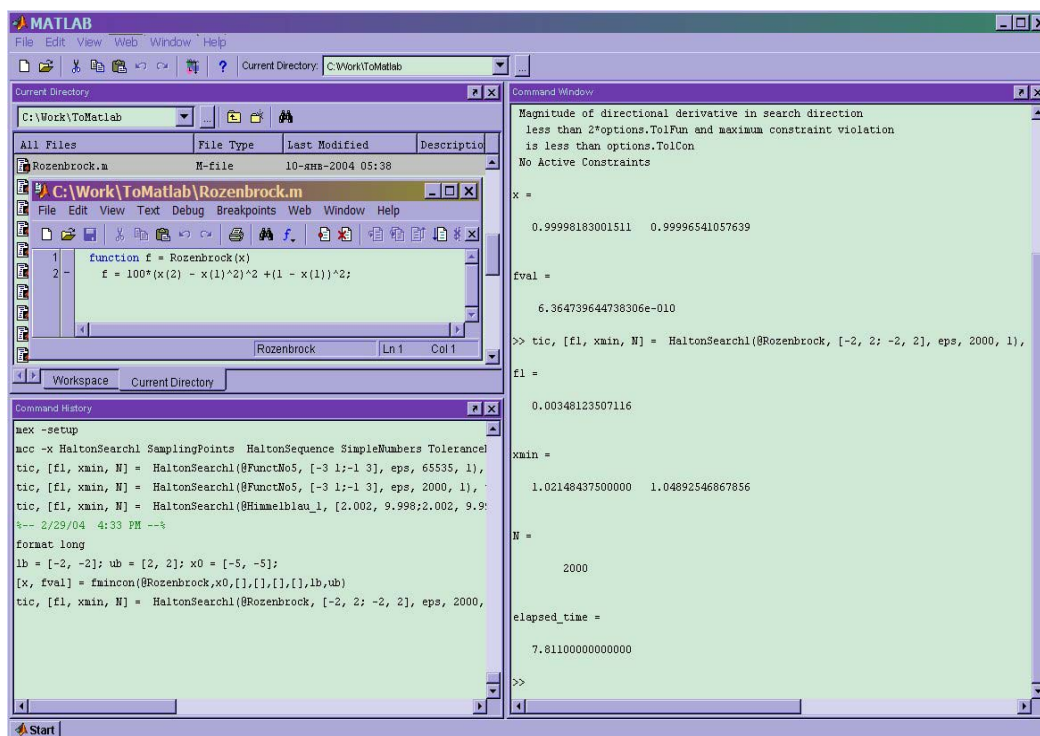


Рис. 3. Пример визуализации результатов тестовых расчетов в среде MATLAB 6.5.

Любая серьезная оптимизационная процедура должна эффективно решать указанные задачи.

Выбор как мультимодальных, так и унимодальных тестовых функций неслучаен. Необходимо убедиться в работоспособности рассматриваемых методов глобальной оптимизации не только для поиска глобальных минимумов мультимодальных функций, но и для поиска минимумов унимодальных функций сложной геометрии, на которых тестируются все известные градиентные методы.

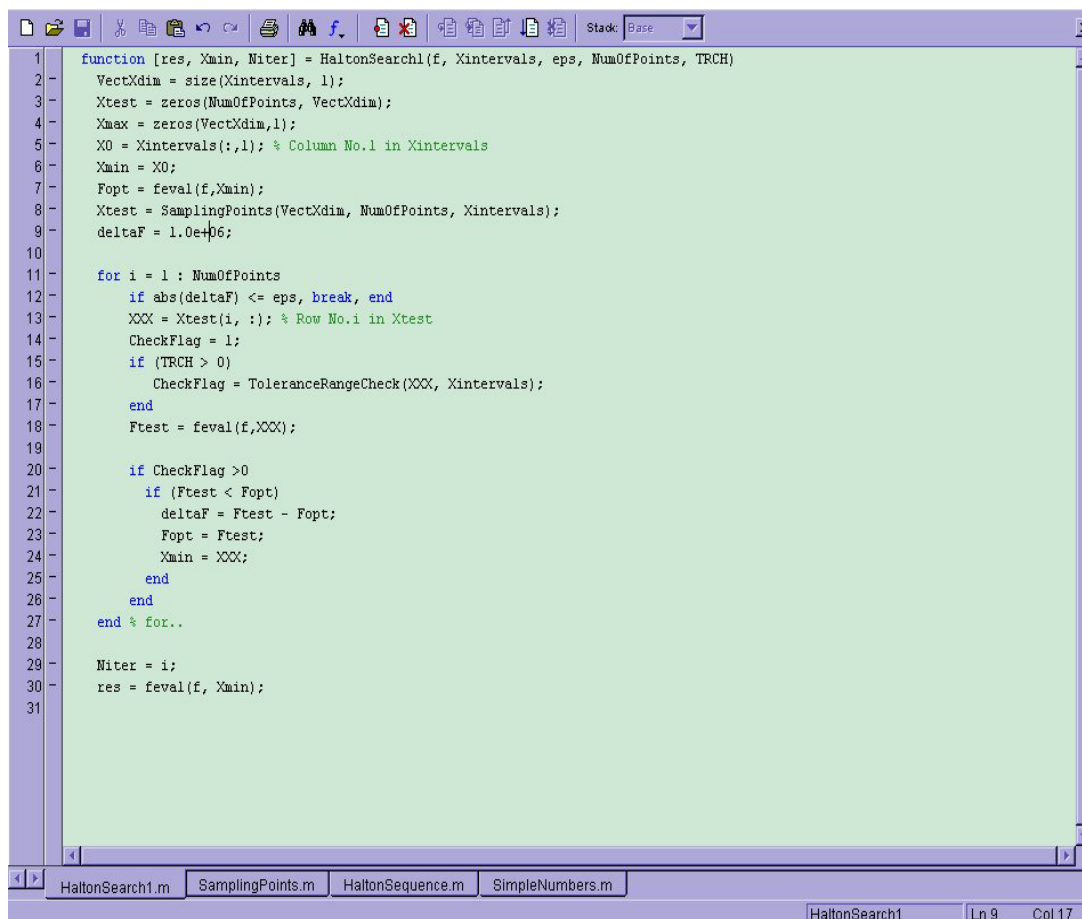
Все рассмотренные в данной работе алгоритмы глобального поиска были реализованы в виде программ в среде MATLAB 6.5, скомпилированы в соответствующие MEX-файлы и библиотеки dll, которые могут быть в дальнейшем успешно использованы при написании прикладных программ на языках программирования высокого уровня, в частности на C⁺⁺.

Ниже представлены результаты расчетов с использованием стандартных оптимизационных процедур MATLAB 6.5 (**fmin**, **fmincon**) и

МEX-файлов, реализующих методы Холтона и ЛП-поиска и предусматривающих возможность уточнения решения, т. е. нахождение точки глобального минимума с любой предварительно заданной точностью. В качестве метода уточнения использовался один из квазиньютоновских методов — метод Марквардта с наиболее распространенным критерием останова для гладких задач вида $|\text{grad } f(\bar{x}^k)| < \varepsilon$. В табл. 1–5 представлены результаты расчетов для соответствующих тестовых функций.

Пример нахождения минимума «функции Розенброка» с помощью стандартной процедуры MATLAB 6.5 — **fmincon** и метода Холтона без уточнения с заданным количеством пробных точек $N=2000$ в среде MATLAB 6.5 показан на рис. 3.

Фрагмент программы, реализующей метод Холтона без уточнения, показан на рис. 4.



```

1 function [res, Xmin, Niter] = HaltonSearch1(f, Xintervals, eps, NumOfPoints, TRCH)
2     VectXdim = size(Xintervals, 1);
3     Xtest = zeros(NumOfPoints, VectXdim);
4     Xmax = zeros(VectXdim, 1);
5     X0 = Xintervals(:, 1); % Column No.1 in Xintervals
6     Xmin = X0;
7     Fopt = feval(f, Xmin);
8     Xtest = SamplingPoints(VectXdim, NumOfPoints, Xintervals);
9     deltaF = 1.0e+06;
10
11     for i = 1 : NumOfPoints
12         if abs(deltaF) <= eps, break, end
13         XXX = Xtest(i, :); % Row No.i in Xtest
14         CheckFlag = 1;
15         if (TRCH > 0)
16             CheckFlag = ToleranceRangeCheck(XXX, Xintervals);
17         end
18         Ftest = feval(f, XXX);
19
20         if CheckFlag > 0
21             if (Ftest < Fopt)
22                 deltaF = Ftest - Fopt;
23                 Fopt = Ftest;
24                 Xmin = XXX;
25             end
26         end
27     end % for..
28
29     Niter = i;
30     res = feval(f, Xmin);
31

```

Рис. 4. Фрагмент программы в виде m-файла, реализующей метод Холтона без уточнения в среде MATLAB 6.5.

1) «Функция Розенброка» ($-2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2$):

Таблица 1.

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП- поиска с уточне- нием ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
2000	$f(\bar{x}^*)=0,6260 \cdot 10^{-2}$ $\bar{x}^* = (1,0078;$ $1,0078)$	$f(\bar{x}^*)=0,3845 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ $1,0000)$	$f(\bar{x}^*)=0,3481 \cdot 10^{-2}$ $\bar{x}^* = (1,0214;$ $1,0489)$	$f(\bar{x}^*)=0,3845 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ $1,0000)$
8192	$f(\bar{x}^*)=0,4364 \cdot 10^{-2}$ $\bar{x}^* = (0,9638;$ $0,9345)$	$f(\bar{x}^*)=0,1147 \cdot 10^{-13}$ $\bar{x}^* = (0,9999;$ $0,9999)$	$f(\bar{x}^*)=0,4207 \cdot 10^{-3}$ $\bar{x}^* = (0,9980;$ $0,9940)$	$f(\bar{x}^*)=0,1147 \cdot 10^{-13}$ $\bar{x}^* = (0,9999;$ $0,9999)$
32767	$f(\bar{x}^*)=0,1216 \cdot 10^{-2}$ $\bar{x}^* = (0,9713;$ $0,9454)$	$f(\bar{x}^*)=0,1147 \cdot 10^{-13}$ $\bar{x}^* = (0,9999;$ $0,9999)$	$f(\bar{x}^*)=0,4207 \cdot 10^{-3}$ $\bar{x}^* = (0,9980;$ $0,9940)$	$f(\bar{x}^*)=0,1147 \cdot 10^{-13}$ $\bar{x}^* = (0,9999;$ $0,9999)$
65535	$f(\bar{x}^*)=0,3650 \cdot 10^{-3}$ $\bar{x}^* = (1,0169;$ $1,0332)$	$f(\bar{x}^*)=0,1835 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ $1,000)$	$f(\bar{x}^*)=0,3650 \cdot 10^{-3}$ $\bar{x}^* = (1,0169;$ $1,0332)$	$f(\bar{x}^*)=0,1835 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ $1,000)$

Решение данной задачи минимизации с помощью системы MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0 = (-5, -5)$: $\bar{x}^* = (0,9999; 0,9999)$; $f(\bar{x}^*) = 6,3647 \cdot 10^{-10}$;

б) при $\bar{x}^0 = (5; 5)$: $\bar{x}^* = (1,0000; 1,0000)$; $f(\bar{x}^*) = 5,8160 \cdot 10^{-10}$;

в) при $\bar{x}^0 = (-5; 5)$: $\bar{x}^* = (0,9998; 0,9997)$; $f(\bar{x}^*) = 1,6852 \cdot 10^{-8}$.

Результат удовлетворительный.

2) «Функция Флетчера — Пауэлла» ($-1 \leq x_1 \leq 1, 0 \leq x_2 \leq 2, 0 \leq x_3 \leq 2$):

Таблица 2.

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП- поиска с уточне- нием ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
2000	$f(\bar{x}^*)=0,2782$ $\bar{x}^* = (0,9521;$ $0,0410;$ $0,0449)$	$f(\bar{x}^*)=0,7612 \cdot 10^{-16}$ $\bar{x}^* = (1,0000;$ $0,5495 \cdot 10^{-8};$ $0,8721 \cdot 10^{-8})$	$f(\bar{x}^*)=0,5483;$ $\bar{x}^* = (0,9450;$ $0,3063;$ $0,4397)$	$f(\bar{x}^*)=0,7612 \cdot 10^{-16}$ $\bar{x}^* = (1,0000;$ $0,5495 \cdot 10^{-8};$ $0,8721 \cdot 10^{-8})$
8192	$f(\bar{x}^*)=0,1721$ $\bar{x}^* = (0,9700;$ $0,2556;$ $0,3991)$	$f(\bar{x}^*)=0,1881 \cdot 10^{-15}$ $\bar{x}^* = (1,0;$ $0,8638 \cdot 10^{-8};$ $0,1371 \cdot 10^{-7})$	$f(\bar{x}^*)=0,4029 \cdot$ $\bar{x}^* = (0,9480;$ $0,3714;$ $0,6025)$	$f(\bar{x}^*)=0,1881 \cdot 10^{-15}$ $\bar{x}^* = (1,0000;$ $0,8638 \cdot 10^{-8};$ $0,1371 \cdot 10^{-7})$

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП- поиска с уточне- нием ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
32767	$f(\bar{x}^*)=0,0970$ $\bar{x}^*=(0,9846;$ 0,1121; 0,1549)	$f(\bar{x}^*)=0,1703 \cdot 10^{-14}$ $\bar{x}^*=(1,0000;$ $0,2599 \cdot 10^{-7};$ $0,4125 \cdot 10^{-7})$	$f(\bar{x}^*)=0,090;$ $\bar{x}^*=(0,9821;$ 0,012; 0,0434)	$f(\bar{x}^*)=0,1703 \cdot 10^{-14}$ $\bar{x}^*=(1,0000;$ $0,2599 \cdot 10^{-7};$ $0,4125 \cdot 10^{-7})$
65535	$f(\bar{x}^*)=0,0867$ $\bar{x}^*=(0,9714;$ 0,0282; 0,0528)	$f(\bar{x}^*)=0,1703 \cdot 10^{-14}$ $\bar{x}^*=(1,0000;$ $0,2599 \cdot 10^{-7};$ $0,4125 \cdot 10^{-7})$	$f(\bar{x}^*)=0,0751;$ $\bar{x}^*=(0,9747;$ 0,0985; 0,1496)	$f(\bar{x}^*)=0,1703 \cdot 10^{-14}$ $\bar{x}^*=(1,0000;$ $0,2599 \cdot 10^{-7};$ $0,4125 \cdot 10^{-7})$

Решение данной задачи минимизации в системе MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0=(-10; -10; -10)$: $\bar{x}^*=(0,9999; 0,3 \cdot 10^{-5}; 0,0000)$; $f(\bar{x}^*)=2,8684 \cdot 10^{-8}$;

б) при $\bar{x}^0=(-0,001; 10; 10)$: $\bar{x}^*=(1,0000; 0,43 \cdot 10^{-4}; 0,54 \cdot 10^{-4})$;
 $f(\bar{x}^*)=2,3244 \cdot 10^{-8}$.

Результат удовлетворительный.

3) «Функция Пауэлла» ($-1 \leq x_i \leq 2, i = 1, 2, 3, 4$):

Таблица 3.

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП-поиска с уточнением ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
2000	$f(\bar{x}^*)=0,6550$ $\bar{x}^*=(0,7890;$ 0,0547; 0,3672; 0,4141)	$f(\bar{x}^*)=0,2925 \cdot 10^{-9}$ $\bar{x}^*=(0,0034;$ $-0,00034;$ 0,0017; 0,0017)	$f(\bar{x}^*)=0,7399;$ $\bar{x}^*=(0,3549;$ 0,0397; 0,0573; 0,0021)	$f(\bar{x}^*)=0,2925 \cdot 10^{-9}$ $\bar{x}^*=(0,0034;$ $-0,00034;$ 0,0017; 0,0017)
8192	$f(\bar{x}^*)=0,1618$ $\bar{x}^*=(0,0518;$ 0,036; 0,1236; 0,013)	$f(\bar{x}^*)=0,7944 \cdot 10^{-9}$ $\bar{x}^*=(-0,0044;$ 0,0044; $-0,0022;$ $-0,0022)$	$f(\bar{x}^*)=0,0711$ $\bar{x}^*=(-0,1427;$ 0,0269; $-0,0273;$ 0,0599)	$f(\bar{x}^*)=0,7944 \cdot 10^{-9}$ $\bar{x}^*=(-0,0044;$ 0,0044; $-0,0022;$ $-0,0022)$
32767	$f(\bar{x}^*)=0,1618$ $\bar{x}^*=(0,0518;$ 0,036; 0,1236; 0,013)	$f(\bar{x}^*)=0,7944 \cdot 10^{-9}$ $\bar{x}^*=(-0,0044;$ 0,0044; $-0,0022;$ $-0,0022)$	$f(\bar{x}^*)=0,0711$ $\bar{x}^*=(-0,1427;$ 0,0269; $-0,0273;$ 0,0599)	$f(\bar{x}^*)=0,7944 \cdot 10^{-9}$ $\bar{x}^*=(-0,0044;$ 0,0044; $-0,0022;$ $-0,0022)$
65535	$f(\bar{x}^*)=0,0625$	$f(\bar{x}^*)=0,3166 \cdot 10^{-9}$	$f(\bar{x}^*)=0,0336;$	$f(\bar{x}^*)=0,3166 \cdot 10^{-9}$

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП-поиска с уточнением ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
	$\bar{x}^* = (0,2141;$ 0,0077; 0,0577; 0,1507)	$\bar{x}^* = (0,00003;$ -0,00003; 0,00189; 0,00189)	$\bar{x}^* = (-0,089;$ -0,0031; 0,0338; 0,0803)	$\bar{x}^* = (0,00003;$ -0,00003; 0,00189; 0,00189)

Решение данной задачи минимизации в системе MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0 = (-5, -5, -5, -5)$: $\bar{x}^* = (-0,0054; 0,0054; -0,0093; 0,0093)$;

$$f(\bar{x}^*) = 1.4248 \cdot 10^{-7};$$

б) при $\bar{x}^0 = (-5, 5, -5, 5)$: $\bar{x}^* = (-0.0236; 0,0235; 0.0023; -0.0113)$;

$$f(\bar{x}^*) = 6.3272 \cdot 10^{-7}.$$

Результат удовлетворительный, однако, точность решения зависит от начальных условий.

4) «Функция Вуда» ($0 \leq x_i \leq 3, i = 1, 2, 3, 4$):

Таблица 4.

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП-поиска с уточнением ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
2000	$f(\bar{x}^*) = 2,6487$ $\bar{x}^* = (0,8862;$ 0,7163; 1,3051; 1,5981)	$f(\bar{x}^*) = 0,2800 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ 1,0000; 0,9999; 0,9999)	$f(\bar{x}^*) = 3,3474$ $\bar{x}^* = (1,1176;$ 1,3525; 0,8851; 0,6272)	$f(\bar{x}^*) = 0,2800 \cdot 10^{-12}$ $\bar{x}^* = (1,0000;$ 1,0000; 0,9999; 0,9999)
8192	$f(\bar{x}^*) = 1,7646$ $\bar{x}^* = (0,0933;$ 0,0186; 1,4117; 1,9288)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)	$f(\bar{x}^*) = 0,8960$ $\bar{x}^* = (0,8909;$ 0,7892; 1,0955; 1,1085)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)
32767	$f(\bar{x}^*) = 1,4175$ $\bar{x}^* = (1,1044;$ 1,1337; 0,9775; 1,0236)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)	$f(\bar{x}^*) = 0,8960$ $\bar{x}^* = (0,8909;$ 0,7892; 1,0955; 1,1085)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)
65535	$f(\bar{x}^*) = 0,1699$ $\bar{x}^* = (0,7767;$ 0,6017; 1,1534; 1,3391)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)	$f(\bar{x}^*) = 0,8959$ $\bar{x}^* = (0,8909;$ 0,7892; 1,0955; 1,1085)	$f(\bar{x}^*) = 0,3620 \cdot 10^{-13}$ $\bar{x}^* = (1,0000;$ 1,0000; 1,0000; 1,0000)

Решение данной задачи минимизации в системе MATLAB 6.5 привело к следующим результатам:

- а) при $\bar{x}^0 = (-5; 4; -3; 2)$: $\bar{x}^* = (1,0000; 1,0001; 0,9999; 0,9998)$;
 $f(\bar{x}^*) = 1,4134 \cdot 10^{-8}$;
- б) при $\bar{x}^0 = (-2; -2; -3; -3)$: $\bar{x}^* = (0,9998; 0,9997; 1,0001; 1,0002)$;
 $f(\bar{x}^*) = 1,2373 \cdot 10^{-7}$;
- в) при $\bar{x}^0 = (0; 0; 0; 0)$: $\bar{x}^* = (0,9998; 0,9997; 1,0001; 1,0002)$;
 $f(\bar{x}^*) = 1,4509 \cdot 10^{-7}$.

Результат удовлетворительный, однако, точность решения зависит от начальных условий.

5) Частный случай «функции Растригина» ($-3 \leq x_1 \leq 1, -1 \leq x_2 \leq 3$):

Таблица 5.

Кол- во то- чек N	Метод ЛП- поиска	Метод ЛП-поиска с уточнением ($\varepsilon=10^{-6}$)	Метод Холтона	Метод Холтона с уточнением ($\varepsilon=10^{-6}$)
2000	$f(\bar{x}^*) = -2,000$ $\bar{x}^* = (0,0000;$ $0,0000)$	$f(\bar{x}^*) = -2,0000$ $\bar{x}^* = (0,0000;$ $0,0000)$	$f(\bar{x}^*) = -1,8175$ $\bar{x}^* = (-0,3633;$ $-0,0105)$	$f(\bar{x}^*) = -1,8175$ $\bar{x}^* = (-0,3633;$ $-0,0105)$
8192	$f(\bar{x}^*) = -1,861$ $\bar{x}^* = (0,0087;$ $-0,3409)$	$f(\bar{x}^*) = -1,8605$ $\bar{x}^* = (0,0087;$ $-0,3409)$	$f(\bar{x}^*) = -1,8605$ $\bar{x}^* = (0,0087;$ $-0,3409)$	$f(\bar{x}^*) = -1,8605$ $\bar{x}^* = (0,0087;$ $-0,3409)$
32767	$f(\bar{x}^*) = -2,000$ $\bar{x}^* = (0,0000;$ $0,0000)$	$f(\bar{x}^*) = -2,0000$ $\bar{x}^* = (0,0000;$ $0,0000)$	$f(\bar{x}^*) = -1,9634$ $\bar{x}^* = (-0,0128;$ $-0,0078)$	$f(\bar{x}^*) = -1,9634$ $\bar{x}^* = (-0,0128;$ $-0,0078)$
65535	$f(\bar{x}^*) = -2,000$ $\bar{x}^* = (0,0000$ $0,0000)$	$f(\bar{x}^*) = -2,0000$ $\bar{x}^* = (0,0000$ $0,0000)$	$f(\bar{x}^*) = -1,9964$ $\bar{x}^* = (0,0046$ $0,0009)$	$f(\bar{x}^*) = -1,9964$ $\bar{x}^* = (-0,0046$ $-0,0009)$

Решение данной задачи минимизации в системе MATLAB 6.5 привело к следующим результатам:

- а) при $\bar{x}^0 = (-3; -3)$: $\bar{x}^* = (-3,0; -1,0)$; $f(\bar{x}^*) = 10,1689$;
- б) при $\bar{x}^0 = (-1; -1)$: $\bar{x}^* = (-1,0408; -1,0000)$; $f(\bar{x}^*) = 0,4296$;
- в) при $\bar{x}^0 = (0,5; -0,5)$: $\bar{x}^* = (-0,3469; 0,3469)$; $f(\bar{x}^*) = -1,7578$;
- г) при $\bar{x}^0 = (0,2; 0,2)$: $\bar{x}^* = (0,3469; 0,3469)$; $f(\bar{x}^*) = -1,7578$;
- д) при $\bar{x}^0 = (0,05; 0,05)$: $\bar{x}^* = (-0,0099; 0,3652)$; $f(\bar{x}^*) = -1,9999$;

Из табл. 5 заключаем, что в системе MATLAB 6.5 задача не решается, в виду зависимости результата решения от принятых начальных условий поиска (система отыскивает локальные минимумы функции, в окрестности которых находится заданная начальная точка).

6) Решение задачи минимизации “функции Растригина” при $n = 3$, $(-2 \leq x_1 \leq 3; -2 \leq x_2 \leq 2; -3 \leq x_3 \leq 1)$ методом Холтона с уточнением ($\varepsilon = 10^{-6}$) с числом пробных точек $N = 2000$ дало следующие результаты:

$$\bar{x}^* = (0,3865 \cdot 10^{-6}; -0,1153 \cdot 10^{-6}; 0,5959 \cdot 10^{-6}); f(\bar{x}^*) = 0,0.$$

Результат удовлетворительный.

Решение задачи минимизации функции 6) в среде MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0 = (3, 2, 1)$: $\bar{x}^* = (0,9510; 0,9511; 0,0001)$; $f(\bar{x}^*) = 1,9028$;

б) при $\bar{x}^0 = (-2, -3, -2)$: $\bar{x}^* = (0,9511; 0,9510; 0,9510)$; $f(\bar{x}^*) = 2,8543$.

Согласно полученным результатам, можно заключить, что в среде MATLAB 6.5 задача не решается при любых начальных условиях, тогда как методы глобального поиска, описанные выше, дают приемлемые результаты даже при небольшом количестве пробных точек.

7) Минимизация “функции Гриеванга” ($-10 \leq x_i \leq 10, i = \overline{1,4}$) методом Холтона без уточнения с числом пробных точек $N = 8192$ дала следующие результаты:

$$\bar{x}^* = (-0,0073; 0,1753; 0,5549; 0,2350); f(\bar{x}^*) = 0,0648.$$

Результат удовлетворительный.

Решение задачи минимизации функции № 7 в среде MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0 = (-10, -10, -10, -10)$: $\bar{x}^* = (-9,4115; -8,8522; -5,4135; 10,0000)$; $f(\bar{x}^*) = 0,7905$;

б) при $\bar{x}^0 = (5,0; 6,0; 5,0; 6,0)$: $\bar{x}^* = (3,1400; 4,4394; 5,4335; 6,2703)$; $f(\bar{x}^*) = 0,0246$.

Как видно из результатов тестовых расчетов, решение поставленной задачи в MATLAB 6.5 не удовлетворительное при любых начальных условиях. Метод Холтона успешно справляется с данной задачей и при сравнительно небольшом количестве пробных точек.

8) Минимизация “Функции Швевеля” ($-500 \leq x_i \leq 500, i = \overline{1,3}$) методом Холтона без уточнения с числом пробных точек $N = 8192$ дала следующие результаты:

$$\bar{x}^* = (420,9043; 421,1550; 420,7750); f(\bar{x}^*) = 0,0096.$$

Результат удовлетворительный.

Решение задачи минимизации функции 8) в MATLAB 6.5 привело к следующим результатам:

а) при $\bar{x}^0 = (-50; -50; -50)$: $\bar{x}^* = (420,9687; 420,9687; 420,9688)$; $f(\bar{x}^*) = 0,0600$ — результат близок к точному решению;

б) при $\bar{x}^0 = (-450; -450; -450)$: $\bar{x}^* = (-450; -450; -450)$; $f(\bar{x}^*) = 2204,40$.

Как видно из вышеприведенных результатов, в MATLAB 6.5 задача не решается, в виду зависимости результата решения от принятых начальных условий поиска (система отыскивает локальные минимумы функции, в окрестности которых находится заданная начальная точка). Метод Холтона успешно справляется с данной задачей и при сравнительно небольшом количестве пробных точек.

Результаты расчетов показывают исключительно высокую эффективность методов Холтона и ЛП-поиска для решения задачи численной глобальной оптимизации функции многих переменных.

К достоинству методов следует отнести также тот факт, что для их применения не требуется не только дифференцируемости целевой функции, но даже ее аналитического задания. Достаточно лишь иметь возможность вычислять значения функции в произвольных точках ее области определения, что обеспечивает широту применения методов и возможность решения большинства практических задач.

Вместе с тем не следует недооценивать достоинства градиентных методов, главным из которых является высокая скорость сходимости к точке минимума. Представляется целесообразным использовать лучшие качества прямых и градиентных методов и использовать их в комбинации, что может существенно уменьшить объем необходимых вычислений для поиска глобального минимума с заданной точностью. Этот вывод подтверждается и результатами проведенного численного эксперимента, представленными выше.

Заключение

В заключение рассмотрим еще одну проблему численной оптимизации — решение задач большой размерности. Известно [4], что в случае, когда размерность n аргумента целевой функции становится очень большой, применение градиентных процедур становится невозможным или вызывает значительные технические трудности.

Если размерность целевой функции $n > 51$, то можно воспользоваться обобщенной ЛП_τ-последовательностью, вычисляемой по тем же формулам для случая $1 \leq j < \infty$. К сожалению, достаточно простых формул для расчета направляющих точек, как это уже было отмечено выше, нет. Поэтому можно воспользоваться уже рассчитанной таблицей чисел V_j^S для $n \leq 51$, а в качестве недостающих координат квазислучайных точек можно выбирать обычные псевдослучайные числа γ , например,

$$Q_i = (q_1^i, q_2^i, \dots, q_{51}^i, \gamma_{52}^i, \gamma_{53}^i, \dots, \gamma_m^i).$$

Так как различные координаты точек Q_i некоррелированы [11] (координаты с меньшими номерами распределены лучше), то переменные целевой функции целесообразно нумеровать так, чтобы наиболее существенные координаты

наты имели меньшие номера, для которых вычисляем q_j^i , а по всем остальным координатам — γ_j^i . Такой способ расчета может ускорить сходимость, по сравнению с расчетом только по случайным точкам γ .

Второй подход к решению задачи глобальной оптимизации в случае большой размерности целевой функции состоит в использовании обобщенной последовательности Холтона $\{P_i\}$, $i = 1, 2, \dots$, точки которой имеют координаты $P_i = (p_{r_1}(i), p_{r_2}(i), \dots, p_{r_n}(i), \dots)$, где $r_1 < r_2 < r_3 < \dots < r_n < \dots$ — последовательность всех простых чисел. Для расчета точек P_i на практике можно задать достаточно большую таблицу простых чисел или запрограммировать какой-нибудь алгоритм их нахождения, например, метод решета (метод Эратосфена). В настоящее время имеются напечатанные таблицы простых чисел до 12 миллионов, т. е. можно получить последовательности точек Холтона практически любой размерности.

Литература

1. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике. Т.1, 2.— М.: Мир, 1986.
2. Химмельблау Д. Прикладное нелинейное программирование.— М.: Мир, 1975.
3. Банди Б. Методы оптимизации (вводный курс).— М.: Радио и связь, 1988.
4. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация.— М.: Мир, 1985.
5. Уайлд Д. Дж. Методы поиска экстремума.— М.: Наука. 1967.
6. Шуп Т. Решение инженерных задач на ЭВМ.— М.: Мир, 1982.
7. Полак Э. Численные методы оптимизации.— М.: Мир, 1974.
8. Растринин Л. А. Статистические методы поиска.— М.: Наука, 1968.
9. Растринин Л. А. Системы экстремального управления.— М.: Физматгиз, 1974.
10. Соболев И. М. Многомерные интегралы и метод Монте-Карло // ДАН.— Т. 114.— 1957.
11. Соболев И. М. Многомерные квадратурные формулы и функции Хаара.— М.: Наука, 1969.
12. Соболев И. М. Численные методы Монте-Карло.— М.: Наука. 1973.
13. Соболев И. М., Статников Р. Б. ЛП-поиск и задачи оптимального конструирования. Проблемы случайного поиска.— Рига, 1972.
14. Соболев И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями.— М.: Наука, 1981.
15. Halton J. H. A retrospective and prospective survey of the Monte Carlo method. // SIAM Rev.— 1970.— V.12.— N.1.— C.1–63.

16. *Halton J. H., Handscomb D. C.* A method for increasing the efficiency of Monte Carlo integrations. // J. Assoc. Comput. Machinery.— 1957.— V.4.— N.3.— С.329–340.
17. *Карманов В. Г.* Математическое программирование.— М.:Наука, 1975.
18. *Васильев Ф. П.* Численные методы решения экстремальных задач.— М.: Наука, 1980.
19. *Сухарев А. Г., Тимохов А. В., Федоров В. В.* Курс методов оптимизации.— М.:Наука, 1986.
20. *Моисеев Н. Н., Иванюков Ю. П., Столярова Е. М.* Методы оптимизации.— М.:Наука, 1978.
21. *Медынский М. М.* Численные методы оптимизации в задачах проектирования систем оснащения ЛА.— М.: Изд-во МАИ, 1993.
22. *Медынский М. М., Антоний Е. В.* Численные методы нелинейной оптимизации: алгоритмы и программы.— М.: Изд-во МАИ, 2003.

УДК 519

ТЕОРЕТИКО–ИГРОВАЯ МОДЕЛЬ ВХОЖДЕНИЯ В РЫНОК КУРНО N ФИРМ

Ершова Т. А., Малафеев О. А.

Санкт-Петербургский государственный университет, Санкт-Петербург,
e-mail: tatyana@TE9758.spb.edu, e-mail: oam@apmath.spbu.ru

Рассматривается обобщение двухпериодной модели Диксита вхождения в рынок двух фирм. В модели Диксита функции предельных затрат предполагались постоянными, а функция ценообразования - линейной от количества товара, поставляемого на рынок двумя фирмами.

Данная модель обобщается на случай нескольких периодов и нескольких игроков, кроме того, функции предельных затрат предполагаются функциями количества товара, они убывают и выпуклы вниз, а функция ценообразования считается нелинейной. Такое обобщение более точно описывает экономические процессы, оно учитывает возрастающую отдачу от производства и нелинейность ценообразования.

Двухпериодная модель вхождения в рынок Курно n фирм

В данном разделе рассматривается обобщение модели Диксита на случай нескольких фирм, функции предельных затрат предполагаются функциями количества товара, они убывают и выпуклы вниз, а функция ценообразования считается нелинейной.

Эта модель является двухпериодной. В первом периоде некоторые фирмы выбирают объемы производства, а во втором - все фирмы одновременно выбирают количества товара, поставляемого на рынок. Фирмы выпускают одинаковый товар, неразличимый с точки зрения покупателя. В результате совместного выбора на рынке устанавливается цена. Далее можно рассчитать функции прибыли фирм во втором периоде.

Первый период. В первом периоде m фирм, $m \leq n$, выбирают объемы производства — K_1, \dots, K_m , неся при этом долгосрочные затраты (затраты на вход) $c_1^0(K_1), \dots, c_m^0(K_m)$, где c_1^0, \dots, c_m^0 — убывающие, выпуклые вниз функции:

$$\begin{cases} (c_1^0)' < 0, \dots, (c_m^0)' < 0, \\ (c_1^0)'' \geq 0, \dots, (c_m^0)'' \geq 0. \end{cases}$$

Второй период. Все фирмы одновременно выбирают количества товара, q_1, \dots, q_n , поставляемого на рынок.

Кроме того, оставшиеся $n - m$ фирм выбирают объемы производства K_{n-m}, \dots, K_n . Объемы производства берутся такими, что $K_{n-m} = q_{n-m}, \dots, K_n = q_n$ (в противном случае, когда $K_{n-m} \leq q_{n-m}, \dots, K_n \leq q_n$, фирмы несут некупаемые долгосрочные затраты). Долгосрочные затраты на производство $n - m$ фирм суть соответственно $c_{n-m}^0(q_{n-m}), \dots, c_n^0(q_n)$.

Предельные затраты на производство для n фирм обозначим через $c_1(q), \dots, c_n(q)$. Будем полагать выполненным неравенство $c_i(q) < c_j(q)$, $i = 1, \dots, m; j = n - m, \dots, n$, отражающее преимущество первых фирм за счет более раннего освоения производственных мощностей. Кроме того, полагаем $c'_1(q) < 0, \dots, c'_n(q) < 0, c''_1(q) \geq 0, \dots, c''_n(q) \geq 0$.

Предельные затраты фирм, выбирающих объемы производства во втором периоде, складываются из краткосрочных и долгосрочных затрат: $c_j^0(K_j) + c_j(q_j)$, $j = n - m, \dots, n$. Для первых фирм справедливо следующее: если $q_i < K_i, i = 1, \dots, m$, то фирма i несет краткосрочные предельные затраты, равные $c_i(q)$, каждая же единица продукции сверх K_i сопряжена с долгосрочными предельными затратами в количестве $c_i^0(q_i) + c_i(q_i)$.

В зависимости от количества товара на рынке складывается цена $p = p(\sum_{i=1}^n q_i)$ — убывающая, выпуклая вниз функция $p' < 0, \dots, p'' > 0$.

Прибыль во втором периоде будет находиться по формулам:

$$D_i = \begin{cases} q_i p - c_i K_i, & q_i \leq K_i \\ q_i p - (c_i + c_i^0) q_i, & q_i > K_i \end{cases} \quad i = 1, \dots, m,$$

$$D_i = q_i p - (c_i + c_i^0) q_i, \quad i = n - m, \dots, n.$$

Многопериодная модель вхождения в рынок Курно n фирм.

В данном разделе рассматривается обобщение модели Диксита на случай нескольких фирм и нескольких периодов, функции предельных затрат предполагаются функциями количества товара, они убывают и выпуклы вниз, а функция ценообразования считается нелинейной.

Эта модель является многопериодной. Фирма выбирает объем производства в одном из периодов, после своего выбора она не может изменить объем производства. В последнем периоде все фирмы одновременно выбирают количества товара, поставляемого на рынок. Фирмы выпускают одинаковый товар, неразличимый с точки зрения покупателя. В результате совместного выбора на рынке устанавливается цена. Далее можно рассчитать функции прибыли фирм в последнем периоде.

Пусть всего m периодов, n_1, \dots, n_m — количество фирм, выбравших объем производства соответственно в $1, \dots, m$ периоде, $\sum_{i=1}^m n_i = n$.

Первый период. В первом периоде n_1 фирм, $n_1 \leq n$, выбирают объемы производства — K_1, \dots, K_{n_1} , неся при этом долгосрочные затраты (затраты на вход) $c_1^0(K_1), \dots, c_{n_1}^0(K_{n_1})$, где $c_1^0, \dots, c_{n_1}^0$ — убывающие, выпуклые вниз функции.

Промежуточный, s -ый период. Пусть n_s фирм выбирают объемы производства — K_j, \dots, K_{j+n_s} , где $j = \sum_{i=1}^{s-1} n_i + 1$ неся при этом долгосрочные затраты $c_j^0, \dots, c_{j+n_s}^0$, где $c_j^0, \dots, c_{j+n_s}^0$ — убывающие, выпуклые вниз функции.

Последний, m -ый период. Все фирмы одновременно выбирают количества товара, q_1, \dots, q_n , поставляемого на рынок. Фирмы при этом несут краткосрочные затраты на производство c_1, \dots, c_n .

Кроме того, оставшиеся n_m фирм выбирают объемы производства — K_j, \dots, K_{j+n_m} , где $j = \sum_{i=1}^{m-1} n_i + 1$ неся при этом долгосрочные затраты $c_j^0(q_j), \dots, c_{j+n_m}^0(q_{j+n_m})$. Объемы производства берутся такими, что $K_j = q_j, \dots, K_{j+n_m} = q_{j+n_m}$ (в противном случае, когда $K_j \leq q_j, \dots, K_{j+n_m} \leq q_{j+n_m}$, фирмы несут некупаемые долгосрочные затраты).

Предельные затраты на производство для n фирм обозначим через $c_1(q), \dots, c_n(q)$, это убывающие, выпуклые вниз функции. Предельные затраты фирм, выбирающих объемы производства в последнем периоде, складываются из краткосрочных и долгосрочных затрат:

$$c_j^0(K_j) + c_j(q_j), j = \sum_{i=1}^{m-1} n_i + 1, \dots, n.$$

Для фирм n_i , где $i = 1, \dots, m-1$ (эти фирмы выбрали объемы производства раньше последнего периода), справедливо следующее: если $q_i < K_i$, то фирма i несет краткосрочные предельные затраты, равные $c_i(q)$, каждая же единица продукции сверх K_i сопряжена с долгосрочными предельными затратами в количестве $c_i^0(K_i) + c_i(q_i)$.

В зависимости от количества товара на рынке складывается цена $p = p(\sum_{i=1}^n q_i)$, — убывающая, выпуклая вниз функция $p' < 0, \dots, p'' > 0$.

Прибыль в последнем периоде будет находиться по формулам:

$$D_i = \begin{cases} q_i p - c_i K_i, & q_i \leq K_i, \\ q_i p - (c_i + c_i^0) q_i, & q_i > K_i, \end{cases} \quad i = 1, \dots, \sum_{j=1}^{m-1} n_j,$$

$$D_i = q_i p - (c_i + c_i^0) q_i, \quad i = \sum_{j=1}^{m-1} n_j + 1, \dots, n.$$

Многопериодная модель вхождения в рынок Курно n фирм.

В данном разделе рассматривается обобщение модели Диксита на случай нескольких фирм и нескольких периодов, функции предельных затрат предполагаются функциями количества товара, они убывают и выпуклы вниз, а функция ценообразования считается нелинейной.

Эта модель является многопериодной. Фирма обладает начальным капиталом и, в один из периодов, вкладывает его в производство. Через несколько периодов фирма выбирает количество товара, поставляемого на рынок. В результате совместного выбора нескольких фирм на рынке устанавливается цена, и фирмы могут получить прибыль, за счет которой в следующем периоде фирма может увеличить объем производства. Фирмы выпускают одинаковый товар, неразличимый с точки зрения покупателя.

Пусть имеется m периодов и n фирм.

Первый период. Пусть $n_1 < n$ фирм обладают начальным капиталом K_1, \dots, K_{n_1} , который вкладывают в производство. При этом фирмы несут предельные долгосрочные затраты, соответственно $c_1^0, \dots, c_{n_1}^0$, затраты являются функциями K_1, \dots, K_{n_1} . Эти функции убывают и выпуклы вниз. В дальнейшем, функции предельных долгосрочных затрат всех фирм предполагаются удовлетворяющими этим условиям.

$$\begin{cases} (c_1^0)' < 0, \dots, (c_{n_1}^0)' < 0, \\ (c_1^0)'' \geq 0, \dots, (c_{n_1}^0)'' \geq 0. \end{cases}$$

Долгосрочные затраты — это затраты на освоение производства, например, закупка оборудования, обучение персонала, аренда помещения и т. п. В данной задаче предполагается, что эти затраты необратимы, то есть нельзя продать оборудование за ту же цену, что оно покупалось, нельзя вернуть деньги, вложенные в обучение персонала и т. п.

Фирмы, которые выбрали объемы производства в данный момент, имеют разную скорость освоения производства, k_1, \dots, k_{n_1} (так как фирма осваивает производство за несколько периодов, то долгосрочные затраты в эти периоды могут быть разными, например, k — на установку станка, m — на обучение. Но в этой модели рассматриваются общие долгосрочные затраты, которые будут являться суммой долгосрочных затрат в каждом периоде, то есть $k + m$). Соответственно, через k_1, \dots, k_{n_1} периодов фир-

мы выберут количества товаров, которые они поставят на рынок, тогда же они получают прибыль и уже в следующем периоде, смогут расширить объем производства (если фирма понесет убытки, большие или равные начальному капиталу минус затраты на производство $K_j - c_j^0(K_j)K_j$, то ей придется свернуть производство).

Промежуточный, s -ый период. Пусть n_s фирм входят в рынок в период s . Обозначим множество таких фирм через N_s . Они выбирают объемы производства — $K_i, i \in N_s$, неся при этом предельные долгосрочные затраты c_i^0 . Все, сказанное про первые n_1 фирм в первом периоде также применимо и к n_s фирмам в s -ом периоде, также они осваивают производство с разными скоростями и прибыль смогут получить только через несколько периодов (см. предыдущий пункт).

Вместе с тем существуют фирмы, которые к данному моменту только освоили производство и те, которые уже освоили производство и теперь его расширяют. Обозначим множества таких фирм через N_o и N_r соответственно. Эти фирмы выбирают количество товара, поставляемого на рынок q_i , где $i \in N_o \cup N_r$. В результате совместного выбора на рынке складыва-

ется цена $p = p\left(\sum_{i=1}^{|N_o \cup N_r|} q_i\right)$.

При выпуске товара фирмы несут предельные краткосрочные затраты на производство $c_i(q_i)$, где $i \in N_o \cup N_r$, эти функции так же, как и функции долгосрочных затрат, убывают и выпуклы вниз. Предположение относительно вида функций предельных затрат (убывающие и выпуклые вниз функции), как долгосрочных так и краткосрочных, характеризует возрастающую отдачу от объемов производства. Далее рассматривается прибыль этих фирм.

Прибыль фирмы i , которая только к этому периоду освоила производство (т. е. $i \in N_o$)

$$D_i = q_i p - c_i^0(K_i)K_i - c_i(q_i)q_i,$$

где q_i удовлетворяет условию: $K_i - c_i^0(K_i)K_i - c_i(q_i)q_i > 0$.

Прибыль фирмы i , которая расширяет производство (т. е. $i \in N_r$)

$$D_i = q_i p - c_i(q_i)q_i,$$

где q_i удовлетворяет условию: $M_{i,s-1} - c_i(q_i)q_i > 0$. Здесь $M_{i,s-1}$ — свободный капитал i -ой фирмы в $s-1$ периоде.

Его можно найти следующим образом: до того момента, когда фирма впервые выбирает количество товара, поставляемого на рынок $M_{i,1} = K_i - c_i^0(K_i)K_i$ (ограничение, накладываемое на количество, q для фирм, только что выпустивших продукцию, можно переписать в виде

$M_{i,s-1} - c_i(q_i)q_i > 0$). Если фирма не выпускает продукцию, то свободный капитал не меняется от периода к периоду. После выпуска товара в s периоде $M_{i,s} = M_{i,s-1} + D_i$.

Последний период. Все фирмы выбирают количества товара, поставляемые на рынок q_1, \dots, q_n . Фирмы при этом несут краткосрочные затраты на производство c_1, \dots, c_n .

Кроме того, оставшиеся n_m фирм выбирают объемы производства - K_i , неся при этом долгосрочные затраты c_i^0 . Они выбирают объемы производства такими, что $K_i = q_i$ (в противном случае, когда $K_i \leq q_i$, фирмы несут некупаемые долгосрочные затраты).

Если оставшиеся фирмы не могут сразу же освоить производство, то они в рынок не входят т. е. $K_i = 0$. Прибыль находится аналогично предыдущему пункту.

Таким образом, построена модель игры с n участниками, фирмы конкурируют друг с другом в выборе количеств товара, поставляемого на рынок. Посредством MATLAB численно найдена точка равновесия по Нэшу.

УДК 519.15:681.42.2

GRTHEORY TOOLBOX — ИНСТРУМЕНТАРИЙ MATLAB ДЛЯ РЕШЕНИЯ ЗАДАЧ НА ГРАФАХ

Иглин С. П.

Национальный технический университет «ХПИ», Харьков,
e-mail: iglin@kpi.kharkov.ua

1. Введение

В стандартном MATLAB нет процедур для решения задач на графах. На сайте компании The MathWorks Inc. в коллекции файлов для обмена отсутствует даже такая категория [1]. Пользователи MATLAB пытаются устранить этот недостаток. Так, в [2, 3] приведены решения некоторых задач. Автором также написан ряд функций с использованием алгоритмов, описанных в книгах [4, 5], статье [6] и других работах. В настоящем докладе делается попытка на основе и тех, и других создать инструментарий для решения задач на графах. Он оформлен по обычным правилам MATLAB Toolbox: все процедуры собраны в один каталог, снабжены справочным файлом **Contents.m** и демонстрационной программой. Добавив этот каталог в список путей MATLAB, пользователь может использовать функции пакета наряду со стандартными, а также изменять их и дополнять пакет своими процедурами.

В докладе описываются функции пакета, порядок их вызова, используемые алгоритмы, приводятся примеры.

2. Рисование графа

Для рисования графов и орграфов разработана функция:

function h=PlotGraph(V,E,p)

Входные параметры:

- **V(n,2)** или **V(n,3)** — координаты вершин: первый столбец — x-е, второй — y-е. Если задан третий столбец, то предполагается, что задан граф со взвешенными вершинами, и третий столбец — это веса вершин. Вершины графа рисуются кружочками, в которых проставляются: для невзвешенного графа — номера вершин, а для взвешенного — их веса. Количество вершин **n** не задается, оно определяется автоматически по размеру массива **V**.
- **E(m,2)** или **E(m,3)** — список ребер графа (дуг орграфа) и, возможно, их веса. Первый и второй элементы каждой строки — это

номера вершин, которые соединяет данное ребро. Третий элемент, если он есть, это вес ребра. Если задан граф с невзвешенными ребрами, возле каждого ребра проставляется его номер, а если со взвешенными — то вес. Количество ребер **m** не задается, а определяется автоматически по размеру массива **E**. Для графа можно переставить 1-й и 2-й элементы каждой строки, для орграфа предполагается, что начало каждой дуги — в вершине, номер которой задан в 1-м элементе, а конец — во 2-м.

- **p** — символ, который определяет, что рисовать: '**g**' — граф, '**o**' — орграф. Это необязательный параметр, по умолчанию рисуется граф.

Выходной параметр **h** — дескриптор созданной фигуры.

Алгоритм работы функции следующий. Определяются границы рисунка. Находится минимальное расстояние между вершинами. Задается радиус кружков, которыми будут рисоваться вершины, как 0.1 от минимального расстояния между вершинами. Из списка ребер удаляются петли (они рисуются отдельно). Ребра сортируются так, чтобы 1-я вершина была меньше 2-й, 1-е номера шли в порядке возрастания, а при одинаковых 1-х номерах 2-е шли также в порядке возрастания. Это позволяет легко выявить кратные ребра. Петли также сортируются в порядке возрастания вершин. Рисуем вершины кружочками, проставляем в каждом из них номер вершины или ее вес. Простые ребра рисуем отрезками прямых, а кратные — дужками различных радиусов. Петли рисуем дугами окружностей, причем кратные — разными радиусами. Если задан орграф, вместо линий рисуем стрелки при помощи функции **arrow.m**, которую можно свободно переписать на сайте [1].

Рассмотрим пример рисования орграфа со взвешенными дугами. Вот фрагмент кода:

```
clear all
V=[[0 0];[1 1];[1 0];[1 -1];...
  [2 1];[2 0];[2 -1];[3 1];...
  [3 0];[3 -1];[4 0]]; % координаты вершин
E=[[ 1 2 5];[ 1 3 5];[ 1 4 5];...
  [ 2 3 2];[ 3 4 2];[ 2 5 3];...
  [ 2 6 2];[ 3 6 5];[ 3 7 2];...
  [ 4 7 3];[ 6 5 1];[ 6 7 1];...
  [ 5 8 5];[ 6 8 2];[ 6 9 3];...
  [ 7 9 2];[ 7 10 3];[ 8 9 2];...
  [ 9 10 2];[ 8 11 5];[ 9 11 4];...
  [10 11 4]]; % дуги и их веса
PlotGraph(V,E,'o'); % рисуем орграф
```

Полученный чертеж орграфа показан на рис. 1 (копия экрана).

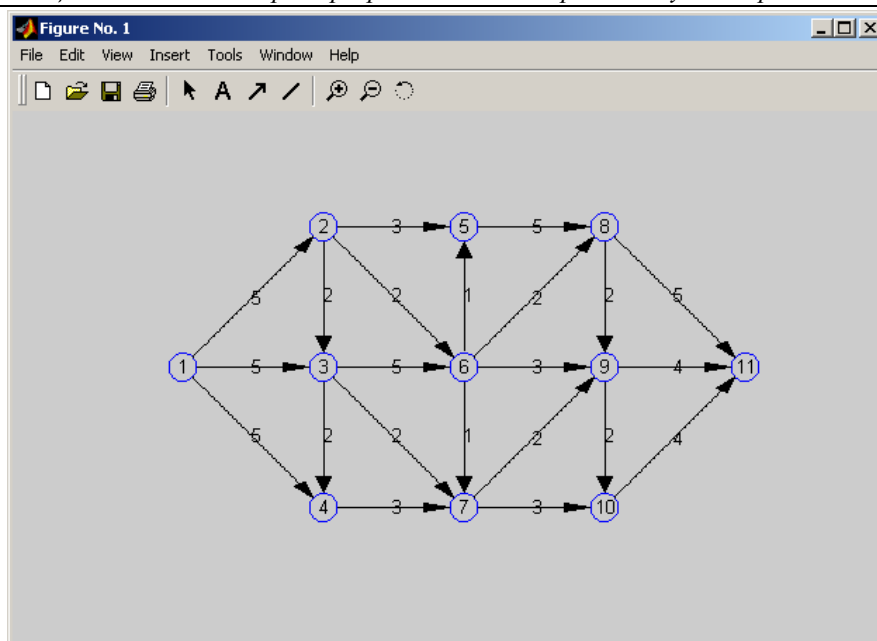


Рис. 1. Чертеж орграфа со взвешенными дугами.

3. Максимальный поток в сети

Сеть рассматривается как орграф со взвешенными дугами, например, такой, как показан на рис. 1. Одна из вершин считается источником сети и обозначается s . Другая вершина — это сток сети, он обозначается t . Например, в сети на рис. 1 можно задать $s=1$, $t=11$. Тогда источником сети будет 1-я вершина, а стоком — 11-я. Неотрицательные веса дуг w_k называются в этой задаче пропускными способностями. Требуется найти такие потоки в каждой дуге e_k , которые:

- 1) были бы неотрицательными и не превышали пропускных способностей w_k в своей дуге;
- 2) обеспечивали бы баланс потоков в каждом узле (вершине): алгебраическая сумма потоков в каждой вершине, кроме s и t , равна нулю;
- 3) давали бы максимальный суммарный поток в источнике.

В такой формулировке — это классическая задача линейного программирования. Действительно, пусть e_k , $k=[1, m]$ — неизвестные потоки в дугах, m — количество дуг, e — вектор e_k . Введем в рассмотрение матрицу инцидентности орграфа A размером $n \times m$ (n — количество вершин), каждый элемент которой $a_{ik}=1$, если из вершины v_i выходит дуга e_k ; $a_{ik}=-1$, если в вершину v_i входит дуга e_k ; и $a_{ik}=0$ в остальных случаях. Обозначим через $A^{(s)}$ s -ю строку матрицы A , а через $A^{(-st)}$ — матрицу A с удаленными строками с номерами s и t . Тогда задача о максимальном потоке формулируется так: требуется максимизировать линейную функцию

$$z = A^{(s)} e \rightarrow \max \quad (1)$$

при линейных ограничениях-равенствах на переменные e :

$$A^{(-st)}e = 0 \quad (2)$$

и заданных пределах изменения переменных:

$$0 \leq e \leq w. \quad (3)$$

Здесь w — вектор весов w_k , а неравенства (3) выполняются по каждой координате. В процедуре по данным о структуре орграфа формулируется задача линейного программирования (1–3), для ее решения вызывается функция **linprog**, а результаты решения возвращаются в вызывающую программу. Заголовок функции имеет вид:

function [v,mf]=MaxFlows(E,s,t)

Входные параметры:

- **E(m,2)** или **E(m,3)** — список дуг орграфа и, возможно, их веса, как в функции **PlotGraph**. Если задан орграф с невзвешенными дугами, то считается, что все веса — единичные.
- **s** и **t** — номера вершин, определяющих источник и сток сети.

Выходные параметры:

- **v(m,1)** — вектор-столбец потоков в дугах.
- **mf** — общий максимальный поток в сети.

Пример вызова этой функции для орграфа, изображенного на рис. 1:

```
s=1 % источник сети
t=11 % сток сети
[v,mf]=MaxFlows(E,s,t); % задача о максимальном потоке
disp('Решение задачи о максимальном потоке')
disp('  N      flow')
fprintf('  %2.0f    %12.8f\n',...
        [1:length(v)];v')
fprintf('Максимальный поток =%12.8f\n',mf)
```

Результат в командном окне MATLAB:

```
s =
    1
t =
   11
Решение задачи о максимальном потоке
  N      flow
  1    5.00000000
  2    5.00000000
  3    3.00000000
  4    0.63432629
  5    0.00000000
  6    2.79872259
  7    1.56695111
  8    4.12415344
```

9	1.51017285
10	3.00000000
11	0.93404510
12	0.17003857
13	3.73276770
14	1.85198251
15	2.73503838
16	1.86897174
17	2.81123968
18	0.58475020
19	1.18876032
20	5.00000000
21	4.00000000
22	4.00000000

Максимальный поток = 13.00000000

4. Максимальное реберное покрытие (паросочетание)

Задача о максимальном паросочетании формулируется следующим образом: для заданного графа требуется найти такое подмножество ребер $E' \subseteq E$ максимальной мощности, в котором никакие два ребра не будут инцидентны. Обобщением этой задачи является задача о максимальном взвешенном паросочетании, когда требуется найти подмножество ребер максимального суммарного веса, среди которых нет инцидентных.

Различные алгоритмы решения задачи о максимальном паросочетании рассматриваются в [4–5, 7–9]. Мы воспользуемся тем, что данная задача может быть сформулирована как задача целочисленного линейного программирования (ЦЛП). Учитывая, что пользователями MATLAB разработаны эффективные процедуры решения задачи ЦЛП [3, 10], мы получим достаточно простую и эффективную процедуру решения задачи о максимальном (взвешенном) паросочетании.

Введем в рассмотрение переменные, ассоциированные с ребрами. Обозначим их так же, как и ребра: e_1, e_2, \dots, e_m . Каждое из них может принимать одно из двух значений: 1, если соответствующее ребро входит в максимальное паросочетание, и 0, если нет. Тогда задача о максимальном реберном покрытии сводится к максимизации линейной функции:

$$z = (\mathbf{1}, \mathbf{e}) \rightarrow \max \quad (4)$$

при условии, что в каждой вершине сумма инцидентных ей переменных (ребер) e_k не превышает единицы. Это ограничение можно записать в виде:

$$A\mathbf{e} \leq \mathbf{1}. \quad (5)$$

Здесь A — матрица инцидентности графа размером $n \times m$, \mathbf{e} — вектор-столбец переменных e_1, e_2, \dots, e_m , $\mathbf{1}$ — вектор из единиц нужной размерности. В матрице инцидентности A элемент $a_{ik} = 1$, если вершина v_i инцидентна ребру e_k , и $a_{ik} = 0$ в противном случае. Кроме того, на переменные e_k накладываются условия целочисленности:

$$e_k = 0 \vee 1; k = [1, m]. \quad (6)$$

Задача о максимальном взвешенном паросочетании отличается только целевой функцией: вместо (4) имеем:

$$z = (w, e) \rightarrow \max. \quad (7)$$

Разработанная функция по заданной структуре графа (и, при необходимости, весах ребер) строит целевую функцию вида (4) или (7) и ограничения (6), дополненные условиями целочисленности (6). Далее решается задача ЦЛП, а результаты возвращаются в вызывающую программу. Задача ЦЛП решается с помощью функции **milp**, которую можно свободно переписать с сайта [3]. Заголовок функции:

function nMM=MaxMatch(E)

Входной параметр **E(m,2)** или **E(m,3)** — список ребер графа и, возможно, их веса, как в функции **PlotGraph**. Если задан граф с невзвешенными ребрами, то считается, что все веса — единичные. Выходной параметр **nMM** — список номеров ребер, входящих в максимальное паросочетание (вектор-столбец). Для графа с рис.1 решение задачи о максимальном взвешенном паросочетании может быть получено так:

```
nMM=MaxMatch(E); % решаем задачу
PlotGraph(V,E(nMM,:), 'g'); % рисуем ребра
```

В результате мы рисуем ребра, входящие в максимальное взвешенное паросочетание (рис. 2).

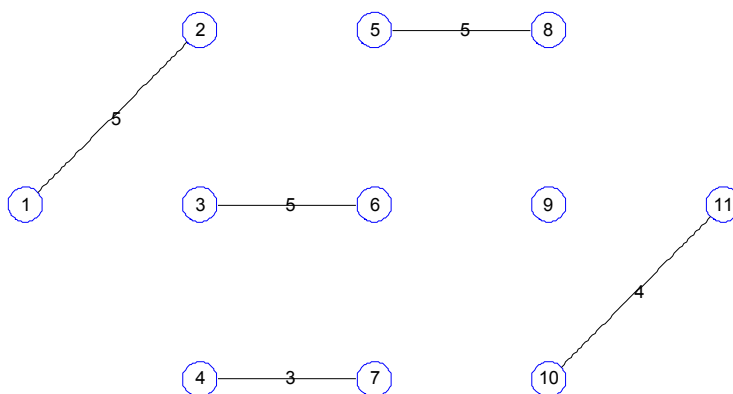


Рис. 2. Решение задачи о максимальном взвешенном паросочетании.

5. Минимальное вершинное покрытие

В этой задаче требуется найти подмножество вершин $V' \subseteq V$ минимальной мощности, которые были бы инцидентны всем ребрам графа. Обобщение — задача о минимальном взвешенном вершинном покрытии,

когда требуется минимизировать суммарный вес вершин, входящих в V' .

Данная задача также может быть приведена к задаче ЦЛП. Для этого введем в рассмотрение переменные, ассоциированные с вершинами. Мы их обозначим так же, как и вершины: v_1, v_2, \dots, v_n . Каждая из них может принимать одно из двух значений: 1, если соответствующая вершина входит в минимальное вершинное покрытие, и 0, если нет. Имеем задачу ЦЛП: нужно минимизировать общее количество вершин:

$$t = (\mathbf{1}, \mathbf{v}) \rightarrow \min \quad (8)$$

при условии, что каждому ребру инцидентна хотя бы одна вершина, т. е. сумма v_i , инцидентных каждому ребру, не менее единицы. Это ограничение можно записать так:

$$A^T \mathbf{v} \geq \mathbf{1}, \quad (9)$$

где \mathbf{v} — вектор-столбец переменных v_1, v_2, \dots, v_n . Ограничения (9) дополняются условиями целочисленности переменных:

$$v_i = 0 \vee 1; i = [1, n]. \quad (10)$$

Для взвешенной задачи вместо целевой функции (8) имеем:

$$t = (\mathbf{d}, \mathbf{v}) \rightarrow \min, \quad (11)$$

где \mathbf{d} — вектор весов вершин.

Функция MATLAB для решения этой задачи по заданной структуре графа (и, при необходимости, весах вершин) строит целевую функцию вида (8) или (11) и ограничения (9), дополненные условиями целочисленности (10). Далее с помощью функции **milp** решается задача ЦЛП, а результаты возвращаются в вызывающую программу. Заголовок функции:

function nMC=MinVerCover(E,d)

Входные параметры:

- **E(m,2)** — список ребер графа, как в функции **PlotGraph**.
- **d** — веса вершин (необязательный параметр). Если веса не заданы, то они считаются единичными.

Выходной параметр **nMC** — список номеров вершин, входящих в минимальное вершинное покрытие. Для графа с рис. 1 решение задачи о минимальном вершинном покрытии может быть получено так:

nMC=MinVerCover(E)

Результат — номера вершин, входящих в минимальное вершинное покрытие:

nMC =

2
3
4
6

6. Минимальное остовное дерево связного графа

Для заданного связного графа требуется найти подмножество ребер $E' \subseteq E$ минимальной мощности (или минимального веса), которые связывают все вершины графа и не образуют циклов. Как доказано в [5], для решения этой задачи применим «жадный» алгоритм, который так же прост, как и его название: бери больше, отдавай меньше. Вот его схема:

1. Выбираем ребро минимального веса — оно обязательно войдет в остовное дерево.
2. Просматриваем все ребра, инцидентные уже построенному остовному дереву и не образующие в нем циклов. Выбираем из них ребро минимального веса. Присоединяем его к строящемуся дереву.
3. Если в остовном дереве охвачены не все вершины, идем на шаг 2, иначе заканчиваем работу алгоритма.

Этот алгоритм реализован в функции:

function nMST=MinSpanTree(E)

Входной параметр **E(m,2)** или **E(m,3)** — список ребер графа и, возможно, их веса, как в функции **PlotGraph**. Если задан граф с невзвешенными ребрами, то считается, что все веса — единичные. Выходной параметр **nMST** — список номеров ребер, входящих в минимальное остовное дерево (вектор-столбец). Для графа с рис. 1 решение задачи об остовном дереве минимального веса может быть получено так:

```
nMST=MinSpanTree(E); % решаем задачу  
PlotGraph(V,E(nMST,:), 'g'); % рисуем решение
```

В результате мы рисуем ребра, входящие в минимальное остовное дерево (рис. 3).

7. Кратчайший путь между вершинами орграфа

Для орграфа со взвешенными дугами требуется найти минимальное расстояние (сумму весов дуг) между заданной парой вершин. По мнению авторов [5], в настоящее время наилучшим способом решения этой задачи является алгоритм Флойда-Уоршелла (R.W.Floyd, S.A.Warshall), который позволяет находить сразу матрицу кратчайших путей между всеми вершинами орграфа. Этот алгоритм описан в [5] и запрограммирован в функции:

function dSP=ShortPath(E)

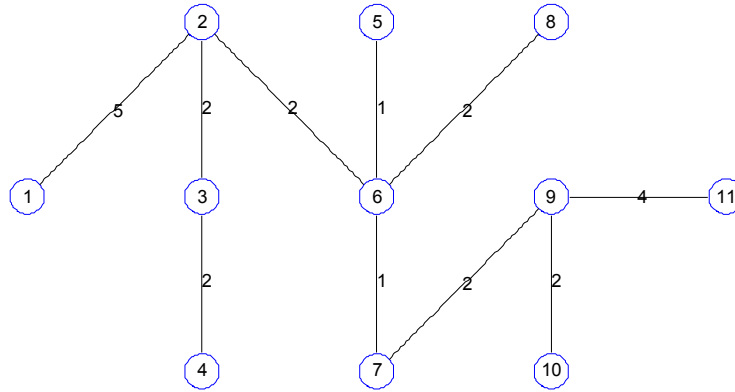


Рис. 3. Решение задачи об остоном дереве минимального веса.

Ее входной параметр **E(m,2)** или **E(m,3)** — список дуг орграфа и, возможно, их веса, как в функции **PlotGraph**. Если задан орграф с невзвешенными дугами, то считается, что все веса — единичные. Выходной параметр **dSP** — матрица $n \times n$ кратчайших расстояний между вершинами орграфа. Если из вершины **i** вершина **j** недоступна, то **dSP(i,j)=inf**. Вот как вычисляются кратчайшие пути между всеми парами вершин орграфа с рис. 1:

```
dSP=ShortPath(E);
disp('Кратчайшие пути между вершинами:')
fprintf('  %2.0f',1:size(dSP,2))
fprintf('\n')
for k1=1:size(dSP,1),
    fprintf('%2.0f',k1)
    fprintf('%6.2f',dSP(k1,:))
    fprintf('\n')
end
```

В результате получаем матрицу расстояний:

Кратчайшие пути между вершинами:

	1	2	3	4	5	6	7	8	9	10	11
1	Inf	5.00	5.00	5.00	8.00	7.00	7.00	9.00	9.00	10.00	13.00
2	Inf	Inf	2.00	4.00	3.00	2.00	3.00	4.00	5.00	6.00	9.00
3	Inf	Inf	Inf	2.00	6.00	5.00	2.00	7.00	4.00	5.00	8.00
4	Inf	Inf	Inf	Inf	Inf	Inf	3.00	Inf	5.00	6.00	9.00
5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5.00	7.00	9.00	10.00
6	Inf	Inf	Inf	Inf	1.00	Inf	1.00	2.00	3.00	4.00	7.00
7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	2.00	3.00	6.00
8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	2.00	4.00	5.00
9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	2.00	4.00
10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	4.00
11	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

8. Правильная раскраска вершин графа

В заданном графе требуется так раскрасить вершины, чтобы использовать минимально возможное количество цветов, и чтобы инцидентные вершины были раскрашены разными цветами.

Эта задача, как и многие другие, также сводится к задаче ЦЛП. Введем в рассмотрение переменные, ассоциированные с вершинами: v_1, v_2, \dots, v_n . Каждая из них может принимать целые значения от 1 до n , которые означают номера красок. Имеем задачу оптимизации:

$$\max(v_k) \rightarrow \min, \quad (12)$$

при условии, что для каждого ребра e_{ik} цвета вершин v_i и v_k должны быть разными:

$$\forall e_{ik}: |v_i - v_k| \geq 1, \quad (13)$$

и все v_i — целые в интервале от 1 до n :

$$\forall v_i \in \mathbb{N} \cap [1, n]. \quad (14)$$

Задачу (12–14) легко сформулировать в терминах ЦЛП, если ввести в рассмотрение целочисленную переменную v_0 , принимающую возможные значения от 1 до n , и целочисленные переменные e_{ik} , ассоциированные с ребрами, которые могут принимать значения 0 или 1. Имеем задачу ЦЛП:

$$v_0 \rightarrow \min, \quad (15)$$

при ограничениях–неравенствах:

$$\forall v_i \leq v_0; \quad (16)$$

$$\forall e_{ik}: \begin{cases} v_i - v_k - ne_{ik} \leq -1; \\ v_k - v_i + ne_{ik} \leq n - 1; \end{cases} \quad (17)$$

и условиях целочисленности:

$$\forall v_i \in \mathbb{N} \cap [1, n]; \quad \forall e_{ik} = 0 \vee 1. \quad (18)$$

Смысл ограничений-неравенств (17): при $e_{ik} = 0$ или 1 цвета вершин v_i и v_k отличаются по крайней мере на единицу в ту или другую сторону.

Функция MATLAB для решения этой задачи по заданной структуре графа строит целевую функцию вида (15), ограничения (16–17), дополненные условиями целочисленности (18). Задача ЦЛП решается с помощью функции **miqp**, которую можно свободно переписать на сайте [10]. Результаты возвращаются в вызывающую программу. Заголовок функции:

function nCol=ColorGraph(E)

Входной параметр **E(m,2)** — список ребер графа, как в функции **PlotGraph**. Выходной параметр **nCol(n,1)** — номера красок для каждой вершины. Для графа с рис. 1 раскраска вершин может быть получена так:

```
nCol=ColorGraph(E); % решаем задачу
PlotGraph([V(:,1:2),nCol],E(:,1:2),'g'); % рисуем решение
```

В результате мы рисуем граф, в котором вместо номеров или весов вершин проставляем номера красок (рис. 4).

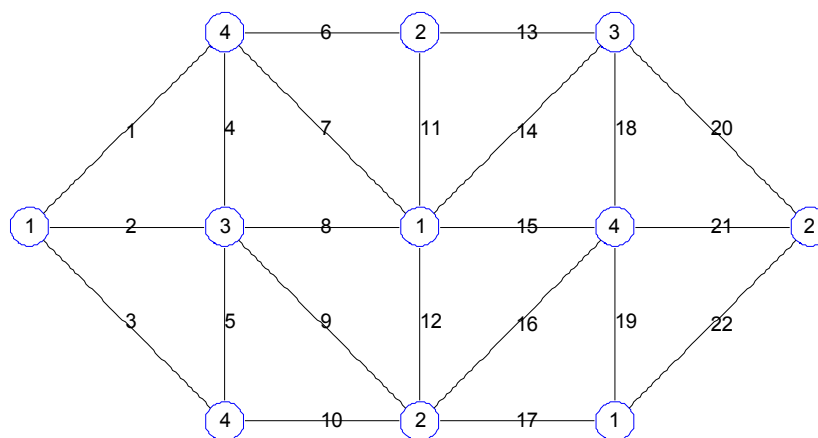


Рис. 4. Раскраска вершин графа.

9. Несимметричная задача коммивояжера

Задана квадратная матрица C размером $n \times n$ (в общем случае несимметричная) расстояний между любой парой из n городов. Требуется составить кольцевой маршрут объезда всех городов минимальной протяженности.

Введем в рассмотрение полный взвешенный граф (клику) K_n , где вершины v_i — это города, дуги e_{ik} — соединяющие их дороги (дуга e_{ik} направлена от города номер i к городу номер k), а вес каждого ребра w_{ik} — это расстояние между соответствующими городами, причем в общем случае $w_{ik} \neq w_{ki}$. Получили задачу нахождения полного цикла минимального веса в ориентированной клике.

Сформулируем эту задачу в терминах ЦЛП. Пусть $e_{11}, e_{12}, e_{13}, \dots, e_{1n}, e_{21}, \dots, e_{n,n-1}$ — переменные, ассоциированные с дугами. Каждая из переменных равна 1, если соответствующая дуга входит в маршрут, и 0, если не входит. Общее число переменных в ориентированной клике K_n равно $n(n-1)$. Имеем задачу ЦЛП: нужно минимизировать функцию

$$z = \sum_{\substack{i,k=1 \\ i \neq k}}^n w_{ik} e_{ik} \rightarrow \min \quad (19)$$

при следующих ограничениях:

1) все переменные равны 0 или 1:

$$\begin{cases} e_{ik} = 0 \vee 1; \\ i, k = \overline{1, n}; i \neq k; \end{cases} \quad (20)$$

2) из каждой вершины выходит одна дуга:

$$\begin{cases} \sum_{k=1}^n e_{ik} = 1; \\ k \neq i \\ i = \overline{1, n}; \end{cases} \quad (21)$$

3) в каждую вершину входит одна дуга:

$$\begin{cases} \sum_{i=1}^n e_{ik} = 1; \\ i \neq k \\ k = \overline{1, n}; \end{cases} \quad (22)$$

4) все циклы — полные, т. е. состоят из n вершин:

$$\begin{cases} u_i - u_k + (n-1) e_{ik} \leq n-2; \\ 2 \leq i \neq k \leq n, \end{cases} \quad (23)$$

где u_i — неограниченные действительные переменные. Ограничения (23) были предложены в [11]. Их смысл такой. Любой цикл, не проходящий через вершину v_1 , противоречит (23). Поэтому все допустимые циклы должны включать вершину v_1 . В совокупности с (21–22) это означает, что допустимыми будут только полные циклы из n дуг. И наоборот, любой цикл из n дуг при надлежащем выборе переменных u_i будет удовлетворять (23).

В процедуре по заданной матрице расстояний C формируется задача ЦЛП (19–23). Она решается с помощью функции **miqp**. По результату решения формируется ответ, который возвращается в вызывающую программу. Заголовок функции:

function [pTS,fmin]=TravSale(C)

Входной параметр C — квадратная матрица расстояний между городами. Выходные параметры:

- **pTS** — порядок обхода городов: вектор-строка длиной $(n+1)$, первый и последний элементы которой равны 1 (начало маршрута взято в 1-м городе);
- **fmin** — найденная минимальная длина маршрута.

Для примера зададим несимметричную матрицу расстояний и определим маршрут.

```
C=[0 3 7 4 6 4];...
  [4 0 3 7 8 5];...
  [6 9 0 3 2 1];...
  [8 6 3 0 9 8];...
  [3 7 4 6 0 4];...
  [4 5 8 7 2 0]]
[pTS,fmin]=TravSale(C);
disp('Порядок обхода городов:');
fprintf('%d ',pTS)
```

```
fprintf('\nМинимальная длина маршрута =%3.0f\n',fmin)
```

Результат работы:

C =

0	3	7	4	6	4
4	0	3	7	8	5
6	9	0	3	2	1
8	6	3	0	9	8
3	7	4	6	0	4
4	5	8	7	2	0

Порядок обхода городов:

1 4 2 3 6 5 1

Минимальная длина маршрута = 19

10. Разбиение орграфа на подграфы из взаимно достижимых вершин и их частичное упорядочение

С точки зрения теории множеств орграф можно рассматривать как множество V из n элементов (вершин), дополненное бинарным соотношением между некоторыми парами его элементов (дугами). Будем обозначать дугу символом \Rightarrow и словом «достигается». Т. е. запись $v_i \Rightarrow v_k$ означает, что из вершины v_i достигается вершина v_k , или существует дуга, идущая из v_i в v_k . Для достижимости выполняются такие аксиомы.

1. *Рефлексивность*: $\alpha \Rightarrow \alpha$. Даже если петли в каждой вершине и не заданы, будем считать, что, если мы находимся в вершине α , то мы уже ее достигли, т. е. из любой вершины эта же вершина достижима.
2. *Транзитивность*: если $(\alpha \Rightarrow \beta) \cap (\beta \Rightarrow \gamma)$, то $\alpha \Rightarrow \gamma$. Функция **ShortPath**, рассмотренная в разделе 7, строит матрицу достижимости **dSP** как раз с учетом этой аксиомы: любой ее конечный элемент **dSP(i,j)** соответствует достижимости от вершины **i** к вершине **j**.
3. *Антисимметричность*: если $(\alpha \Rightarrow \beta) \cap (\beta \Rightarrow \alpha)$, то $\alpha \sim \beta$, где символом \sim обозначена эквивалентность. Согласно этой аксиоме взаимно достижимые вершины считаются эквивалентными.

В соответствии с этими аксиомами множество вершин V может быть разбито на непересекающиеся классы эквивалентности, в каждый из которых попадают эквивалентные (взаимно достижимые) вершины, и эти классы частично упорядочены по достижимости. Так, например, в орграфе на рис. 1 никакие вершины не эквивалентны между собой по достижимости. Поэтому в каждый класс эквивалентности попадает только по одной вершине. Если поменять местами номера вершин 5 и 6, то получим частичное упорядочение вершин по достижимости: из вершины с меньшим номером можно перейти только в вершину с большим номером. Но это упорядочение не полное, а только частичное, т. к., например, из вершины 4 вершина 8 не достижима.

Рассматриваемая в этом разделе функция разбивает множество вершин орграфа на классы эквивалентности и проводит их частичное упорядочение по достижимости. Используемый алгоритм описан в [6], где он применяется для анализа взаимоотношений в коллективе, выявления лидеров и групп влияния. Заголовок функции:

function [Decomp,PartOrder]=DecompPartOrder(E)

Входной параметр **E(m,2)** — список дуг орграфа, как в функции **PlotGraph**. Выходные параметры:

- **Decomp(n,ns)** — булевский массив номеров вершин, входящих в каждый класс эквивалентности. Здесь **n** — количество вершин, **ns** — количество классов эквивалентности. В каждом столбце истинные элементы соответствуют номерам вершин, входящих в данный класс. Столбцы упорядочены по достижимости.
- **PartOrder(ns,ns)** — булевский массив частичной упорядоченности. Истинному элементу **PartOrder(i,j)** соответствует достижимость из зоны номер **i** (вершины, входящие в **Decomp(:,i)**) в зону номер **j**, т. е. до всех вершин из **Decomp(:,j)**. В силу частичной упорядоченности все истинные элементы этого массива расположены выше главной диагонали и на ней (рефлексивность).

В качестве примера рассмотрим орграф с 25 вершинами и 46 дугами. Обращение к функции имеет вид:

```
V1=[0 4];[1 4];[2 4];[3 4];[4 4];...
    [0 3];[1 3];[2 3];[3 3];[4 3];...
    [0 2];[1 2];[2 2];[3 2];[4 2];...
    [0 1];[1 1];[2 1];[3 1];[4 1];...
    [0 0];[1 0];[2 0];[3 0];[4 0];
E1=[1 2];[ 3 2];[ 4 3];[ 5 4];[ 6 1];...
    [ 2 7];[ 8 2];[ 3 8];[ 9 4];[ 9 5];...
    [10 5];[ 7 6];[ 8 7];[ 8 9];[10 9];...
    [11 6];[ 7 12];[13 8];[14 9];[15 10];...
    [12 11];[13 12];[13 14];[14 13];[15 14];...
    [16 11];[12 17];[13 18];[20 15];[17 16];...
    [17 18];[18 17];[19 18];[19 20];[21 16];...
    [17 22];[18 22];[22 18];[18 23];[19 24];...
    [20 25];[21 22];[22 21];[23 24];[24 23];[24 25];
PlotGraph(V1,E1,'o'); % рисуем орграф
[Decomp,PartOrder]=DecompPartOrder(E1);
disp('Зоны с взаимно достижимыми вершинами:')
disp(' N   вершины')
for k1=1:size(Decomp,2),
    fprintf('%2.0f ',k1)
    fprintf('%d ',find(Decomp(:,k1)))
    fprintf('\n')
end
```



```
fprintf('Частичное упорядочение зон:\n ')
fprintf('%3.0f',1:size(PartOrder,2))
fprintf('\n')
for k1=1:size(PartOrder,1),
    fprintf('%2.0f ',k1)
    fprintf(' %1.0f ',PartOrder(k1,:))
    fprintf('\n')
end
```

На рис. 5 показан нарисованный оргграф, а далее — результат решения задачи.

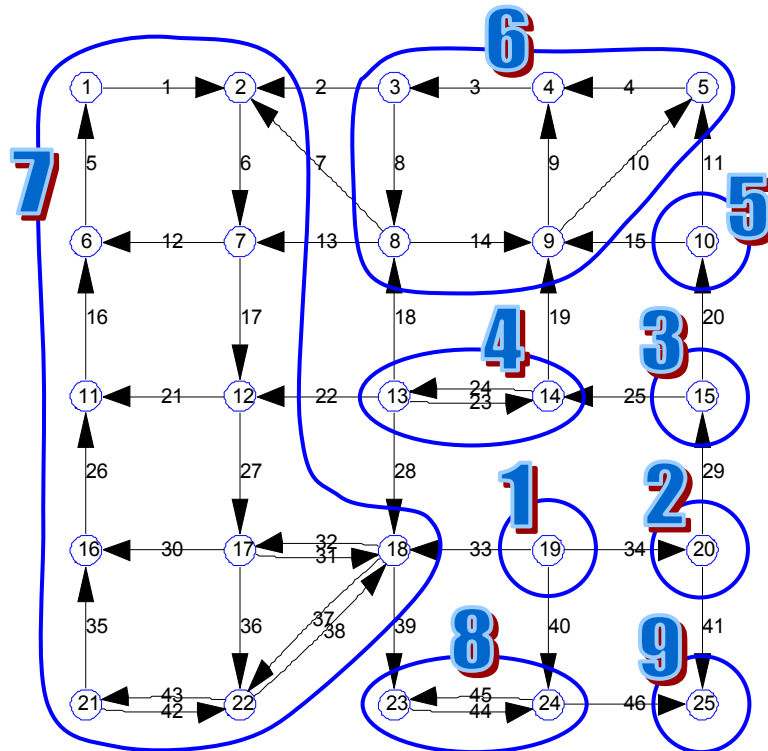


Рис. 5. Пример оргграфа с 25 вершинами и 46 дугами.

Зоны с взаимно достижимыми вершинами:

N вершины

- 1 19
- 2 20
- 3 15
- 4 13 14
- 5 10
- 6 3 4 5 8 9
- 7 1 2 6 7 11 12 16 17 18 21 22
- 8 23 24
- 9 25

Частичное упорядочение зон:

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	1	1

```

4 0 0 0 1 0 1 1 1 1
5 0 0 0 0 1 1 1 1 1
6 0 0 0 0 0 1 1 1 1
7 0 0 0 0 0 0 1 1 1
8 0 0 0 0 0 0 0 1 1
9 0 0 0 0 0 0 0 0 1

```

Множество вершин данного орграфа разбивается на 9 частей (зон), в каждой из которых все вершины взаимно достижимы. Эти зоны показаны на рис. 5 замкнутыми линиями с номерами. Из матрицы частичного упорядочения следует, что почти из любой зоны с меньшим номером можно достигнуть зоны с большим номером. Исключение составляют зоны 4 и 5: из зоны 4 достигнуть зоны 5 нельзя.

11. Полная система независимых циклов

При расчетах электрических цепей по законам Кирхгофа цепь удобно представить в виде графа с n вершинами (узлы цепи) и m ребрами (ветви). Чтобы определить токи в m ветвях, нужно составить систему из m уравнений. Первый закон Кирхгофа (баланс токов в каждом из n узлов равен нулю) дает только $n-1$ независимых уравнений, т. к. на n уравнений баланса накладывается одно ограничение: общий баланс токов в цепи равен нулю. Оставшиеся $m-n+1$ уравнений получаются из второго закона Кирхгофа: в каждом замкнутом контуре суммарная ЭДС равна суммарному падению напряжений. Для автоматического построения этих уравнений нужно найти $m-n+1$ независимых контуров, охватывающих все ребра графа, кроме «хвостов», которые не входят ни в какой контур. В инструментарий GrTheory Toolbox включена процедура для построения таких циклов. Ее заголовок:

function Cycles=CycleBasis(E)

Входной параметр **E(m,2)** — список ребер графа, как в функции **PlotGraph**. Выходной параметр **Cycles(m,m-n+1)** — булевский массив номеров ребер, входящих в независимые циклы. В каждом столбце истинные элементы соответствуют номерам ребер, входящих в данный цикл.

Алгоритм решения задачи такой. Сначала строится остовное дерево данного графа, которое состоит из $n-1$ ребер. Для этого используется функция **MinSpanTree**, описанная в разделе 6. Затем мы берем одно из оставшихся ребер, присоединяем его к остовному дереву (получается цикл), и в полученном последовательно удаляем все висячие ребра, оставляя только цикл. Т. к. после построения остовного дерева остается $m-n+1$ ребер, то в результате мы получим $m-n+1$ разных циклов, в каждый из которых входит хотя бы одно ребро, которого нет в других циклах. Значит,

полученные циклы — независимые, и их — максимально возможное число: $m - n + 1$.

Рассмотрим пример мостиковой схемы: граф с 7 вершинами, 9 ребрами и двумя «хвостами». Ниже приведен ввод его данных, а на рис. 6 — рисунок графа.

```
V2=[[0 2];[1 2];[2 2];[3 2];...
    [0 1];[1 1];[0 0]];
E2=[[1 2];[2 3];[3 4];...
    [1 5];[1 6];[2 5];[2 6];[5 6];[5 7]];
PlotGraph(V2,E2);
```

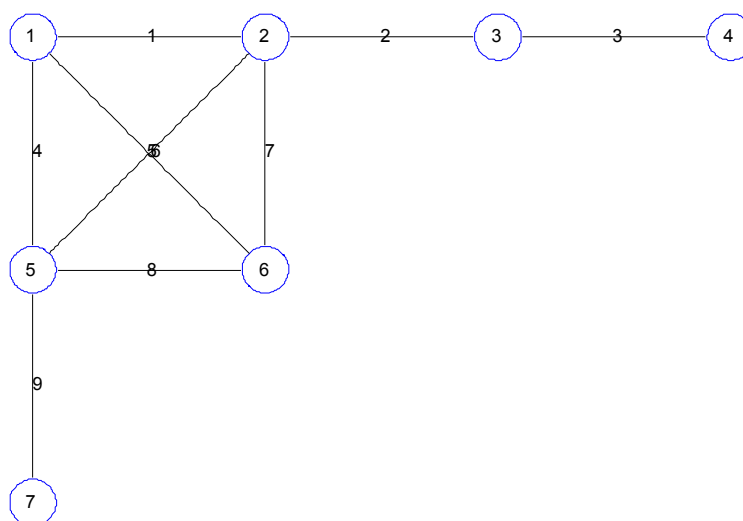


Рис. 6. Пример графа с 7 вершинами, 9 ребрами и 2 «хвостами».

Найдем и нарисуем все независимые циклы данного графа. Фрагмент кода приведен ниже, а результаты — на рис. 7-9.

```
Cycles=CycleBasis(E2);
for k1=1:size(Cycles,2),
    PlotGraph(V2,E2(find(Cycles(:,k1)),:));
    title(['Cicle N' num2str(k1)])
end
```

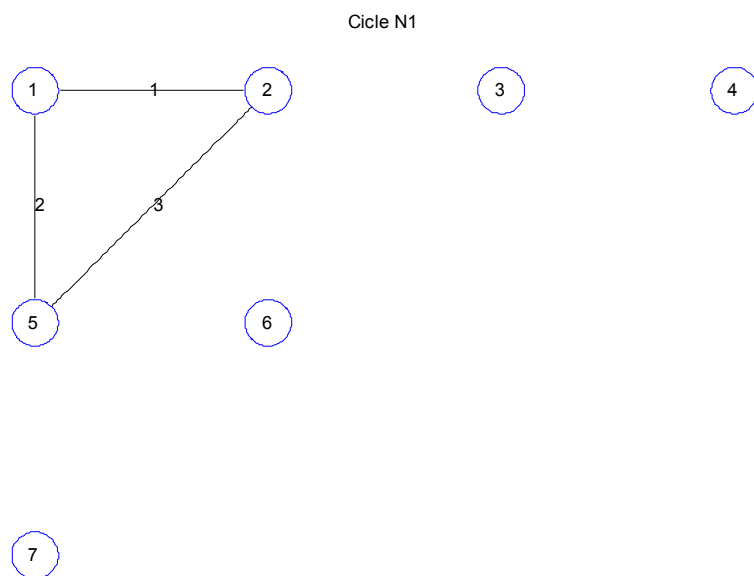


Рис. 7. Цикл 1.

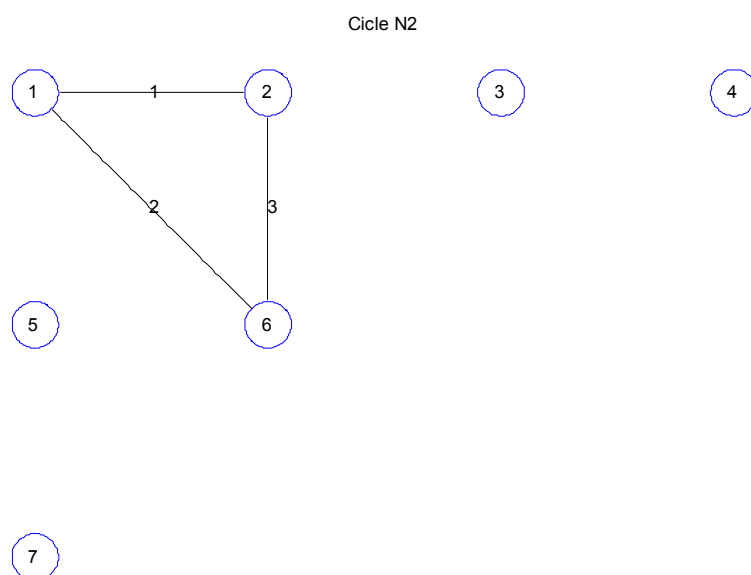


Рис. 8. Цикл 2.

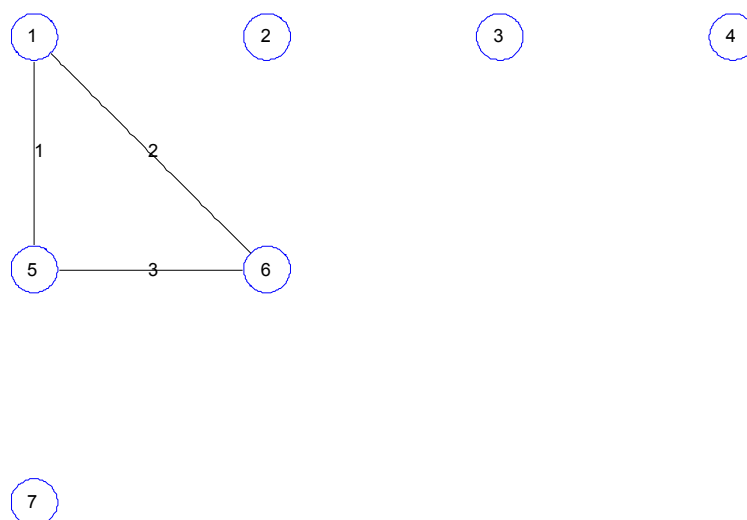


Рис. 9. Цикл 3.

12. Выводы

1. В докладе рассмотрен новый инструментарий для решения задач на графах: GrTheory Toolbox. В него включены процедуры для рисования графов и орграфов, решения задач о максимальном паросочетании, минимальном вершинном покрытии, максимальном потоке, минимальном остовном дереве, раскраске, кратчайшем пути и др.
2. Инструментарий оформлен по обычным правилам MATLAB: все функции помещены в один каталог, дополнены справочной информацией и примерами. Пользователю остается добавить этот каталог в список путей MATLAB.
3. Все функции записаны в исходном коде, что дает возможность пользователю изменять их и добавлять свои функции.
4. Инструментарий размещен на сайте [1], страница «Математика — общие вопросы» и может быть свободно скопирован.

Литература

1. *MATLAB Central File Exchange* (<http://www.mathworks.com/MATLABcentral/fileexchange/loadCategory.do>).
2. Software by Kevin Murphy (<http://www.ai.mit.edu/~murphyk/Software/>).
3. *Matlog: Logistics Engineering MATLAB Toolbox* (<http://www.ie.ncsu.edu/kay/matlog/>).
4. Алгоритмы и программы решения задач на графах и сетях / Нечипуренко М. И., Попков В. К., Майнагашев С. М. и др. — Новосибирск: Наука. Сиб. отд-ние, 1990. — 515 с.

5. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность.— М.: Мир, 1985.— 512 с.
6. *Грицак В. В., Грицак Ю., Иглин С. П.* Стационарная алгоритмическая модель общественных отношений // Соционика, ментология и психология личности.— 2002.— №1(40).
7. *Micali S., Vazirani Av. V.* An $O(\sqrt{|V|} |E|)$ algorithm for finding maximum matching in general graphs // Proc. 21st Ann. Symp. On the Foundations of Comp. Sci., Long Beach, California: IEEE.— 1980.— P.17–27.
8. *Edmonds J.* Maximum matching and a polyhedron with 0-1 vertices // J. of Res. Nat. Bur. Of Stand., 69B.— 1965.— P.125–130.
9. *Gabow H. N.* An efficient implementation of Edmonds algorithm for maximum matching on graphs // J. ACM.— 1976.— V.23.— P.221–234.
10. *TOMLAB Optimization Inc.*(<http://tomlab.biz/>).
11. *Miller C. E., Tucker A. W., Zemlin R. A.* Integer Programming Formulation of Traveling Salesman Problems. // J. ACM.— 1960.— V.7.— P.326–329.

УДК 004

РАЗРАБОТКА ИНТЕРАКТИВНОЙ СРЕДЫ РАСЧЕТА МАГНИТНОЙ СИСТЕМЫ МАРИ СПЕКТРОМЕТРА НА БАЗЕ ПАКЕТА MATLAB

Кальнеус Е. В., Стась Д. В.

*Институт химической кинетики и горения СО РАН, Новосибирск,
e-mail: kalneus@ns.kinetics.nsc.ru*

Введение

При изучении короткоживущих радикалов (десятки наносекунд) в последнее время успешно используется метод МАРИ (Magnetically Affected Reaction Yield) спектроскопии [1]. При этом резонансные линии МАРИ (зависимости интенсивности рекомбинационной флуоресценции облучаемого образца от внешнего магнитного поля) наблюдаются в нулевом и слабых (~ 1 -10 мТ) магнитных полях. Однако существующий на сегодня МАРИ спектрометр, основанный на модифицированной установке ЭПР не является оптимальным для данного класса задач и не удовлетворяет дальнейшим потребностям развития метода МАРИ. В связи с этим разработка магнитной системы для нового МАРИ спектрометра представляла актуальную научно-техническую задачу. В данной работе представлен метод расчета магнитной системы, выполненный в среде MATLAB с использованием графического интерфейса пользователя (GUI). Показаны преимущества данного подхода: быстрота реализации, гибкость исполнения и легкость модификации для решения широкого круга аналогичных задач. Фактически в процессе данной работы на основе GUI была создана графическая среда для разработки, анализа и представления результатов решаемой задачи.

Постановка задачи

Магнитная система нового МАРИ спектрометра разделяется на ряд подсистем. Это силовая магнитная система, создающая магнитное поле в образце с относительно большим шагом по абсолютной величине, сканирующая магнитная система, с помощью которой в процессе эксперимента производится плавная развертка по полю, система для компенсации остаточного земного поля и система модуляции, служащая в конечном итоге, для улучшения отношения сигнал/шум. Особенностью магнитной системы разрабатываемого спектрометра является отсутствие любых ферромагнитных материалов в системе, что позволяет складывать магнитные поля от

разных систем аддитивно. Силовая, сканирующая системы и система модуляции спроектированы как системы аксиально-симметричных колец, а система компенсации — как две перпендикулярные друг другу пары прямоугольных катушек для компенсации соответствующих компонент земного поля. Основная задача, которая ставилась в процессе вычислений — это нахождение параметров колец силовой и сканирующей систем таких, чтобы создаваемое ими магнитное поле в относительно протяженном рабочем цилиндрическом объеме (8х1см) было максимально однородно (абсолютная неоднородность поля не хуже 5×10^{-6} Тл) и максимально быстро спадало за пределами рабочей области, так чтобы на расстоянии 15-18см от образца максимальная его величина не превышала 1×10^{-2} Тл. Малая величина магнитного поля вне рабочей области позволяет размещать магниточувствительные элементы спектрометра, такие как рентгеновская трубка и ФЭУ, достаточно близко от образца, что приводит к увеличению чувствительность прибора в целом.

Расчет магнитных систем

В общем случае синтез магнитных систем обычно сводится к нахождению таких их параметров, чтобы квадрат отклонения создаваемого магнитного поля $B(r)$ от желаемого поля $f(r)$ в необходимом объеме V был минимален, т. е. производится минимизация функционала $F = \int_V (B(\vec{r}) - f(\vec{r}))^2 dv$. Однако для системы N коаксиальных колец однородность поля вдоль оси системы z автоматически означает однородность поля в некотором прилегающем объеме [2]. Поэтому в нашем случае для расчета параметров системы достаточно было минимизировать

$$F = \int_{z_1}^{z_2} \left(\frac{B(z) - f(z)}{B_0(z)} \right)^2 dz, \quad (1)$$

где $B(z_0)$ — поле в центре системы. За основу расчетов была взята методика, предложенная в [3]. Однако, при минимизации функционала (1) мы варьировали не только геометрические параметры колец, но и токи в них, т. е. для каждой пары колец было три независимых параметра. В ходе расчетов интеграл (1) заменялся на сумму

$$\sum_{i=1}^M \left(\frac{B(z_i) - f(z_i)}{B(z_0)} \right)^2, \quad (2)$$

в которой для системы из N тонких круговых витков с током I_k , радиусом R_k , расположенных на расстоянии Z_k от начала координат по оси z , поле $B(z_i)$ вычислялось в эквидистантно расположенных M точках в рассчи-

тываемой области $-l < z_i < l$. При этом, для системы бесконечно тонких колец, согласно закону Био-Савара,

$$B(z) = \frac{\mu_0}{2} \sum_{k=1}^N I_k R_k^2 [R_k^2 + (z - Z_k)^2]^{-3/2}, \quad (3)$$

где $\mu_0 = 4\pi \cdot 10^{-7} \text{ Гн/м}$ — магнитная проницаемость вакуума.

В случае толстых колец, т. е. когда сечением обмоток пренебречь нельзя, поле на оси z согласно [3] можно выразить как

$$B(z) = \frac{\mu_0}{2} \sum_{k=1}^N J_k \left(\begin{aligned} &(z_{2k} - z) \ln \frac{a_{2k} + [a_{2k}^2 + (z_{2k} - z)^2]^{1/2}}{a_{1k} + [a_{1k}^2 + (z_{2k} - z)^2]^{1/2}} - \\ &(z_{1k} - z) \ln \frac{a_{2k} + [a_{2k}^2 + (z_{1k} - z)^2]^{1/2}}{a_{1k} + [a_{1k}^2 + (z_{1k} - z)^2]^{1/2}} \end{aligned} \right), \quad (4)$$

где a_{1k} и a_{2k} — внутренний и внешний радиусы колец ($a_{1k} = r_{0k} - 1/2\Delta r_k$, $a_{2k} = r_{0k} + 1/2\Delta r_k$), z_{1k} и z_{2k} — положение левой и правой стороны кольца ($z_{1k} = z_{0k} - 1/2\Delta z_k$, $z_{2k} = z_{0k} + 1/2\Delta z_k$).

Расчет системы с экранированием поля имеет свои особенности. Для нахождения параметров этой системы минимизировалась сумма $\delta = k \sum_{i=1}^M (B(z_i) - f(z_i))^2$, где $f(z) \equiv 1$ и $k = k_1$ при $-l_1 < z < l_1$; $f(z) \equiv 0$ и $k = k_2$ при $-l_3 < z < l_2$ и $l_2 < z < l_3$. Т. к. так как для нашей задачи важна была высокая однородность поля в центре системы (относительное среднеквадратичное отклонение не более 1×10^{-4}), а на области экранирования достаточно было только малости абсолютной величины поля, то при расчетах для сумм квадратичных отклонений в областях однородности и экранирования были введены относительные веса k_1 и k_2 , где $k_1 > k_2$, а на заключительном этапе минимизации $k_1 \gg k_2$. При поиске оптимальной конфигурации колец M полагалось равным 20–100, т. к. от дальнейшего его увеличения результат практически не меняется. Минимизация производилась с использованием функций **fminunc** и **fmincon**.

Последовательность расчетов была следующей. Сначала, исходя из (3), находятся параметры системы для тонких колец. Затем рассчитываются сечения обмоток, учитывая полученные оптимальные величины токов I_k и заданную их плотность J_k . Т. к. все обмотки включаются последовательно, то J_k в первом приближении принимаются одинаковыми. Сечения обмоток первоначально выбираются квадратными ($\Delta z_k = \Delta r_k$). Затем, используя (4), производится поиск оптимальных размеров для сечений обмоток. Центры сечений обмоток z_{0k} , r_{0k} при этом фиксированы и являются найденными ранее параметрами для тонких колец. Если после данной оптимизации оказывается, что обмотки пересекаются в пространстве, накладываются дополнительные ограничения или

кладываются дополнительные ограничения или изменяются начальные параметры и процесс расчета повторяется вновь с поиска оптимальных z_{0k} и r_{0k} . После успешного завершения этого этапа производится упаковка обмотки, т. е. длина и ширина сечения каждой обмотки изменяются так, чтобы они были кратны ближайшему целому числу диаметров обмоточного провода. Плотности токов J_k в пересчете на равномерное распределение тока по сечению определяются на завершающем этапе с учетом конечной толщины изоляции проводов и толщины прокладок между слоями в обмотке. После этого при фиксированных Δr_k , Δz_k и J_k окончательно находятся оптимальные z_{0k} и r_{0k} .

Первой рассчитывалась система с экранированием. При этом учитывались следующие геометрические ограничения. Для размещения компенсирующей системы внутри рассчитываемой системы должен оставаться свободный цилиндрический объем не менее 10 см в длину и 10 см диаметром. При этом для экранирования образца от низкочастотных переменных полей и защиты от рентгеновского излучения данный объем должен быть размещен внутри алюминиевого цилиндра толщиной не менее 1 см. Для свободного доступа ампулы с образцом расстояние между ближайшими внутренними кольцами должно быть не менее 5 см, а для максимально близкого расположения рентгеновской трубки и ФЭУ расстояние между центром системы и внешней стороной последнего кольца должно быть не более 15 см. Чтобы обеспечить свободный аксиальный доступ к рабочему объему внутренний диаметр колец должен быть более 5 см. Затрачиваемая мощность для создания магнитного поля 400 Гс не должна превышать 1 кВт. Относительная неоднородность поля на интервале ± 4 см от центра системы должна быть не хуже 1×10^{-4} , а остаточное поле в области экранирования не должно превышать $0.1 \times B_{\max}$. Расчеты показали, что три пары колец, две из которых включены в одном направлении, а третья служит для экранировки, являются минимальной конфигурацией, которая удовлетворяет заданным требованиям. При этом теоретически достижимая однородность поля в заданном объеме $\sim 5 \times 10^{-6}$, а мощность для создания поля 0.04 Тл при этом ~ 800 Вт. Так как для рассеивания такой мощности необходимо водяное охлаждение, то при оптимизации параметров колец помимо перечисленных геометрических ограничений учитывались также ограничения на взаимное расположение колец и системы охлаждения. Результаты проведенных расчетов представлены на рис. 1, 2.

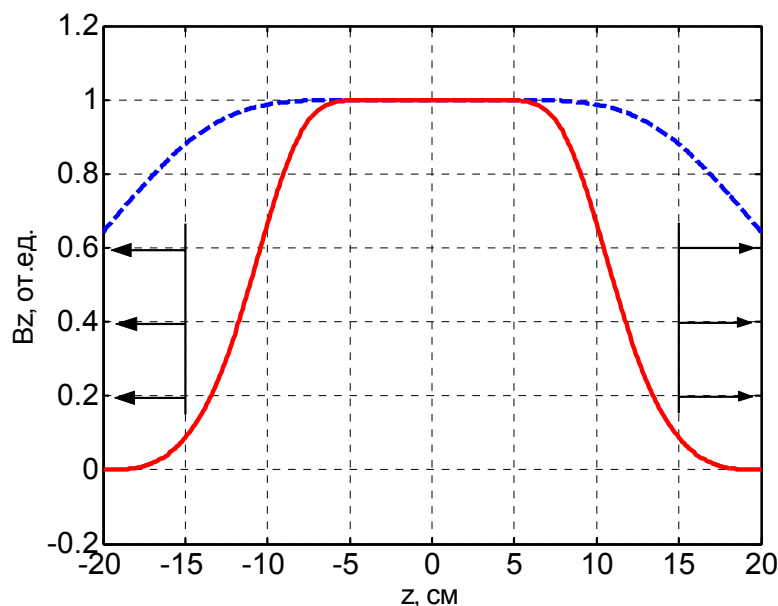


Рис. 1. Сравнительная зависимость V_z компонент поля силовой (сплошная линия) и сканирующей (пунктирная линия) систем по оси z . Стрелками показана область активного экранирования.

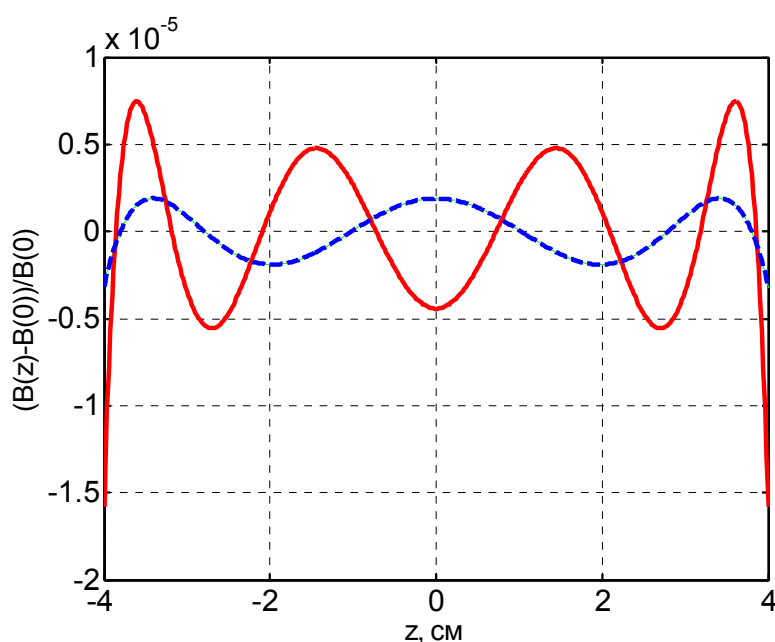


Рис. 2. Зависимость относительного изменения V_z компоненты поля силовой (сплошная линия) и сканирующей (пунктирная линия) систем по оси z в рабочей области.

После расчета силовой системы находились параметры сканирующей магнитной системы. Основными ограничениями в этом случае было непересечение с уже рассчитанной силовой системой и требование относительной неоднородности поля в рабочем объеме не хуже 1×10^{-4} . Удовлетворить предъявляемым требованиям по однородности и по затрачиваемой

мощности с помощью только одной пары колец не удастся. Поэтому была выбрана конфигурация с двумя парами катушек. Надо отметить, что при данной оптимизации не ставилась задача получения максимальной теоретической однородности поля в заданном объеме. Важно было лишь, чтобы однородность была не хуже определенной величины и при этом были соблюдены все геометрические ограничения.

В результате оптимизационных расчетов удалось так подобрать параметры систем, что при всех требуемых ограничениях для силовой системы относительное среднеквадратичное отклонение B_z компоненты поля в рабочей области составило 4.31×10^{-6} , а для сканирующей системы 1.39×10^{-6} (рис. 2). Разрез магнитной системы в целом представлен на рис. 3.

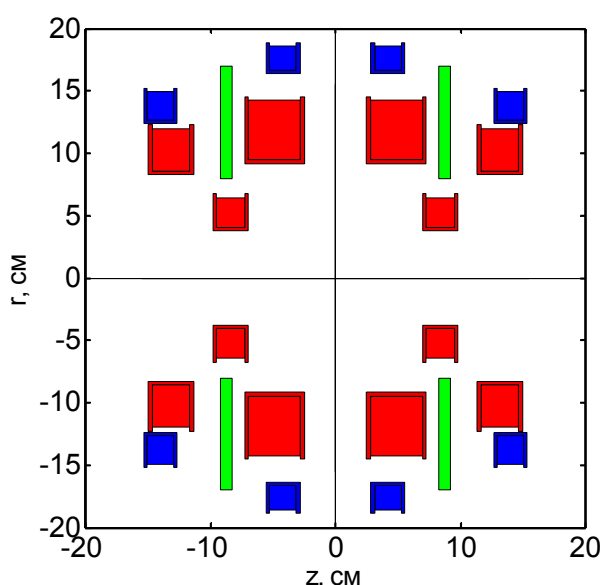


Рис. 3. Разрез магнитной системы плоскостью, проходящей через ось системы z . Три пары симметричных колец (красный цвет) — силовая система. Две пары (синий цвет) — сканирующая система. Пара тонких колец (зеленый цвет) — охлаждающая система.

После проведения оптимизационных расчетов для B_z составляющей магнитного поля силовой и сканирующей систем были проведены контрольные расчеты аксиальной B_z и радиальной B_r компонент обеих систем во всем рабочем пространстве. При этом для расчетов использовалась встроенная функция **ellipke** для вычисления полных эллиптических интегралов первого и второго рода $K(k)$ и $E(k)$, с помощью которых согласно [4] B_r и B_z компоненты поля от тонкого кольца радиуса a с током I можно представить как

$$B_r = \frac{\mu_0}{2\pi} I \frac{z}{r} \left[(r+a)^2 + z^2 \right]^{-1/2} \times \left(-K(k) + \frac{a^2 + r^2 + z^2}{(a-r)^2 + z^2} E(k) \right),$$

$$B_z = \frac{\mu_0}{2\pi} I \left[(r+a)^2 + z^2 \right]^{-1/2} \times \left(K(k) + \frac{a^2 - r^2 - z^2}{(a-r)^2 + z^2} E(k) \right),$$

где $k^2 = 4ra / [(r+a)^2 + z^2]$.

В процессе данных контрольных расчетов поле от каждой реальной катушки моделировалось суммой полей n тонких витков с одинаковым током I и параметрами, соответствующими реальному расположению центра n -го витка в обмотке, при этом n — реальное число витков обмотки для данной катушки. Графики контрольного расчета B_r и B_z компонент полей силовой системы представлены на рис. 4. Видно, что магнитное поле в рабочем объеме (± 4 см по z и ± 0.5 см по r в цилиндрических координатах) направлено только по оси z и удовлетворяет предъявляемым требованиям по однородности.

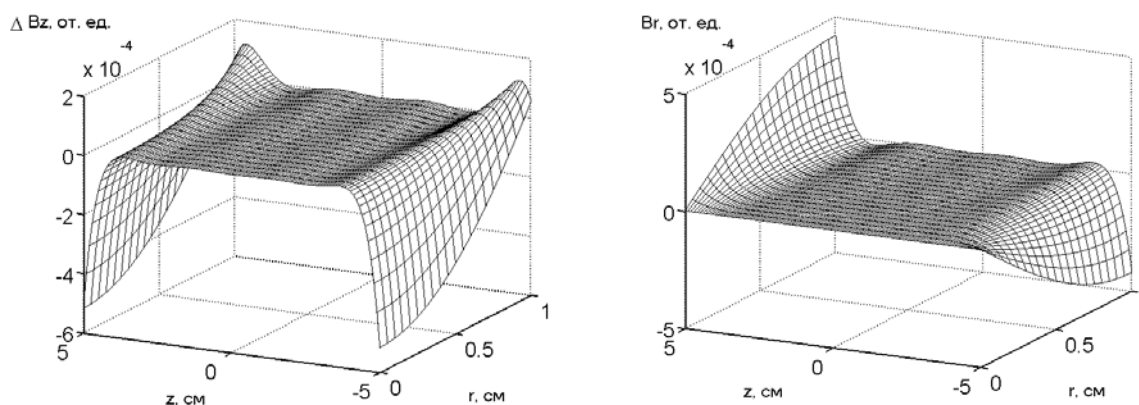


Рис. 4. Относительное отклонение B_z компоненты и B_r компонента магнитного поля, создаваемого силовой системой в рабочей области.

После получения реальных геометрических параметров был проведен анализ влияния ошибок изготовления колец системы на однородность создаваемого магнитного поля. На рис. 5 показаны зависимости изменения однородности B_z компоненты магнитного поля на оси системы z в рабочей области от относительного симметричного изменения одного из параметров соответствующей пары колец силовой системы при сохранении остальных геометрических параметров оптимальными. Из данного графика можно оценить верхнюю границу неоднородности поля при заданной ошибке изготовления. Видно, что при характерном радиусе системы 10 см и точности изготовления катушек ± 0.1 мм однородность поля $\sim 5 \times 10^{-4}$.

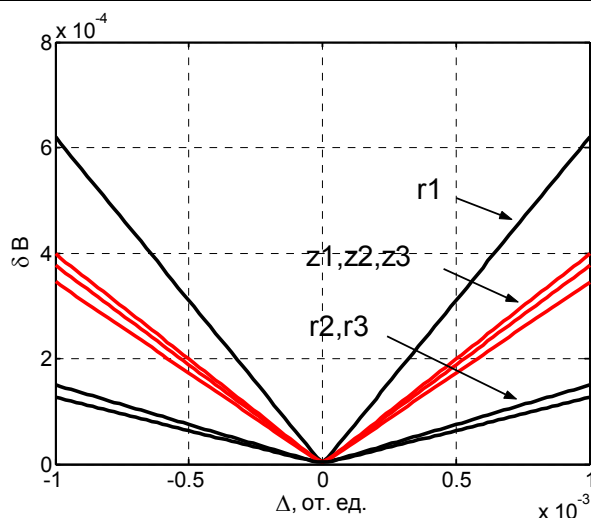


Рис. 5. Зависимости изменения однородности B_z компоненты магнитного поля на оси системы z в рабочей области от относительного симметричного изменения одного из параметров соответствующей пары колец силовой системы при сохранении остальных геометрических параметров оптимальными. Стрелками показаны кривые для изменяемых параметров.

Однако, т. к. увеличение неоднородности за счет отклонения одного параметра может в существенной степени компенсироваться изменением другого параметра, то при одновременном учете всех случайных неточностей изготовления, однородность создаваемого поля может оказаться выше, чем при учете отклонения от оптимального только одного параметра. На рис. 6 представлена гистограмма распределения однородности B_z компоненты поля силовой системы вдоль оси z на рабочем интервале при случайном распределении относительных отклонений геометрических параметров силовой системы в пределах $\pm 10^{-3}$. Видно, что максимум распределения соответствует $\sim 10^{-4}$, что подтверждает высказанное предположение.

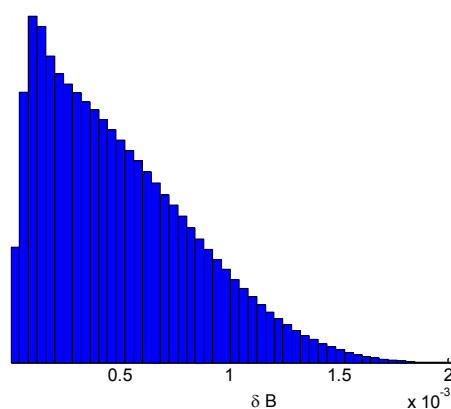


Рис. 6. Гистограмма распределения относительного среднеквадратичного отклонения B_z компоненты поля силовой системы в рабочей области при равномерном случайном относительном отклонении параметров в диапазоне 10^{-3} .

После расчета силовой и сканирующей систем производилась оптимизация для компенсирующей магнитной системы. В силу конструктивных особенностей здесь удобнее было использовать прямоугольные кольца. Соответствующие компоненты полей при этом можно найти, используя выражения [4] для компонент поля от прямоугольного кольца с током I :

$$B(x) = \frac{\mu_0 I}{4\pi} \sum_{i=1}^4 (-1)^{i+1} \frac{z}{r_i(r_i + y_i)},$$

$$B(y) = \frac{\mu_0 I}{4\pi} \sum_{i=1}^4 (-1)^{i+1} \frac{z}{r_i(r_i + x_i)},$$

$$B(z) = \frac{\mu_0 I}{4\pi} \sum_{i=1}^4 (-1)^i \frac{r_i(x_i + y_i) + x_i^2 + y_i^2}{r_i(r_i + x_i)(r_i + y_i)},$$

где $x_1 = x_4 = x + a$, $x_2 = x_3 = x - a$, $y_1 = y_2 = y + b$, $y_3 = y_4 = y - b$.

В данных обозначениях a и b — половины сторон прямоугольного кольца, а r_i — вектор, проведенный в точку наблюдения от i -го его угла. Начало координат расположено в центре кольца, а ось z направлена перпендикулярно его плоскости. Задачей расчета компенсирующей системы являлось нахождение геометрических параметров двух колец, симметрично расположенных относительно начала координат, так чтобы однородность создаваемого ими магнитного поля $B(z)$ вдоль оси x на расстоянии ± 4 см от центра была максимальна. Так как рассчитываемая пара колец в этом случае должна быть вписана в цилиндрический объем заданного радиуса и длины, то для варьирования остается только один параметр. Для компенсирующей системы расчет проводился только в приближении тонких колец, т. к. поле, создаваемое этой системой мало ($\sim 1 \times 10^{-4}$ Тл) и, следовательно, мало сечение обмоток. Результаты расчета представлены на рис. 7.

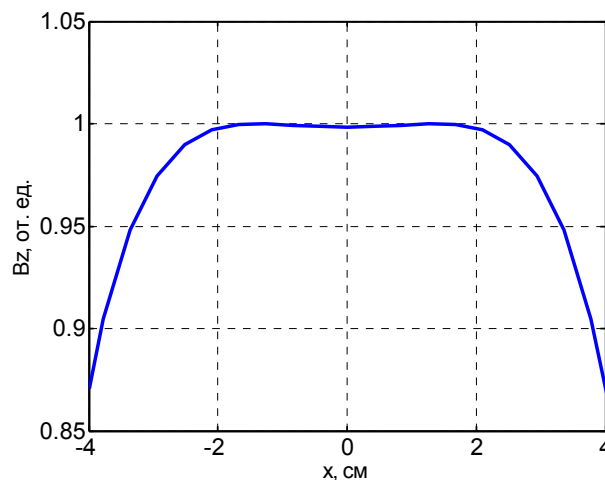


Рис. 7. Зависимость B_z компоненты поля одной пары компенсирующих колец на оси системы в рабочем интервале.

Разработка интерактивной среды

Так как минимизируемая функция отклонения магнитного поля имеет множество локальных минимумов в многомерном пространстве варьируемых параметров и при этом минимум, который находит процедура минимизации сильно зависит от начальных параметров в том числе, то для того чтобы ускорить процедуру минимизации была создана интерактивная система, позволяющая выводить результаты не только в численном виде, но и в графическом. При этом автоматически учитывались и отображались все необходимые геометрические ограничения. Это существенно ускорило и упростило процесс задания начальных значений и весь процесс разработки в целом. Графическое окно интерфейса (рис. 8) разделено на ряд функциональных областей. Это область для ввода файла данных и выбора из него нужных записей, область задания общих параметров для выбора расчета, область начальных параметров минимизации, область для графического и численного отображения результатов минимизации.

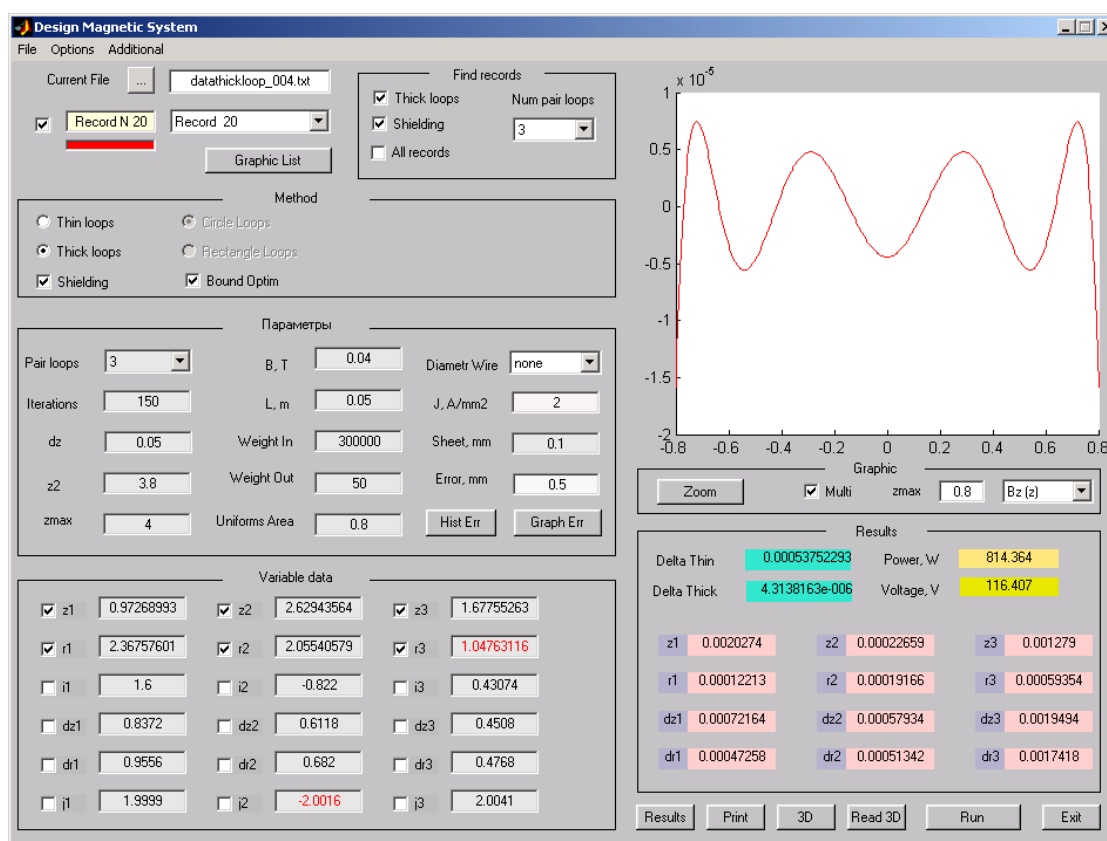


Рис. 8. Окно графического интерфейса программы для разработки магнитных систем.

В ходе работы был разработан формат для представления данных. Все расчеты можно сохранять в одном файле, а в процессе его открытия можно по установленным флажкам отсортировать по ряду критериев все входящие в него записи и легко найти необходимую в данный момент. Для

удобства анализа результатов разработан способ одновременного графического просмотра и сравнения рассчитанных вариантов. При этом выбранные записи заносятся в список, который легко редактируется. Данная часть графического интерфейса реализована группой подпрограмм, которые могут быть переориентированы для удобного графического просмотра и редактирования данных других приложений. Например, на их основе была разработана программа для просмотра МАРИ спектров. Вывод графического изображения может быть осуществлен не только на внутреннее графическое окно программы. Нажатием одной кнопки содержимое внутреннего окна может быть открыто в стандартном графическом окне MATLAB (рис. 9), что позволяет использовать все возможности графического вывода, редактирования и печати изображения и не загромождать программу ненужными в процессе расчета кнопками. Из стандартных возможностей реализована только кнопка «zoom», реально необходимая при быстром просмотре полученных результатов. Надо отметить, что с большинством графических объектов связаны соответствующие контекстные меню. Это также позволило не перегружать основное окно программы лишней информацией и в то же время реализовать все необходимые функции. Контекстное меню графического окна позволяет выбрать вид графического представления (компоненты поля B_z вдоль оси системы, либо разрез плоскостью, проходящей через ось системы), а также относительный или абсолютный масштаб представления. При этом в разрезе отображается не только сечение намотки колец, но и весь каркас катушки с учетом задаваемой толщины стенок каркаса, а также дополнительно могут быть включены заранее заданные другие геометрические ограничения, например, разрез системы охлаждения.

При наших расчетах все вводимые параметры оказалось возможным разместить в основном окне программы, хотя, например, при аналогичных расчетах для системы из большего числа пар колец можно легко организовать их ввод и вывод в отдельном графическом окне. Так как в расчетах использовалась функция минимизации с ограничениями, задание последних для каждого отдельного параметра осуществлялось введением необходимых максимальных и минимальных значений в модальном окне диалога после щелчка правой кнопкой мыши на соответствующем параметре. Ряд других параметров также имеют связанные с ними контекстные меню, которые позволяют уточнить выбор или задать дополнительные действия.

Для каждой пары рассчитываемой системы колец максимальное число параметров минимизации равно трем и пяти для тонких и толстых колец соответственно. При этом любой из этих параметров может быть независимым образом отключен. Когда запускается программа расчета, то минимизация автоматически проводится только для включенных параметров. В программе также реализован ряд сервисных функций, основанных на изменении цвета элементов, сокрытии недоступных объектов в данный

момент расчета или представлении их в неактивном варианте. Примером может служить изменение цвета рассчитываемого параметра, если он в ходе расчетов стал равен ограничивающему значению. Нужно также отметить, что любые изменения, вводимые вручную в текущие параметры сразу же приводят к соответствующему изменению графического изображения, что делает легким и удобным просмотр зависимости влияния этого параметра на конечный результат, а также упрощает процесс подбора начальных значений.

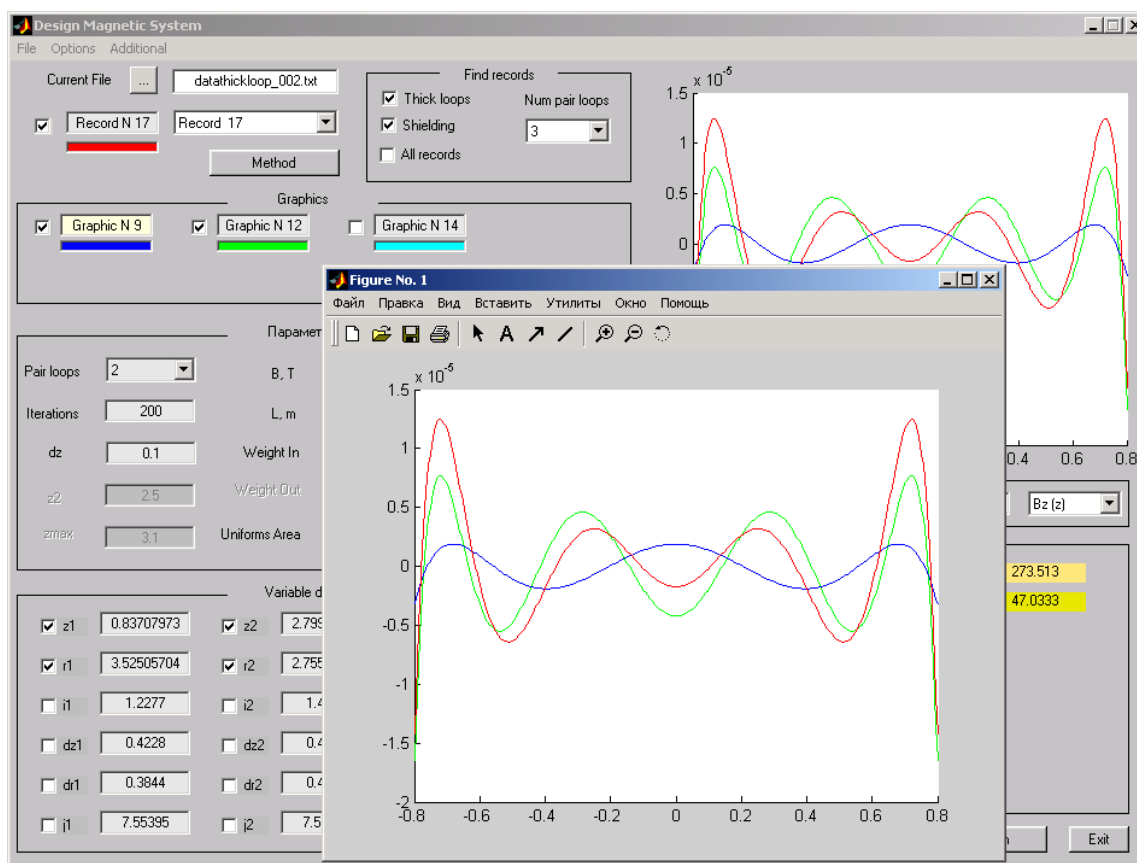


Рис. 9. Иллюстрация одновременного просмотра нескольких вариантов расчетов и возможности быстрого вывода графических данных в отдельное стандартное окно MATLAB.

Помимо собственно разработки, расчета и проверки в данную программу вошел также анализ влияния возможных ошибок изготовления на выходные параметры магнитной системы в целом. Ряд подпрограмм, написанных для этой цели запускаются соответствующими кнопками, а результаты отображаются по выбору во внутреннем или внешнем графических окнах. Сами данные таких расчетов сохраняются в отдельных бинарных файлах, открыть которые и задать параметры для их просмотра можно из меню «Additional».

Результаты

В ходе работы в среде MATLAB на основе графического интерфейса пользователя был разработан программный комплекс для расчета магнитной системы МАРИ спектрометра. По заданным параметрам и ограничениям рассчитаны силовая, сканирующая и компенсационная магнитные системы, а также проведен анализ влияния возможных ошибок изготовления на однородность магнитного поля в рабочей области.

Благодарности

Авторы выражают благодарность академику Ю.Н. Молину и д. ф.-м. н. Ю.А. Гришину за полезные обсуждения в ходе выполнения данной работы.

Работа выполнена при финансовой поддержке РФФИ, проект 03-03-32331, и Минвуза, проект E02-5.0-49.

Литература

1. *Sviridenko F. B., Stass D. V., Molin Yu. N.* Study of interaction of aliphatic alcohols with primary radical cations of n-alkanes using MARY spectroscopy // *Mol. Phys.*— 101.— 2003.— P.1839–1850.
2. *Афанасьев Ю. В., Студенцов Н. В., Хорев В. Н., Чечурина Е. Н., Щелкин А. П.* Средства измерения параметров магнитного поля.— Л.: Энергия. Ленингр. отд-ние, 1979.
3. *Lugansky L. B.* Optimal coils for producing uniform magnetic fields // *J. Phys. E: Sci.Instrum.*— 20.— 1987.— P.277–285.
4. *Malmivuo J., Lekkala J., Kontro P., Suomaa L., Vihinen H.* Improvement of the properties of an eddy current magnetic shield with active compensation // *J. Phys. E: Sci.Instrum.*— 20.— 1987.— P.151–160.

УДК 537

ПАКЕТ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ РАСПРЕДЕЛЕНИЯ БИОПОТЕНЦИАЛОВ ЭЛЕКТРИЧЕСКОЙ АКТИВНОСТИ СЕРДЦА НА ОСНОВЕ РЯДА ЛАПЛАСА

Кацюба О. А., Гуцин А. В.,

*Самарская государственная академия путей сообщения, Самара,
e-mail:stdtcen@samtel.ru*

Угнич К. А.

Самарский государственный медицинский университет, Самара

Методы обработки статистических данных, существующих клинических задач, способны найти широкое и доступное применение, исходя из возможностей современных систем математической обработки данных. Их представление в универсальной матричной форме, мощные средства диалога, графики и комплексной визуализации делают систему MATLAB инструментом плодотворного использования для коллективного и персонального пользователя в системе его научной работы, производства или образования. При научном подходе к определенной проблеме, естественно, возникает концепция той или иной модели. В представляемом докладе, статистика регистрации измерений точечного распределения биопотенциала на поверхности тела человека, будет обработана и представлена некоторой статической моделью. Или, точнее, рядом однотипных моделей, согласно порядку временных измерений состояний миокарда посредством некоторой системы электродного отведения с поверхности тела. Построение модели обусловлено одной из актуальных проблем современной кардиологии — получение максимально полной информации об электрическом потенциале сердца, на основании которой можно было бы расширить диагностику патологических состояний миокарда, его электрофизиологических свойств.

Идея построения модели электрического поля по поверхности объекта вовсе не является упрощением, принятых до сих пор, попыток определения его генезиса. Объективно характеристики поля, представленные стандартным отведением электрокардиограммы, служат в медицине для диагностики качества функций сердца по результату его электрофизиологической деятельности. Все методы диагностики, по-прежнему, являются результатом эмпирического опыта исследования текущего состояния объекта. Естественно, по мере накопления и развития математических методов прогноза, оценок, роста вычислительной производительности и адаптивной возможности техники, возникает идея прогнозирования будущего

состояния реальной кардиосистемы на основании ряда однотипных статических моделей, построенных по кадрам-измерениям электродного отведения с некоторым временным дискретным шагом друг от друга. Собственно, материал данного доклада и представляет реализацию подобной статической модели средствами MATLAB, как системы, предоставляющей пользователям мощный язык программирования, ориентированный на математические расчеты, реализацию численных методов.

Предлагаемая математическая модель, по существу, является базой для создания диагностических алгоритмов в кардиологии в форме стохастической линейной нестационарной модели на основе базисных сферических функций [1]. Структура модели описывается следующим образом:

$$\varphi_i(p) = \sum_n C_n(p_i) a_n(i) + \xi_i, \quad (1)$$

где $a_n(i)$ — коэффициенты, зависящие от времени, $\{\xi_i\}$ — последовательность независимых случайных величин с нулевым математическим ожиданием и ограниченной дисперсией; φ_i — пространственный потенциал в момент времени i ; p_i — сферические координаты; $C_n(p_i)$ — сферические функции.

Для решения задачи оценивания параметров $a_n(i)$ в каждый момент времени определяется пространственная модель, т. е. идентифицируется некоторый вектор коэффициентов $a_n(i)$. В виду линейности модели относительно оцениваемых параметров, эффективно применение стандартного МНК. При этом оценки параметров будут несмещенными, состоятельными, эффективными.

Теперь следует определить численные методы из теории потенциалов и прокомментировать их практическую адаптацию к исполнению в среде MATLAB. Известно, что с помощью оператора Лапласа интегрирование по объему можно заменить интегрированием по поверхности. Рассмотрим частный случай, когда точка P не является ни внутренней ни внешней: она лежит на поверхности тела [2, стр.5]. Будем считать, что радиус малой сферы δ настолько мал, что функция объема V внутри этой сферы — постоянная величина. Гармонической функцией U координат x, y, z соответственно непрерывна вместе со своими первыми и вторыми производными в некоторой области (S) поверхности сферы и удовлетворяет во всех точках этой области уравнению Лапласа $\Delta U = 0$. Тогда шаровой функцией степени n будет гармоническая функция Y_n , являющаяся однородным степенным полиномом и подчиняющаяся уравнению Лапласа. А структура модели (1) является шаровой функцией потенциала на сферических функциях. Тогда, с учетом сферической системы координат этих функций, определяются переменные-координаты для программного алгоритма: λ — долгота, θ — полярное расстояние, ρ — радиус-вектор точки P .

Дифференциальное уравнение для сферической функции U порядка n имеет вид [2, стр.12]:

$$n(n+1)Y_n(\theta, \lambda) + \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial Y_n(\theta, \lambda)}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2 Y_n(\theta, \lambda)}{\partial \lambda^2} = 0. \quad (2)$$

Заменим переменную $\cos \theta$ на t , тогда $dt = -\sin \theta d\theta$, поэтому дифференциальное уравнение (2) можно переписать так:

$$n(n+1)Y_n(t, \lambda) + \frac{\partial}{\partial t} \left[(1-t^2) \frac{\partial Y_n(t, \lambda)}{\partial t} \right] + \frac{1}{1-t^2} \frac{\partial^2 Y_n(t, \lambda)}{\partial \lambda^2} = 0$$

В результате решения этого уравнения в виде:

$$Y_n(t, \lambda) = P(t)Q(\lambda), \quad (3)$$

доказывается [2, стр.13], что второй член уравнения (3) $Q(\lambda)$ есть уравнение гармонических колебаний, зависит и существует при любых λ , а его решение для любых действительных значений m имеет вид

$$Q(\lambda) = A \cos(m\lambda) + B \sin(m\lambda),$$

где A и B - постоянные интегрирования. Тогда как первый член уравнения (3) $P(t)$ зависит от t и постоянных m и n . Обозначив решение через $P_{nm}(t)$, получим:

$$\left(n(n+1) - \frac{m^2}{1-t^2} \right) P_{nm}(t) + \frac{d}{dt} \left((1-t^2) \frac{dP_{nm}(t)}{dt} \right) = 0$$

Функция $P_{nm}(t)$ при целочисленных значениях m носит название присоединенной (ассоциативной) функции Лежандра. В случае $m=0$, эти функции становятся степенными полиномами, которые называются полиномами Лежандра. Полагая в уравнении $m=0$, имеем дифференциальное уравнение для полиномов Лежандра:

$$n(n+1)P_n(t) + \frac{d}{dt} \left((1-t^2) \frac{dP_n(t)}{dt} \right) = 0$$

Присоединенные функции Лежандра и полиномы Лежандра связаны между собой соотношением:

$$P_{nm}(t) = (1-t^2)^{\frac{m}{2}} \frac{d^m P_n(t)}{dt^m}. \quad (4)$$

В итоге, выпишем окончательный вид решения дифференциального уравнения для сферических функций:

$$Y_n(\theta, \lambda) = \sum_{m=0}^n (A_{nm} \cos m\lambda + B_{nm} \sin m\lambda) P_{nm}(\cos \theta), \quad (5)$$

где постоянные n и m называются степенью и порядком сферических функций.

Если учесть свойство ортогональности сферических функций, то это позволит аналитически представить поле электрической активности сердца человека графическими средствами MATLAB. Можно визуализировать поле вариантами картографического рельефа на плоскости и в виде карты на сферической поверхности. В этих случаях сферические функции играют роль приближенного представления некоторой функции потенциала, ко-

торый задается суммой сферических гармоник или рядом Лапласа. Наилучшая средняя квадратическая аппроксимация функции (5), заданной на сфере многочленом, составленным из нормированных сферических гармоник степени n и порядка N , имеет вид [2, стр.14]:

$$\tilde{f}_n(\theta, \lambda) = \sum_{n=0}^N \sum_{m=0}^n (A_{nm} \cos m\lambda + B_{nm} \sin m\lambda) P_{nm}(\cos \theta), \quad (6)$$

где $P_{nm}(\cos \theta)$ — нормированная присоединенная функция Лежандра. Ее связь с полиномом Лежандра выражает (4), где $t = \cos \theta$. Рекуррентная формула позволяет вычислить полином $n+1$ степени, если известны значения полиномов n и $n-1$ степеней:

$$(n+1)P_{n+1}(t) = (2n+1)tP_n(t) - nP_{n-1}(t). \quad (7)$$

Итак, выражение (6) принимаем за аналитическое выражение модели, согласно ее структуре (1). Остается определить некоторые особенности и условия при переходе от аналитической записи модели к программному алгоритму MATLAB. A_{nm} и B_{nm} — неизвестные параметры модели. Полученная модель линейна относительно данных параметров и для их определения эффективен стандартный МНК. Для упрощения демонстрационного алгоритма, при получении аналитического ряда потенциала, выбирается приемлемое практическое значение порядка многочлена, например $N=3$. Также учитывается особенность параметра B , т. е. его возможность принимать значение нуля, когда n или m равны нулю, т. к. для безошибочного решения МНК матричным способом из ряда значений функций следует исключить их нулевые значения. Т. е. вектор функциональных значений при параметрах A и B для расчета МНК будет формироваться не двойным циклом, как это видно из выражения (6), а функциональным рядом при заданном N . Остается найти аналитическое выражение для 4-х членов ряда $P(t)$ (4) из (7) для вычисления значений присоединенной функции (4) ($t = \cos \theta$):

$$P_0(x)=1, P_1(x)=x, P_2(x)=(3x^2-1)/2, P_3(x)=(5x^3-3x)/2$$

Закончить подготовку расчетной части модели для алгоритма программы MATLAB, можно представлением аналитического выражения (6) рядом сферических функций порядка $N=3$:

$$\begin{aligned} \varphi(\theta, \lambda) = & A_{0,0}P_{0,0}\cos(\mu) + A_{1,0}P_{1,0}\cos(\mu) + A_{1,1}\cos\lambda P_{1,1}\cos(\mu) + \\ & + B_{1,1}\sin\lambda P_{1,1}\cos(\mu) + A_{2,0}P_{2,0}\cos(\mu) + A_{2,1}\cos\lambda P_{2,1}\cos(\mu) + B_{2,1}\sin\lambda P_{2,1}\cos(\mu) + \\ & + A_{2,2}\cos 2\lambda P_{2,2}\cos(\mu) + B_{2,2}\sin 2\lambda P_{2,2}\cos(\mu) + A_{3,0}P_{3,0}\cos(\mu) + \\ & + A_{3,1}\cos\lambda P_{3,1}\cos(\mu) + B_{3,1}\sin\lambda P_{3,1}\cos(\mu) + A_{3,2}\cos 2\lambda P_{3,2}\cos(\mu) + \\ & + B_{3,2}\sin 2\lambda P_{3,2}\cos(\mu) + A_{3,3}\cos 3\lambda P_{3,3}\cos(\mu) + B_{3,3}\sin 3\lambda P_{3,3}\cos(\mu) \end{aligned} \quad (8)$$

Очевидно, что модель имеет два входа — широту и долготу, что сразу же позволит перейти к ее картографическому, объемному изображению как альтернативы «плоского» электрокардиографического описания результирующего электрического возбуждения сердца на поверхности тела пациента. Эксперимент получения исходных данных модели проводился

путем снятия потенциалов, с помощью электрокардиографа, в период начального желудочкового комплекса, или комплекса QRS . Комплекс окружен изоэлектрическими отрезками сердечного цикла. Между ними и идет просчет ряда пространственных моделей поля через выбранный период времени 2,5 мл.сек. Традиционными средствами ЭКГ суммарный электрический вектор может проецироваться на фронтальную, горизонтальную, сагиттальную плоскость тела. В стандартных отведениях форма комплекса QRS зависит от электрической оси сердца. А выбранная модель должна показать объективно, в 3-х мерном пространстве, рельеф электрического поля, направление его распространения по миокарду желудочков не зависимо от направления его электрической оси. Для замера потенциалов выбирается пациент без патологий (I-я группа здоровья). Это позволит идентифицировать электрическую ось сердца согласно принятому ее геометрическому положению в пространстве и комментировать ЭКГ как частную составляющую общей пространственной модели.

От тела пациента было сделано 46 отведений по трансверсальным плоскостям с расположением системы координат XYZ в центре трансверсального сечения, расположенного между 4 и 5 межреберьями. Ось OZ совпадает с вертикальной осью тела. Ось OX перпендикулярна фронтальной плоскости. Ось OY , соответственно лежит в фронтальной плоскости и направлена влево от клиента.

Зная центр координат, право-левый диаметр средней трансверсальной плоскости, передне-задний диаметр средней трансверсальной плоскости, высоту грудины - рассчитаем \cos и \sin углов отведений. Далее, посредством обратных тригонометрических функций, определяется в радианах широта и долгота отведений. Путем подачи углов на вход модели (8) решается выход — Z -аппликаты матрицы M потенциалов по всей поверхности XOY :

$$M_{i,j} = \phi(\theta_{i,j}, \lambda_{i,j}),$$

где $\theta_{ij}, \lambda_{ij}$ — соответственно углы широты и долготы с индексами i, j матрицы углов координатной сетки плоскостной карты потенциалов (размерность матрицы $i \times j$). Таким образом, получается рельефное картографическое изображение «развернутого в плоскость» электрического поля. Можно получить изображение карты на сфере. Для чего фигура сферы задается каркасом с определенным шагом по широте и долготе. Потенциалы измерений располагаются между узлами каркаса согласно угловым координатам отведений. Определяем МНК параметры выражения (8) и просчитываем потенциалы модели в точках отведений и в узлах каркаса:

$$X_{i,j} = \phi(\theta_{i,j}, \lambda_{i,j})(\sin(\theta_{i,j}) \cos(\lambda_{i,j}))$$

$$Y_{i,j} = \phi(\theta_{i,j}, \lambda_{i,j})(\sin(\theta_{i,j}) \sin(\lambda_{i,j}))$$

$$Z_{i,j} = \phi(\theta_{i,j}, \lambda_{i,j}) \cos(\theta_{i,j}),$$

Для правильной фиксации на сфере отрицательных потенциалов, можно определить для нее радиус r_0 и его отношением к единице измерения потенциала задать коэффициент масштабного выхода изображения потенциала, подобно как это делается для шкалы осциллографа. Но вместо нулевой оси на плоскости изобразить сферу нулевого потенциала радиуса r_0 . Тогда выражение Z -аппликата примет вид:

$$X_{i,j} = (\phi(\theta_{i,j}, \lambda_{i,j}) + r_0)(\sin(\theta_{i,j}) \cos(\lambda_{i,j}))$$

$$Y_{i,j} = (\phi(\theta_{i,j}, \lambda_{i,j}) + r_0)(\sin(\theta_{i,j}) \sin(\lambda_{i,j}))$$

$$Z_{i,j} = (\phi(\theta_{i,j}, \lambda_{i,j}) + r_0) \cos(\theta_{i,j}),$$

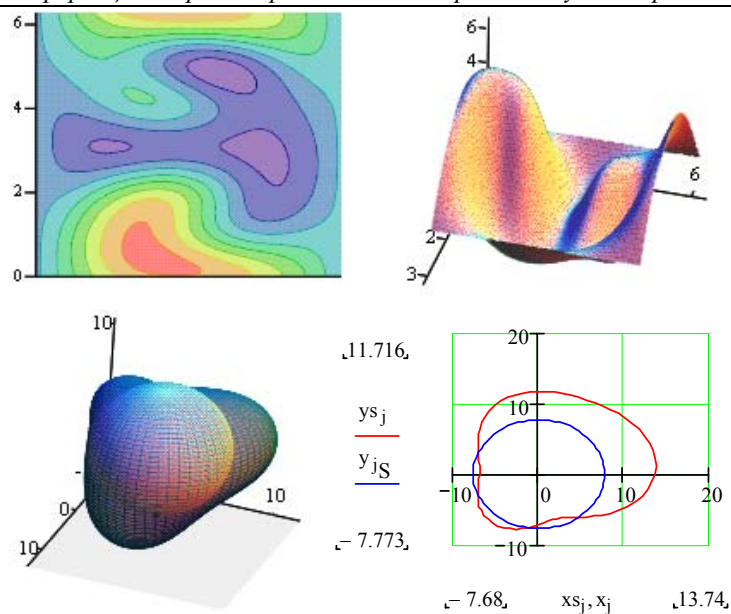
Как частный случай реализации объемной модели можно также получить любую проекцию «среза» сферы на плоскости системы координат, например в плоскости XOY :

$$xs_j = Ku(\phi(\theta, \lambda_j) + r_0)(\sin(\theta) \cos(\lambda_j))$$

$$ys_j = Ku(\phi(\theta, \lambda_j) + r_0)(\sin(\theta) \sin(\lambda_j)),$$

где xs , ys соответственно проекция потенциала на ось абсцисс и ординат. Ku — коэффициент усиления сигнала, θ — задается как константа в интервале $0.. \pi$ для получения «среза» на нужной широте. Это элементарно упрощает получение формы петли вектора сердечного потенциала в любой плоскости в сравнении с традиционным методом ЭКГ. Теперь графически можно отобразить все выше перечисленные способы визуализации уровня биопотенциалов в определенном масштабе и прокомментировать фазы QRS комплекса, отображенные моделью. Из проекций среза демонстрируется горизонтальная плоскость XOY , или, согласно терминам электрокардиографии, — векторкардиограмма во фронтальной плоскости. На изображении объема и плоскости будут присутствовать сфера и окружность правильной формы. Это и есть участки объема и плоскости, где потенциал равен нулю. Потенциал над ними — положителен, под ними, соответственно, отрицателен.

Наиболее ранний период деполяризации (рис. 1) желудочков представлен интервалом 5 мл.сек. от нулевого момента времени начала комплекса ЭКГ и начинается с комплекса QRS. Характеризуется появлением начального экстремума положительного потенциала на передней поверхности грудной клетки, преимущественно в прекардиальной области, и начальным экстремумом отрицательного потенциала на передней боковой поверхности грудной клетки по средне подмышечной линии. Дипольное распределение потенциала совпало по времени с ранним возбуждением левой части межжелудочковой перегородки и внутренних слоев стенки правого желудочка. Его моментный электрический вектор ориентирован слева-направо и несколько вперед. Спустя еще 5–10 мл. сек. минимум потенциала будет быстро смещаться в центр спины, тогда как максимум остается на прежнем месте.



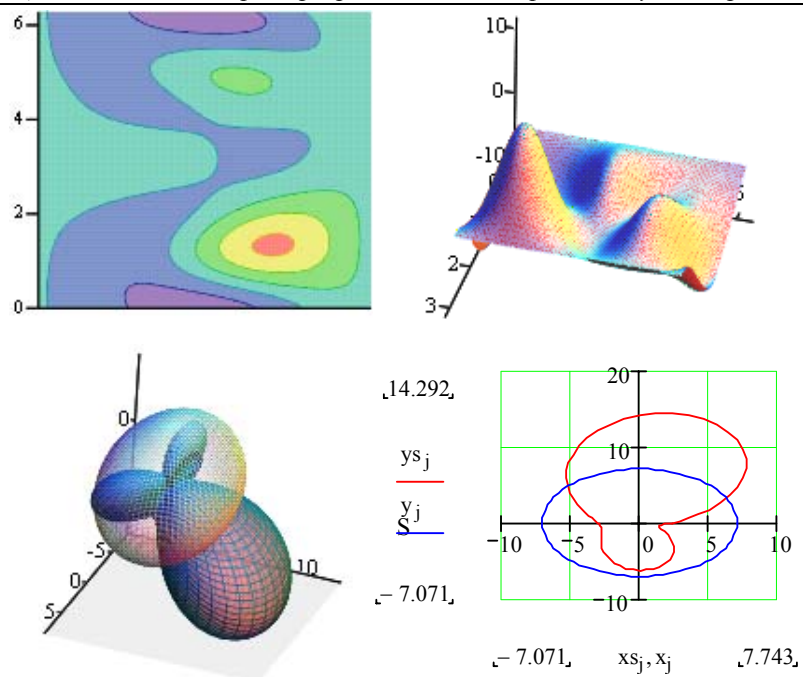
$(X, Y, Z), (x_n, y_n, z_n)$

Рис. 1. Деполяризация 5–10 мл.сек.

За период 27.5 мл. сек. (рис. 2) от начала деполяризации желудочков, выявлен феномен правожелудочкового эпикардального «прорыва». Что соответствует выходу фронта деполяризации на эпикард правого желудочка.

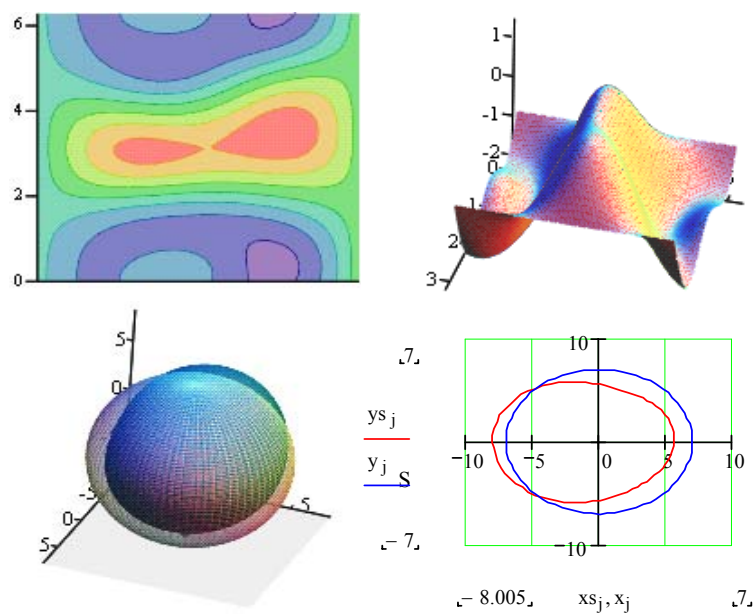
Период 62.5 мл.сек. (рис. 3) — выявляется феномен верхушечного, или левожелудочкового, эпикардального прорыва. Что соответствует выходу фронта деполяризации на эпикард правого желудочка.

В последней фазе желудочковой активизации (рис. 4), к еще не исчезнувшим экстремумам деполяризации, с небольшой амплитудой присоединяются экстремумы реполяризации желудочков. В результате чего образуется «двойное дипольное» или «мозаичное» распределение, которое сохраняется до полного исчезновения потенциалов деполяризации. Это соответствует 80-й мл.сек. длительности комплекса QRS.



$(X, Y, Z), (x_n, y_n, z_n)$

Рис. 2. Деполяризация 27.5 мл.сек.



$(X, Y, Z), (x_n, y_n, z_n)$

Рис. 3. Деполяризация 62.5 мл.сек.

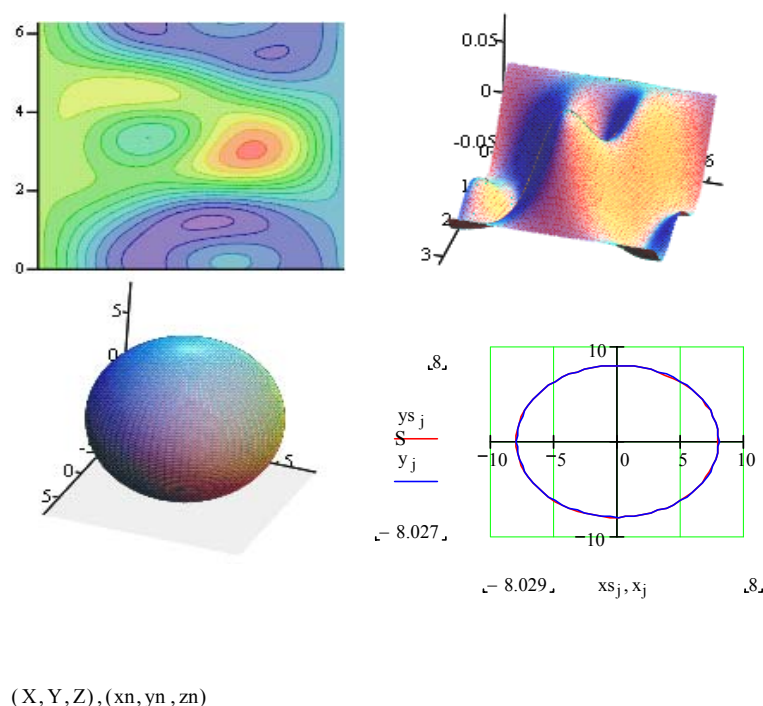


Рис. 4. Деполяризация 80 мл.сек.

На сфере мозаика потенциалов столь незначительная, что поле практически совпало с нулевой фигурой нулевого уровня, а срез с нулевой окружностью. Оценки дисперсии и среднеквадратичного отклонения достигают своего максимального качества для данной модели: $\sigma = (1.997 \times 10^{-4})$, $(\sigma)^{1/2} = (0,014)$.

Таким образом, практический тест интерпретации модели традиционным комплексом ЭКГ, подтверждает вероятное приближение модели к реальному полю биопотенциала сердца на поверхности тела человека. Данный результат можно представить не только как перспективное расширение методов анализа того или иного патологического состояния, но и как способ представления, изучения динамики электрокардиографии в объеме. А это, в свою очередь, позволит понять природу патологических изменений со стороны сердца, не укладывающиеся в известные классические синдромы, по его биоэлектрической активности.

Литература

1. Титомир Л. И., Кнеппо П. Математическое моделирование биоэлектрического генератора сердца.— М.: Наука, 1999.
2. Пантелеев В. Л. Теория фигуры Земли. Курс лекций. Глава 3.— М.: МГУ им. М. В. Ломоносова. Физический факультет, 2000.— С.4–14.

УДК 517.83

АНАЛИЗ ДАННЫХ МНОГОФАКТОРНЫХ ЭКСПЕРИМЕНТОВ С ИСПОЛЬЗОВАНИЕМ КОДОВ ГРЕЯ

Корчагин П. Н.

Пензенский государственный университет, Пенза

e-mail: paul_korch@mail.ru

При испытаниях и аттестации датчиковой аппаратуры широкое применение находят спектры планов эксперимента типа 2^n , оптимальные в широком смысле. При расчете параметров полиномиальных моделей функций отклика по экспериментальным данным [1] используются либо информативные матрицы Адамара, либо Уолша, либо Пэли [2]. В любом из указанных случаев порядок следования строк в информативных матрицах и матрицах спектра плана задается в двоичной системе счисления, если принять значение фактора $x_i = -1$ за 0 и $x_i = 1$ за 1. Здесь следует выделить характерную особенность порядка представления данных, заключающуюся в неравномерной дискретизации n -мерной функции отклика [3, 4]. Например, на рис. 1 показана последовательность представления отсчетов для плана $S = 2^3$ при записи значений уровней факторов x_3, x_2, x_1 в двоичной системе счисления. В случае обработки данных с использованием матрицы Адамара реализуется спектр плана \bar{S} .

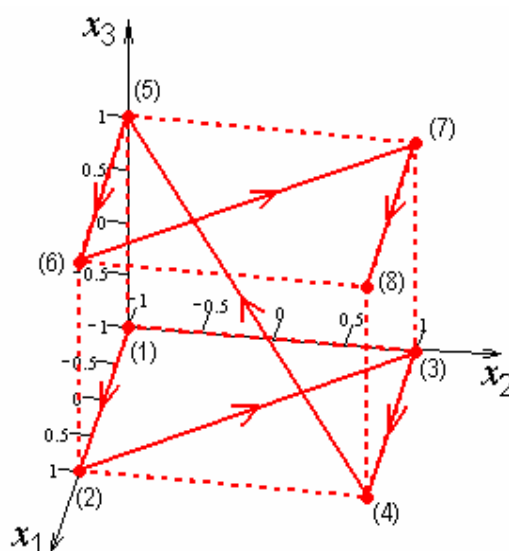


Рис. 1. Последовательность представления отсчетов для плана $S = 2^3$.

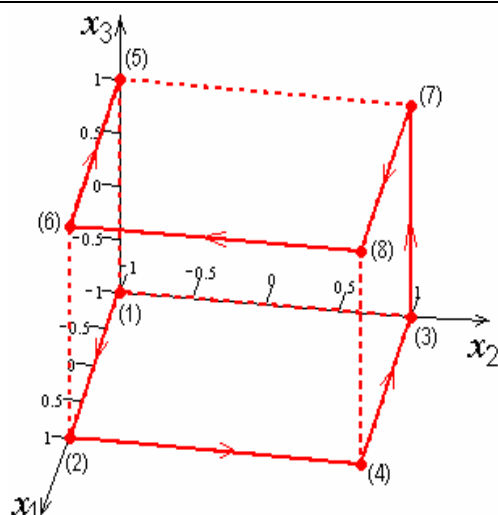


Рис. 2. Последовательность представления отсчетов для плана \bar{S} .

Как следует из рис. 1, расстояния между отсчетами принимают значения из ряда $2, 2\sqrt{2}, 2\sqrt{3}$. Такая неравномерность отсчетов является не существенной для получения значений коэффициентов полиномиальной модели, которые позволяют оценить влияние на функцию отклика как отдельных факторов, так и их совместного воздействия. Но, с точки зрения интерпретации результатов, объяснения физического смысла тех или иных коэффициентов, получаемые модели усложняют сущность изучаемых объектов за счет представления результатов в обобщенном виде.

Для упрощения процедур интерпретации результатов предлагается в качестве наблюдаемых параметров использовать первые конечные разности от экспериментальных значений функции отклика. Корректное определение сравниваемых конечных разностей в n -мерной области, очевидно, достигается при равном расстоянии между отсчетами, или равномерной дискретизации. Требуемая равномерная дискретизация достигается на исходном массиве данных путем их определенного упорядочивания. Опираясь на принятую форму представления точек спектра плана в виде двоичных кодов, назовем соответствующие спектры, по ассоциации с кодами Грея, планами Грея [3, 4].

Для пояснения рассмотрим табл. 1, где в поле А приведены номера отсчетов в спектре плана S (поле В). При упорядочении по плану Грея-321, где цифры указывают порядок следования индексов факторов (см. поле D), последовательность отсчетов выстраивается в ряд NG-321, показанный в поле С.

Таблица 1.

NS	S			NG-321	G-321			NG-123	NG-312
	x_3	x_2	x_1		x_3	x_2	x_1		
A	B			C	D			E	F
1	0	0	0	1	0	0	0	1	1
2	0	0	1	2	0	0	1	5	3
3	0	1	0	4	0	1	1	7	4
4	0	1	1	3	0	1	0	3	2
5	1	0	0	7	1	1	0	4	6
6	1	0	1	8	1	1	1	8	8
7	1	1	0	6	1	0	1	6	7
8	1	1	1	5	1	0	0	2	5

В данном случае последовательность равномерных отсчетов иллюстрируется на рис.2., где расстояния между отсчетами равны 2.

Рассмотрим трехфакторную полиномиальную модель:

$$y(x_1, x_2, x_3) = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_1x_2 + a_5x_1x_3 + a_6x_2x_3 + a_7x_1x_2x_3. \quad (1)$$

Расчет конечных разностей для такой модели при изменении x_1 производится по формуле:

$$\Delta y_{x_1} = 2(a_1 + a_4x_2 + a_5x_3 + a_7x_2x_3). \quad (2)$$

Формулы для получения конечных разностей при изменении x_2 и x_3 запишутся в виде:

$$\Delta y_{x_2} = 2(a_2 + a_4x_1 + a_6x_3 + a_7x_1x_3), \quad (3)$$

$$\Delta y_{x_3} = 2(a_3 + a_5x_1 + a_6x_2 + a_7x_1x_2). \quad (4)$$

Из (3) и (4) следует, что для разностей Δy_{x_2} и Δy_{x_3} логично использовать планы G-312 и G-123, представленные на рис. 3 и 4 соответственно.

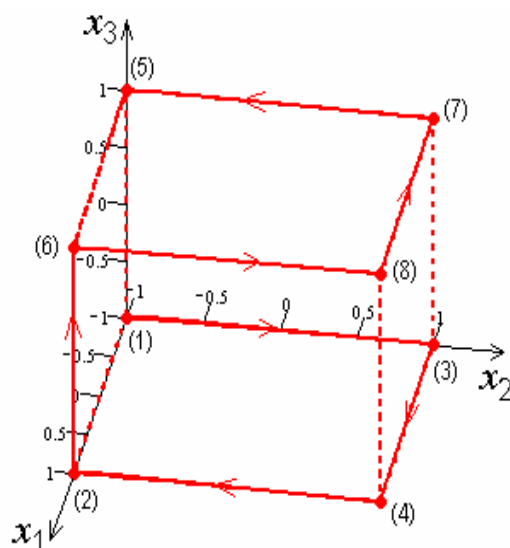


Рис. 3. Последовательность представления отсчетов для плана G-312.

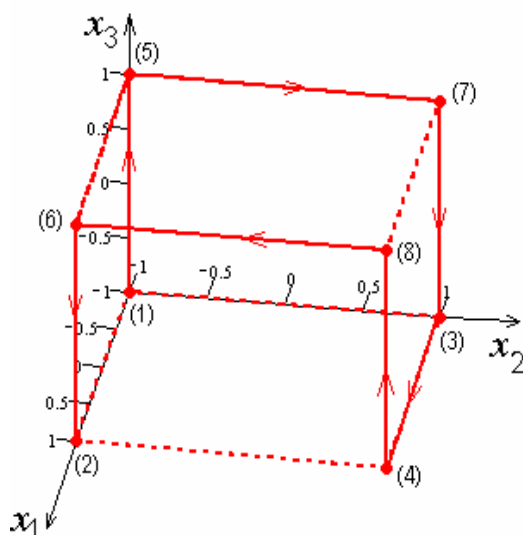


Рис. 4. Последовательность представления отсчетов для плана G-123.

Результаты анализа представляются в виде матрицы, в первом столбце которой представлены номера факторов, а во втором - максимальные значения конечных разностей, наблюдающиеся при изменении этих факторов.

Таким образом, производится факторный анализ (ФА) данных и исследователь получает информацию о том, какой фактор оказывает наибольшее влияние на функцию отклика.

Для проверки работы программы в качестве исходных данных были использованы спектр плана $S = 2^3$ и результаты математического моделирования эксперимента.

Коэффициенты модели (1) задавались случайным образом:

$$a = \begin{pmatrix} 2,898 \\ 2,237 \\ 3,974 \\ 0,892 \\ -1,87 \\ 2,721 \\ -0,993 \\ 1,417 \end{pmatrix}.$$

Результаты анализа:

residuals =

5.6060	1.0000	16.5080	2.0000	1.1620	3.0000
6.3780	3.0000	8.4780	3.0000	6.8680	2.0000
10.8220	1.0000	6.8680	2.0000	8.4780	3.0000
6.8680	2.0000	10.8220	1.0000	7.5420	1.0000
9.0100	1.0000	5.0560	2.0000	8.0740	3.0000

8.0740	3.0000	8.0740	3.0000	5.0560	2.0000
7.5420	1.0000	3.3600	2.0000	6.3780	3.0000

```
fact_result =
3.0000  8.4780
1.0000 10.8220
2.0000 16.5080
```

Видно, что максимальная конечная разность соответствует фактору №2, а минимальная фактору №3, что соответствует коэффициентам a_3 и a_4 .

Литература

1. Ткачев С. В., Михотин В. Д. Планирование эксперимента для испытания датчиковой аппаратуры на метрологическую надежность.— Пенза: Изд-во Пензенского гос. тех. ун-та, 1996.— 184 с.
2. Трахтман А. М., Трахтман В. А. Основы теории дискретных сигналов на конечных интервалах.— М.: Сов. Радио, 1975.— 208 с.
3. Михотин В. Д., Сазонов В. В., Корчагин П. Н. Упрощенный факторный анализ по планам 2^n . // Сб. докл. междунар. конф. «Методы и средства измерения в системах контроля и управления».— Пенза: Пенз. гос. ун-т, 2002.— С.160–163.
4. Михотин В. Д., Сазонов В. В., Корчагин П. Н. Применение кода Грея в факторном анализе // Сб. докл. междунар. школы-семинара «Синтез и сложность управляющих систем».— М.: МГУ, 2002.— Ч.II.— С.163–165.