

УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Методические указания к выполнению лабораторных работ по курсу
«Уравнения математической физики»
для студентов 3 курса ФПМИ
специальности 010500 дневного отделения

Составители:

к.т.н., доцент М.Г. Персова
д.т.н., проф. Ю.Г.Соловейчик
д.т.н., проф. Г.М.Тригубович
к.т.н., доцент А.Г. Задорожный

Рецензент:

к.т.н., доцент М.Г. Токарева

Работа подготовлена на кафедре прикладной математики НГТУ

ЛАБОРАТОРНАЯ РАБОТА № 1

РЕШЕНИЕ ЭЛЛИПТИЧЕСКИХ КРАЕВЫХ ЗАДАЧ МЕТОДОМ КОНЕЧНЫХ РАЗНОСТЕЙ

Цель работы

Разработать программу решения эллиптической краевой задачи методом конечных разностей. Протестировать программу и численно оценить порядок аппроксимации.

Теоретическая часть

Метод конечных разностей [1] основан на разложении функции нескольких независимых переменных в окрестности заданной точки в ряд Тейлора:

$$\begin{aligned} u(\mathbf{x}_1 + \mathbf{h}_1, \dots, \mathbf{x}_n + \mathbf{h}_n) = & u(\mathbf{x}_1, \dots, \mathbf{x}_n) + \sum_{j=1}^n \mathbf{h}_j \frac{\partial}{\partial \mathbf{x}_j} u(\mathbf{x}_1, \dots, \mathbf{x}_n) + \\ & \frac{1}{2} \left(\sum_{j=1}^n \mathbf{h}_j \frac{\partial}{\partial \mathbf{x}_j} \right)^2 u(\mathbf{x}_1, \dots, \mathbf{x}_n) + \dots + \frac{1}{k!} \left(\sum_{j=1}^n \mathbf{h}_j \frac{\partial}{\partial \mathbf{x}_j} \right)^k u(\mathbf{x}_1, \dots, \mathbf{x}_n) +, \quad (1.1) \\ & + \frac{1}{(k+1)!} \left(\sum_{j=1}^n \mathbf{h}_j \frac{\partial}{\partial \mathbf{x}_j} \right)^{k+1} u(\xi_1, \dots, \xi_n) \end{aligned}$$

где \mathbf{h}_j - произвольные приращения соответствующих аргументов, $\xi_j \in [\mathbf{x}_j, \mathbf{x}_j + \mathbf{h}_j]$, функция $u(\mathbf{x}_1, \dots, \mathbf{x}_n)$ обладает ограниченными производными до $(k+1)$ -го порядка включительно.

При использовании двух слагаемых при разложении функции в ряд Тейлора (1.1) производные первого порядка могут быть аппроксимированы следующими конечными разностями первого порядка:

$$\nabla_h^+ u_i = \frac{u_{i+1} - u_i}{h_i}, \quad (1.2)$$

$$\nabla_h^- u_i = \frac{u_i - u_{i-1}}{h_{i-1}}, \quad (1.3)$$

$$\bar{\nabla}_h u_i = \frac{u_{i+1} - u_{i-1}}{h_i + h_{i-1}}, \quad (1.4)$$

где $\nabla_h^+ u_i$ - правая разность, $\nabla_h^- u_i$ - левая разность, $\bar{\nabla}_h u_i$ - двусторонняя разность первого порядка, $u_{i-1} = u(\mathbf{x}_{i-1})$, $u_i = u(\mathbf{x}_i)$, $u_{i+1} = u(\mathbf{x}_{i+1})$.

Через конечные разности первого порядка рекуррентно могут быть определены разности второго и более высокого порядка, аппроксимирующие различ-

ные производные. На неравномерной сетке производная второго порядка может быть получена следующим образом:

$$\mathbf{V}_h \mathbf{u}_i = \frac{2\mathbf{u}_{i-1}}{\mathbf{h}_{i-1}(\mathbf{h}_i + \mathbf{h}_{i-1})} - \frac{2\mathbf{u}_i}{\mathbf{h}_{i-1}\mathbf{h}_i} + \frac{2\mathbf{u}_{i+1}}{\mathbf{h}_i(\mathbf{h}_i + \mathbf{h}_{i-1})}, \quad (1.5)$$

с погрешностью порядка $\mathcal{O}(h)$.

Если сетка равномерная, то

$$\mathbf{V}_h \mathbf{u}_i = \frac{\mathbf{u}_{i-1} - 2\mathbf{u}_i + \mathbf{u}_{i+1}}{h^2}, \quad (1.6)$$

и погрешность аппроксимации имеет уже второй порядок, если функция обладает ограниченной производной четвертого порядка [1].

Пусть область Ω двумерная и определена прямоугольная сетка Ω_h как совокупность точек $(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_1), (\mathbf{x}_1, \mathbf{y}_2), \dots, (\mathbf{x}_n, \mathbf{y}_2), (\mathbf{x}_1, \mathbf{y}_m), \dots, (\mathbf{x}_n, \mathbf{y}_m)$. Тогда для двумерного оператора Лапласа

$$\mathbf{V}\mathbf{u} = \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}^2} + \frac{\partial^2 \mathbf{u}}{\partial \mathbf{y}^2}$$

дискретный аналог на неравномерной прямоугольной сетке может быть определен пятиточечным разностным выражением

$$\begin{aligned} \mathbf{V}_h \mathbf{u}_{i,j} = & \frac{2\mathbf{u}_{i-1,j}}{\mathbf{h}_{i-1}^x(\mathbf{h}_i^x + \mathbf{h}_{i-1}^x)} + \frac{2\mathbf{u}_{i,j-1}}{\mathbf{h}_{j-1}^y(\mathbf{h}_j^y + \mathbf{h}_{j-1}^y)} + \frac{2\mathbf{u}_{i+1,j}}{\mathbf{h}_i^x(\mathbf{h}_i^x + \mathbf{h}_{i-1}^x)} + \\ & \frac{2\mathbf{u}_{i,j+1}}{\mathbf{h}_j^y(\mathbf{h}_j^y + \mathbf{h}_{j-1}^y)} - \left(\frac{2}{\mathbf{h}_{i-1}^x \mathbf{h}_i^x} + \frac{2}{\mathbf{h}_{j-1}^y \mathbf{h}_j^y} \right) \mathbf{u}_{i,j} \end{aligned} \quad (1.7)$$

На равномерной сетке пятиточечный разностный оператор Лапласа выглядит следующим образом

$$\mathbf{V}_h \mathbf{u}_{i,j} = \frac{\mathbf{u}_{i-1,j} - 2\mathbf{u}_{i,j} + \mathbf{u}_{i+1,j}}{h_x^2} + \frac{\mathbf{u}_{i,j-1} - 2\mathbf{u}_{i,j} + \mathbf{u}_{i,j+1}}{h_y^2}. \quad (1.8)$$

и имеет второй порядок погрешности.

Учет краевых условий.

Для узлов расположенных на границе \mathbf{S}_1 , на которых заданы краевые условия первого рода, соответствующие разностные уравнения заменяются соотношениями точно передающими краевые условия, т.е. диагональные элементы матрицы, соответствующие этим узлам заменяются на 1, а соответствующий элемент вектора правой части заменяется на значение \mathbf{u}_g функции в этом узле.

Если расчетная область представляет собой прямоугольник со сторонами, параллельными координатным осям, то направление нормали к границе \mathbf{S}_2 и \mathbf{S}_3 , на которых заданы краевые условия второго и третьего рода, совпадает с одной из координатных линий, и тогда методы аппроксимации производной по норма-

ли $\frac{\partial \mathbf{u}}{\partial \mathbf{n}}$ (которая в этом случае будет равна либо $\pm \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$, либо $\pm \frac{\partial \mathbf{u}}{\partial \mathbf{y}}$) сводятся к одномерным (1.2)–(1.4).

Практическая часть.

1. Построить прямоугольную сетку в области в соответствии с заданием. Допускается использовать фиктивные узлы для сохранения регулярной структуры.
2. Выполнить конечноразностную аппроксимацию исходного уравнения в соответствии с заданием. Получить формулы для вычисления компонент матрицы **A** и вектора правой части **b**.
3. Реализовать программу решения двумерной эллиптической задачи методом конечных разностей с учетом следующих требований:
 - язык программирования C++ или Фортран;
 - предусмотреть возможность задания неравномерных сеток по пространству, учет краевых условий в соответствии с заданием;
 - матрицу хранить в диагональном формате, для решения СЛАУ использовать метод блочной релаксации [2, стр. 886], [3].
4. Протестировать разработанные программы на полиномах соответствующей степени.
5. Провести исследования порядка аппроксимации реализованного методов для различных задач с неполиномиальными решениями. Сравнить полученный порядок аппроксимации с теоретическим.

Варианты заданий.

1. Область имеет П-образную форму. Предусмотреть учет первых и вторых краевых условий.
2. Область имеет П-образную форму. Предусмотреть учет первых и третьих краевых условий.
3. Область имеет Т-образную форму. Предусмотреть учет первых и вторых краевых условий.
4. Область имеет Т-образную форму. Предусмотреть учет первых и третьих краевых условий.
5. Область имеет L-образную форму. Предусмотреть учет первых и вторых краевых условий.
6. Область имеет L-образную форму. Предусмотреть учет первых и третьих краевых условий.
7. Область имеет Г-образную форму. Предусмотреть учет первых и вторых краевых условий.
8. Область имеет Г-образную форму. Предусмотреть учет первых и третьих краевых условий.
9. Область имеет Ш-образную форму. Предусмотреть учет первых и вторых краевых условий.
10. Область имеет Ш-образную форму. Предусмотреть учет первых и третьих краевых условий.

Контрольные вопросы и задания.

1. Понятие порядка аппроксимации конечноразностной схемы.
2. Понятие порядка точности конечноразностной схемы.
3. Понятие устойчивости конечноразностной схемы.
4. Учет вторых краевых условий.
5. Учет третьих краевых условий.

ЛАБОРАТОРНАЯ РАБОТА № 2

РЕШЕНИЕ НЕЛИНЕЙНЫХ НАЧАЛЬНО-КРАЕВЫХ ЗАДАЧ

Цель работы

Разработать программу решения нелинейной одномерной краевой задачи методом конечных элементов. Провести сравнение метода простой итерации и метода Ньютона для решения данной задачи.

Теоретическая часть

Определение нелинейной задачи

Нелинейными называют такие дифференциально-краевые задачи, у которых параметры дифференциального уравнения или краевых условий зависят от решения.

В общем виде нелинейное дифференциальное уравнение эллиптического типа можно записать следующим образом:

$$-\operatorname{div}(\lambda(\mathbf{u}) \operatorname{grad} \mathbf{u}) + \gamma(\mathbf{u}) \mathbf{u} = \mathbf{f}(\mathbf{u}), \quad (2.1)$$

т.е. параметры λ , γ и \mathbf{f} являются некоторыми функциями от \mathbf{u} .

Возможна также ситуация, когда один из коэффициентов (чаще всего это коэффициент диффузии λ) зависит от производных решения, например, $\lambda = \lambda(\operatorname{grad} \mathbf{u})$.

Краевая задача может оказаться нелинейной не только в том случае, когда параметры дифференциального уравнения λ , γ и \mathbf{f} зависят от решения \mathbf{u} , но и тогда, когда параметры краевых условий θ , β или \mathbf{u}_β зависят от \mathbf{u} . Так, краевое условие второго рода делает краевую задачу нелинейной, если параметр θ зависит (не линейно) от \mathbf{u} :

$$\lambda \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \theta(\mathbf{u}). \quad (2.2)$$

Краевое условие же третьего рода является нелинейным, если его параметры β или \mathbf{u}_β зависят от \mathbf{u} . В общем случае нелинейное краевое условие третьего рода может быть записано в виде:

$$\lambda \frac{\partial \mathbf{u}}{\partial \mathbf{n}} + \beta(\mathbf{u})(\mathbf{u} - \mathbf{u}_\beta(\mathbf{u})) = 0 \quad (2.3)$$

(при этом, естественно, предполагается, что $\mathbf{u}_\beta(\mathbf{u})$ – не линейная функция \mathbf{u}).

Обратим внимание, что параметры дифференциального уравнения λ , γ и \mathbf{f} и параметры краевых условий θ , β или \mathbf{u}_β могут зависеть не только от решения, но и от пространственных переменных, т.е. $\lambda = \lambda(\mathbf{u}, \mathbf{x})$, $\mathbf{f} = \mathbf{f}(\mathbf{u}, \mathbf{x})$ и т.д.

В результате конечноэлементной аппроксимации нелинейной краевой задачи (2.1), (2.2), (2.3) получается система нелинейных уравнений

$$\mathbf{A}(\mathbf{q})\mathbf{q} = \mathbf{b}(\mathbf{q}), \quad (2.4)$$

у которой компоненты матрицы $\mathbf{A}(\mathbf{q})$ и вектора правой части $\mathbf{b}(\mathbf{q})$ определяются следующим образом:

$$\mathbf{A}_{ij}(\mathbf{q}) = \int_{\Omega} \lambda(\mathbf{u}^h(\mathbf{q})) \text{grad } \psi_j \text{grad } \psi_i d\Omega + \int_{\Omega} \gamma(\mathbf{u}^h(\mathbf{q})) \psi_j \psi_i d\Omega + \int_{S_3} \beta(\mathbf{u}^h(\mathbf{q})) \psi_j \psi_i dS, \quad (2.5)$$

$$\mathbf{b}_i(\mathbf{q}) = \int_{\Omega} \mathbf{f}(\mathbf{u}^h(\mathbf{q})) \psi_i d\Omega + \int_{S_2} \theta(\mathbf{u}^h(\mathbf{q})) \psi_i dS + \int_{S_3} \beta(\mathbf{u}^h(\mathbf{q})) \mathbf{u}_{\beta}(\mathbf{u}^h(\mathbf{q})) \psi_i dS, \quad (2.6)$$

где

$$\mathbf{u}^h(\mathbf{q}) = \sum_j \mathbf{q}_j \psi_j. \quad (2.7)$$

Рассмотрим методы решения этой системы нелинейных уравнений. Подробно решение нелинейных задач описано в [2, стр. 829].

Метод простой итерации

Наиболее простым для решения системы нелинейных конечноэлементных уравнений (2.4) является так называемый метод простой итерации. Он заключается в следующем.

Выбирается некоторое начальное приближение – функция \mathbf{u}^0 (или вектор весов $\mathbf{q}^{(0)}$ разложения \mathbf{u}^0 по базису $\{\psi_i\}$). Это начальное приближение обычно выбирается на основе некоторых априорных данных.

С помощью выбранного начального приближения $\mathbf{q}^{(0)}$ вычисляются коэффициенты конечноэлементной матрицы $\mathbf{A}_j(\mathbf{q}^{(0)})$ и вектора правой части $\mathbf{f}_i(\mathbf{q}^{(0)})$ (см. соотношения (2.5)–(2.7)). При этом параметры дифференциального уравнения и краевых условий, зависящие от решения \mathbf{u} , могут быть представлены на каждом конечном элементе Ω_I в виде линейной комбинации базисных функций. Например, коэффициент диффузии и правая часть могут быть представлены интерполянтами

$$\lambda = \sum_j \lambda_j(\mathbf{u}^h(\mathbf{q}^{(0)})) \psi_j, \quad \mathbf{f} = \sum_j \mathbf{f}_j(\mathbf{u}^h(\mathbf{q}^{(0)})) \psi_j$$

где (в случае использования элементов лагранжева типа) $\lambda_j(\mathbf{u}^h(\mathbf{q}^{(0)}))$ – значение коэффициента $\lambda(\mathbf{u}^h(\mathbf{q}^{(0)}))$ в \mathbf{j} -м узле, $\mathbf{f}_j(\mathbf{u}^h(\mathbf{q}^{(0)}))$ – значение коэффициента $\mathbf{f}(\mathbf{u}^h(\mathbf{q}^{(0)}))$ в \mathbf{j} -м узле, ψ_j – базисная функция, соответствующая \mathbf{j} -му узлу.

Итерационный процесс в методе простой итерации строится следующим образом. На первом шаге решается система линейных алгебраических уравнений $\mathbf{A}(\mathbf{q}^{(0)})\mathbf{q}^{(1)} = \mathbf{b}(\mathbf{q}^{(0)})$. В результате решения этой СЛАУ находится новое приближение $\mathbf{q}^{(1)}$ решения нелинейной системы (2.4), которое, в свою очередь, ис-

пользуется для нахождения компонент матрицы $\mathbf{A}_{ij}(\mathbf{q}^{(1)})$ и вектора $\mathbf{b}_i(\mathbf{q}^{(1)})$, затем решается система $\mathbf{A}(\mathbf{q}^{(1)})\mathbf{q}^{(2)} = \mathbf{b}(\mathbf{q}^{(1)})$ и т.д.

Останов итерационного процесса

Решение $\mathbf{q}^{(k)}$ на k -й итерации можно считать решением системы нелинейных конечноэлементных уравнений (2.4), если выполняется условие:

$$\frac{\|\mathbf{A}(\mathbf{q}^{(k)})\mathbf{q}^{(k)} - \mathbf{b}(\mathbf{q}^{(k)})\|}{\|\mathbf{b}\|} < \varepsilon, \quad (2.8)$$

где ε – некоторое малое число, выбранное как требуемая точность решения нелинейной задачи. В данном случае норма абсолютной невязки нелинейной системы $\mathbf{A}(\mathbf{q}^{(k)})\mathbf{q}^{(k)} - \mathbf{b}(\mathbf{q}^{(k)})$ делится на норму её правой части для того, чтобы в качестве ε можно было бы использовать относительное, а не абсолютное значение порога невязки, что гораздо удобнее при решении большинства практических задач. При этом в качестве вектора \mathbf{b} можно брать как вектор $\mathbf{b}(\mathbf{q}^{(0)})$, вычисленный по начальному приближению, так и вектор $\mathbf{b}(\mathbf{q}^{(k)})$, вычисленный по значению $\mathbf{q}^{(k)}$, полученному на текущей итерации по нелинейностям.

В ряде случаев в качестве критерия выхода можно использовать условие $\|\mathbf{q}^{(k)} - \mathbf{q}^{(k-1)}\| / \|\mathbf{q}^{(k)}\| < \delta$, где δ – некоторое малое число. Это условие означает, что при переходе к следующей итерации решение почти не меняется. Однако следует помнить, что такое условие выхода не гарантирует получение решения нелинейной задачи с требуемой точностью и может использоваться в «аварийных» случаях, когда итерационный процесс по нелинейностям начинает стагнировать. Можно также для остановки итерационного процесса по нелинейностям при его стагнации предусматривать выход по количеству итераций.

Использование релаксации

Для ускорения сходимости процесса решения нелинейной задачи, а во многих случаях и для её обеспечения может быть использован принцип релаксации. В этом случае каждое последующее приближение решения строится как

$$\mathbf{q}^{(k)} = \omega^k \bar{\mathbf{q}}^{(k)} + (1 - \omega^k) \mathbf{q}^{(k-1)}, \quad (2.9)$$

где ω^k – коэффициент релаксации, а $\bar{\mathbf{q}}^{(k)}$ – решение системы

$$\mathbf{A}(\mathbf{q}^{(k-1)})\bar{\mathbf{q}}^{(k)} = \mathbf{b}(\mathbf{q}^{(k-1)}).$$

Метод простой итерации очень прост по программной реализации, но, как правило, требует выполнения большого количества итераций, а в ряде случаев вообще не даёт возможности получить решение нелинейной задачи, особенно в случаях с существенной зависимостью параметров задачи от решения (или его производных). Метод же Ньютона, хотя и является значительно более сложным для про-

граммной реализации, обладает гораздо лучшей сходимостью при решении многих нелинейных задач.

Метод Ньютона

Метод Ньютона основан на линеаризации нелинейных уравнений системы (2.4) с использованием разложения в ряд Тейлора. Суть такой линеаризации заключается в следующем.

Каждый нелинейный член уравнений системы (2.4) представляется в виде разложения в ряд Тейлора в окрестности точки \mathbf{q}^0 (\mathbf{q}^0 – вектор весов, полученный на предыдущей итерации по нелинейностям или являющийся начальным приближением), причём этот ряд ограничивается членами с первыми производными. Так, слагаемые левой части уравнений системы (2.4) представляются в виде

$$\mathbf{A}_{ij}(\mathbf{q})\mathbf{q}_j \approx \mathbf{A}_{ij}(\mathbf{q}^0)\mathbf{q}_j^0 + \sum_r \left. \frac{\partial (\mathbf{A}_{ij}(\mathbf{q})\mathbf{q}_j)}{\partial \mathbf{q}_r} \right|_{\mathbf{q}=\mathbf{q}^0} (\mathbf{q}_r - \mathbf{q}_r^0),$$

а компоненты вектора правой части – в виде

$$\mathbf{b}_i(\mathbf{q}) \approx \mathbf{b}_i(\mathbf{q}^0) + \sum_r \left. \frac{\partial \mathbf{b}_i(\mathbf{q})}{\partial \mathbf{q}_r} \right|_{\mathbf{q}=\mathbf{q}^0} (\mathbf{q}_r - \mathbf{q}_r^0).$$

В результате из системы (2.4) получается система линейных алгебраических уравнений

$$\mathbf{A}^L \mathbf{q} = \mathbf{b}^L, \quad (2.10)$$

которую мы будем называть линеаризованной по методу Ньютона конечноэлементной системой.

Сразу заметим, что при программной реализации линеаризацию уравнений системы (2.4) после сборки глобальной матрицы выполнять чрезвычайно сложно, поскольку процедуру сборки глобальной матрицы конечноэлементной системы уравнений легко запрограммировать тогда, когда компоненты локальных матриц и векторов являются числами, а не некоторыми функциями вектора решения \mathbf{q} . Поэтому линеаризацию отдельных слагаемых уравнений системы (2.4) нужно проводить на уровне локальных матриц и векторов, и только после этого собирать из них глобальную матрицу \mathbf{A}^L и вектор правой части \mathbf{b}^L линеаризованной конечноэлементной системы.

Рассмотрим конечный элемент Ω_m . Обозначим через $\hat{\mathbf{A}}$ локальную матрицу элемента Ω_m , через $\hat{\mathbf{b}}$ – его локальный вектор правой части, а через $\hat{\mathbf{q}}$ – подвектор глобального вектора \mathbf{q} , состоящий из весов (в локальной нумерации) базисных функций конечного элемента Ω_m .

Представим слагаемое $\hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})\hat{\mathbf{q}}_j$ в виде разложения в ряд Тейлора в окрестности значения $\hat{\mathbf{q}}^0$ и ограничимся членами с первой производной:

$$\begin{aligned}
\hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})\hat{\mathbf{q}}_j &\approx \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}^0)\hat{\mathbf{q}}_j^0 + \sum_{r=1}^{\hat{n}} \left. \frac{\partial (\hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})\hat{\mathbf{q}}_j)}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} (\hat{\mathbf{q}}_r - \hat{\mathbf{q}}_r^0) = \\
&= \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}^0)\hat{\mathbf{q}}_j^0 + \sum_{\substack{r=1 \\ r \neq j}}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_j^0 (\hat{\mathbf{q}}_r - \hat{\mathbf{q}}_r^0) + \left\{ \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}^0) + \left. \frac{\partial \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_j^0 \right\} (\hat{\mathbf{q}}_j - \hat{\mathbf{q}}_j^0) = \\
&= \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}^0)\hat{\mathbf{q}}_j + \sum_{r=1}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_j^0 (\hat{\mathbf{q}}_r - \hat{\mathbf{q}}_r^0), \tag{2.11}
\end{aligned}$$

где \hat{n} – размер локальной матрицы (и, естественно, вектора) элемента Ω_m , а $\hat{\mathbf{q}}^0$ – значение $\hat{\mathbf{q}}$ на предыдущей итерации метода Ньютона.

Аналогично линейризуются и компоненты локального вектора правой части:

$$\hat{\mathbf{b}}_i(\hat{\mathbf{q}}) \approx \hat{\mathbf{b}}_i(\hat{\mathbf{q}}^0) + \sum_{r=1}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{b}}_i(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} (\hat{\mathbf{q}}_r - \hat{\mathbf{q}}_r^0). \tag{2.12}$$

Из соотношения (2.11) видно, что в результате линейризации слагаемого $\hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})\hat{\mathbf{q}}_j$ появляются слагаемые, содержащие кроме $\hat{\mathbf{q}}_j$ ещё и другие компоненты вектора решения. Это значит, что в результате линейризации слагаемого $\hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})\hat{\mathbf{q}}_j$ мы получим вклады не только в компоненту $\hat{\mathbf{A}}_{ij}^L$ локальной матрицы $\hat{\mathbf{A}}^L$ линейризованной СЛАУ, но и в другие компоненты i -й строки $\hat{\mathbf{A}}^L$. В свою очередь, из соотношения (2.12) видно, что в результате линейризации компоненты $\hat{\mathbf{b}}_i(\hat{\mathbf{q}})$ вектора правой части появляются слагаемые, содержащие компоненты вектора решения. Это означает, что в результате линейризации компонент вектора правой части системы (2.4) появляются добавки в компоненты матрицы линейризованной конечноэлементной системы. Таким образом, линейризованная система (2.10) собирается из локальных матриц $\hat{\mathbf{A}}^L$ и локальных векторов $\hat{\mathbf{b}}^L$ следующего вида:

$$\hat{\mathbf{A}}_{ij}^L = \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}^0) + \sum_{r=1}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_r^0 - \left. \frac{\partial \hat{\mathbf{b}}_i(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0}, \tag{2.13}$$

$$\hat{\mathbf{b}}_i^L = \hat{\mathbf{b}}_i(\hat{\mathbf{q}}^0) + \sum_{j=1}^{\hat{n}} \hat{\mathbf{q}}_j^0 \sum_{r=1}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_r^0 - \sum_{r=1}^{\hat{n}} \left. \frac{\partial \hat{\mathbf{b}}_i(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_r} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\mathbf{q}}_r^0. \tag{2.14}$$

Итерационный процесс в методе Ньютона строится аналогично тому, как был построен итерационный процесс в методе простой итерации. Очередной его шаг заключается в генерации по \mathbf{q}^0 (на первой итерации \mathbf{q}^0 формируется из начального приближения, на последующих в качестве \mathbf{q}^0 берётся $\mathbf{q}^{(k-1)}$) линеаризованной системы (2.10) и её решении, в результате чего (возможно, с использованием релаксации (2.9)) находится новое приближение $\mathbf{q}^{(k)}$. После этого проверяется критерий останова (2.8), и в случае его невыполнения осуществляется переход к следующей итерации метода Ньютона.

Обратим внимание, что при оценке невязки используются матрица и вектор нелинеаризованной системы. Это значит, что при вычислении невязки должны быть использованы не матрица \mathbf{A}^L и вектор \mathbf{b}^L , собираемые из локальных матриц и векторов вида (2.13) и (2.14), а матрица $\mathbf{A}(\mathbf{q})$ и вектор $\mathbf{b}(\mathbf{q})$, компоненты которых вычисляются, как и в методе простой итерации, по формулам (2.5)–(2.7), и в этих формулах в качестве вектора \mathbf{q} берётся вектор $\mathbf{q}^{(k)}$, полученный в результате решения линеаризованной системы (2.10).

Вычисление производных

Вычисление производных продемонстрируем на примере, когда коэффициент диффузии λ зависит от $\partial \mathbf{u} / \partial \mathbf{x}$. Рассмотрим конечный элемент Ω_m . Введём на нем локальную нумерацию узлов и базисных функций: $\hat{\mathbf{x}}_1 = \mathbf{x}_m$, $\hat{\mathbf{x}}_2 = \mathbf{x}_{m+1}$, $\hat{\psi}_1 = \psi_m$, $\hat{\psi}_2 = \psi_{m+1}$ (ψ_i – глобальные кусочно-линейные базисные функции).

Коэффициент $\lambda(\partial \mathbf{u} / \partial \mathbf{x})$ на элементе Ω_m представим в виде линейного интерполянта

$$\lambda^{\Omega_m}(\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x}) = \lambda^{\Omega_m} \left(\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_1} \right) \hat{\psi}_1(\mathbf{x}) + \lambda^{\Omega_m} \left(\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_2} \right) \hat{\psi}_2(\mathbf{x}),$$

где $\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} = \partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \Big|_{\Omega_m} = \sum_{i=1}^n \hat{\mathbf{q}}_i \partial \hat{\psi}_i(\mathbf{x}) / \partial \mathbf{x}$, и для случая кусочно-линейных базисных функций

$$\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} = \hat{\mathbf{q}}_1 \partial \hat{\psi}_1(\mathbf{x}) / \partial \mathbf{x} + \hat{\mathbf{q}}_2 \partial \hat{\psi}_2(\mathbf{x}) / \partial \mathbf{x}.$$

Тогда компоненты локальной матрицы $\hat{\mathbf{A}}$ конечного элемента Ω_m могут быть вычислены следующим образом:

$$\begin{aligned} \hat{\mathbf{A}}_{ij}(\hat{\mathbf{q}}) &= \int_{\Omega_m} \lambda^{\Omega_m} \left(\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_1} \right) \hat{\psi}_1 \text{grad } \hat{\psi}_j \text{grad } \hat{\psi}_i d\Omega + \\ &+ \int_{\Omega_m} \lambda^{\Omega_m} \left(\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_2} \right) \hat{\psi}_2 \text{grad } \hat{\psi}_j \text{grad } \hat{\psi}_i d\Omega + \int_{\Omega_m} \gamma \hat{\psi}_j \hat{\psi}_i d\Omega. \end{aligned} \quad (2.15)$$

Вычислим производные по $\hat{\mathbf{q}}_j$ компонент локальной матрицы:

$$\begin{aligned}
\left. \frac{\partial \hat{\mathbf{A}}_r(\hat{\mathbf{q}})}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} &= \int_{\Omega_m} \frac{\partial \lambda^{\Omega_m} \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_1} \right)}{\partial \hat{\mathbf{q}}_j} \bigg|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\psi}_1 \operatorname{grad} \hat{\psi}_i \operatorname{grad} \hat{\psi}_r d\Omega + \\
&+ \int_{\Omega_m} \frac{\partial \lambda^{\Omega_m} \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_2} \right)}{\partial \hat{\mathbf{q}}_j} \bigg|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} \hat{\psi}_2 \operatorname{grad} \hat{\psi}_i \operatorname{grad} \hat{\psi}_r d\Omega. \quad (2.16)
\end{aligned}$$

Таким образом, нам необходимо вычислить производные $\left. \frac{\partial \lambda^{\Omega_m} \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right)}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0}$, которые уже не зависят от неизвестных $\hat{\mathbf{q}}_j$ и фактически являются константами на Ω_m , и вычислить интегралы $\int_{\Omega_m} \hat{\psi}_i \operatorname{grad} \hat{\psi}_i \operatorname{grad} \hat{\psi}_r d\Omega$.

Значения производных от коэффициента диффузии можно получить следующим образом:

$$\left. \frac{\partial \lambda^{\Omega_m} \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right)}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} = \frac{\partial \lambda^{\Omega_m}}{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x}} \bigg|_{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \big|_{\mathbf{x}=\mathbf{x}_l}} \cdot \left. \frac{\partial \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right)}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0}, \quad (2.17)$$

где значение производной от производной приближённого решения \mathbf{u}^h по неизвестной $\hat{\mathbf{q}}_j$ вычисляется как

$$\left. \frac{\partial \left(\left. \frac{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right)}{\partial \hat{\mathbf{q}}_j} \right|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} = \frac{\partial \sum_{\mathbf{r}} \hat{\mathbf{q}}_{\mathbf{r}} \left(\left. \frac{\partial \hat{\psi}_{\mathbf{r}}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right)}{\partial \hat{\mathbf{q}}_j} \bigg|_{\hat{\mathbf{q}}=\hat{\mathbf{q}}^0} = \left(\left. \frac{\partial \hat{\psi}_j(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_l} \right). \quad (2.18)$$

Способ же вычисления производной $\frac{\partial \lambda^{\Omega_m}}{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x}} \bigg|_{\partial \mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x}) / \partial \mathbf{x} \big|_{\mathbf{x}=\mathbf{x}_l}}$ зависит от того, каким образом задана зависимость λ от $\partial \mathbf{u} / \partial \mathbf{x}$. Если эта зависимость задана таблично, то обычно строится аппроксимирующий её сплайн (чаще всего кубический сплайн с непрерывными первыми производными), и по нему вычисляется

значение производной $\partial\lambda/(\partial\mathbf{u}/\partial\mathbf{x})$. Этот случай является наиболее общим. В частных же случаях, когда зависимость λ от $\partial\mathbf{u}/\partial\mathbf{x}$ задаётся аналитически (формулой), производную $\partial\lambda/(\partial\mathbf{u}/\partial\mathbf{x})$ можно вычислять аналитически. Например, если зависимость задана в виде $\lambda(\mathbf{u}) = (\partial\mathbf{u}/\partial\mathbf{x})^2 + 1$, то производная будет вычислена следующим образом

$$\left. \frac{\partial\lambda^{\Omega_m}}{\partial\mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})/\partial\mathbf{x}} \right|_{\partial\mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}=\mathbf{x}_I}} = 2 \left(\partial\mathbf{u}^h(\hat{\mathbf{q}}, \mathbf{x})/\partial\mathbf{x} \Big|_{\mathbf{x}=\mathbf{x}_I} \right) = 2 \sum_j \hat{\mathbf{q}}_j^0 \left(\partial\hat{\psi}_j(\mathbf{x})/\partial\mathbf{x} \right) \Big|_{\mathbf{x}=\mathbf{x}_I}$$

Особенности решения нелинейной задачи для уравнения параболического типа

Рассмотрим в общем виде нелинейную начально-краевую задачу для уравнения параболического типа:

$$-\operatorname{div}(\lambda(\mathbf{u}) \operatorname{grad} \mathbf{u}) + \sigma(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(\mathbf{u}), \quad (2.19)$$

с краевыми условиями (2.2), (2.3) и начальным условием $\mathbf{u}|_{t=t_0} = \mathbf{u}_0$ (не путайте с начальным приближением для итераций по нелинейности).

Производную $\frac{\partial \mathbf{u}}{\partial t}$ на каждом временном слое \mathbf{t}_s будем аппроксимировать по двухслойной неявной схеме:

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{t=\mathbf{t}_s} = \frac{\mathbf{u}_s - \mathbf{u}_{s-1}}{\Delta \mathbf{t}_s}$$

Таким образом, решение задачи (2.19) сводится к нахождению решения на каждом временном слое путем решения задачи:

$$-\operatorname{div}(\lambda(\mathbf{u}_s) \operatorname{grad} \mathbf{u}_s) + \frac{\sigma(\mathbf{u}_s)}{\Delta \mathbf{t}_s} \mathbf{u}_s = \mathbf{f}_s(\mathbf{u}_s) + \frac{\sigma(\mathbf{u}_s)}{\Delta \mathbf{t}_s} \mathbf{u}_{s-1}, \quad (2.20)$$

где $\mathbf{u}_s = \mathbf{u}(\mathbf{x}, \mathbf{t}_s)$ – решение на s -м временном слое, $\Delta \mathbf{t}_s = \mathbf{t}_s - \mathbf{t}_{s-1}$ – шаг по времени, $\mathbf{f}_s = \mathbf{f}(\mathbf{x}, \mathbf{t}_s)$ – значение правой части на s -м временном слое.

Также, как и при рассмотрении эллиптической задачи обратим внимание на то, что параметры дифференциального уравнения λ , γ и \mathbf{f} и параметры краевых условий θ , β или \mathbf{u}_β могут зависеть не только от решения, но и от пространственных и временной переменных, т.е. $\lambda = \lambda(\mathbf{u}, \mathbf{x}, \mathbf{t})$, $\mathbf{f} = \mathbf{f}(\mathbf{u}, \mathbf{x}, \mathbf{t})$ и т.д.

В результате конечноэлементной аппроксимации нелинейной начально-краевой задачи (2.20), (2.2), (2.3) получается система нелинейных уравнений

$$\mathbf{A}(\mathbf{q}_s) \mathbf{q}_s = \mathbf{b}(\mathbf{q}_s), \quad (2.21)$$

у которой компоненты матрицы $\mathbf{A}(\mathbf{q}_s)$ и вектора правой части $\mathbf{b}(\mathbf{q}_s)$ определяются следующим образом:

$$\begin{aligned} \mathbf{A}_{ij}(\mathbf{q}_s) = & \int_{\Omega} \lambda_s(\mathbf{u}^h(\mathbf{q}_s)) \text{grad } \psi_j \text{grad } \psi_i d\Omega + \\ & + \frac{1}{\Delta t_s} \int_{\Omega} \sigma_s(\mathbf{u}^h(\mathbf{q}_s)) \psi_j \psi_i d\Omega + \int_{S_3} \beta_s(\mathbf{u}^h(\mathbf{q}_s)) \psi_j \psi_i dS, \end{aligned} \quad (2.22)$$

$$\begin{aligned} \mathbf{b}_i(\mathbf{q}_s) = & \int_{\Omega} \mathbf{f}_s(\mathbf{u}^h(\mathbf{q}_s)) \psi_i d\Omega + \frac{1}{\Delta t_s} \int_{\Omega} \sigma_s(\mathbf{u}^h(\mathbf{q}_s)) \mathbf{u}^h(\mathbf{q}_{s-1}) \psi_i d\Omega + \\ & + \int_{S_2} \theta_s(\mathbf{u}^h(\mathbf{q}_s)) \psi_i dS + \int_{S_3} \beta_s(\mathbf{u}^h(\mathbf{q}_s)) \mathbf{u}_{\beta,s}(\mathbf{u}^h(\mathbf{q}_s)) \psi_i dS, \end{aligned} \quad (2.23)$$

где

$$\mathbf{u}^h(\mathbf{q}_s) = \sum_j \mathbf{q}_{j,s} \psi_j, \quad \mathbf{u}^h(\mathbf{q}_{s-1}) = \sum_j \mathbf{q}_{j,s-1} \psi_j. \quad (2.24)$$

Таким образом, для решения нелинейной задачи (2.20), (2.2), (2.3) выполняется следующая последовательность действий.

Пусть \mathbf{q}^0 вектор весов разложения начального условия \mathbf{u}_0 (т.е. решения \mathbf{u} при $\mathbf{t} = \mathbf{t}^0$). Для каждого временного слоя $\mathbf{s} = 1, 2, \dots, \mathbf{T}$ методом Ньютона или простой итерации решается нелинейная система (2.21)–(2.24), т.е. будет организован итерационный процесс $\mathbf{k} = 1, 2, \dots$, где на каждом шаге будет решаться СЛАУ $\mathbf{A}(\mathbf{p}^{k-1}) \mathbf{p}^k = \mathbf{b}(\mathbf{p}^{k-1})$ (или линеаризованная система $\mathbf{A}^L(\mathbf{p}^{k-1}) \mathbf{p}^k = \mathbf{b}^L(\mathbf{p}^{k-1})$ в случае использования метода Ньютона).

При этом в качестве начального приближения на каждом слое по времени можно выбирать решение с предыдущего слоя, т.е. $\mathbf{p}^0 := \mathbf{q}_{s-1}$. На каждом временном слое по времени выполняются итерации по нелинейности до тех пор пока не выполнится условие $\frac{\|\mathbf{A}(\mathbf{p}^{(k)}) \mathbf{p}^{(k)} - \mathbf{b}(\mathbf{p}^{(k)})\|}{\|\mathbf{b}\|} < \varepsilon$. Если это условие выполняется

итерационный процесс по нелинейности на \mathbf{s} -м временном слое останавливается, значению \mathbf{q}_s присваивается значение \mathbf{p}^k и осуществляется переход к следующему временному слою.

Практическая часть.

1. Выполнить конечноэлементную аппроксимацию исходного уравнения в соответствии с заданием. Получить формулы для вычисления компонент матрицы \mathbf{A} и вектора правой части \mathbf{b} для метода простой итерации.
2. Реализовать программу решения нелинейной задачи методом простой итерации с учетом следующих требований:
 - язык программирования C++ или Фортран;
 - предусмотреть возможность задания неравномерных сеток по пространству и по времени, разрывность параметров уравнения по подобластям, учет краевых условий;

- матрицу хранить в ленточном формате, для решения СЛАУ использовать метод **LU**-разложения;
 - предусмотреть возможность использования параметра релаксации.
3. Выполнить линеаризацию нелинейной системы алгебраических уравнений с использованием метода Ньютона. Получить формулы для вычисления компонент линеаризованных матрицы \mathbf{A}^L и вектора правой части \mathbf{b}^L
 4. Реализовать программу решения нелинейной задачи методом Ньютона.
 5. Протестировать разработанные программы.
 6. Провести исследования реализованных методов на различных зависимостях коэффициента от решения (или производной решения) в соответствии с заданием. На одних и тех же задачах сравнить по количеству итераций метод простой итерации и метод Ньютона. Исследовать скорость сходимости от параметра релаксации.

Варианты заданий.

1. Уравнение: $-\mathbf{div}(\lambda(\mathbf{u}) \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f}$. Базисные функции – линейные.
2. Уравнение: $-\mathbf{div}(\lambda(\mathbf{u}) \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f}$. Базисные функции – квадратичные.
3. Уравнение: $-\mathbf{div}\left(\lambda\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right) \mathbf{grad} \mathbf{u}\right) + \gamma \mathbf{u} = \mathbf{f}$. Базисные функции – линейные.
4. Уравнение: $-\mathbf{div}\left(\lambda\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right) \mathbf{grad} \mathbf{u}\right) + \gamma \mathbf{u} = \mathbf{f}$. Базисные функции – квадратичные.
5. Уравнение: $-\mathbf{div}(\lambda(\mathbf{u}) \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – линейные.
6. Уравнение: $-\mathbf{div}(\lambda(\mathbf{u}) \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – квадратичные.
7. Уравнение: $-\mathbf{div}\left(\lambda\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right) \mathbf{grad} \mathbf{u}\right) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – линейные.
8. Уравнение: $-\mathbf{div}\left(\lambda\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right) \mathbf{grad} \mathbf{u}\right) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – квадратичные.
9. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – линейные.
10. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – квадратичные.

11. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right) \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – линейные.
12. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right) \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}$. Базисные функции – квадратичные.
13. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f}(\mathbf{u})$. Базисные функции – линейные.
14. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f}(\mathbf{u})$. Базисные функции – квадратичные.
15. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)$. Базисные функции – линейные.
16. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \gamma \mathbf{u} = \mathbf{f} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)$. Базисные функции – квадратичные.
17. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(\mathbf{u})$. Базисные функции – линейные.
18. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f}(\mathbf{u})$. Базисные функции – квадратичные.
19. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)$. Базисные функции – линейные.
20. Уравнение: $-\mathbf{div}(\lambda \mathbf{grad} \mathbf{u}) + \sigma \frac{\partial \mathbf{u}}{\partial t} = \mathbf{f} \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \right)$. Базисные функции – квадратичные.

Контрольные вопросы и задания.

1. Какие задачи называют нелинейными?
2. Алгоритм метода простой итерации для решения нелинейного эллиптического уравнения.
3. Условия выхода из итерационного процесса.
4. Параметр релаксации. Возможность его определения на каждой итерации.
5. Алгоритм метода простой итерации для решения нелинейного параболического уравнения.
6. Суть метода Ньютона при решении нелинейной конечноэлементной системы уравнений.
7. Получите вклады в линеаризованную по методу систему, если правая часть дифференциального уравнения зависит от решения.

8. Получите вклады в линеаризованную по методу систему, если коэффициент диффузии дифференциального уравнения зависит от производной решения.
9. Получите вклады в линеаризованную по методу систему, если параметр второго краевого условия дифференциального уравнения зависит от решения.
10. Получите вклады в линеаризованную по методу систему, если параметр σ параболического уравнения зависит от решения.
11. Сколько итераций будет сделано методом Ньютона, если правая часть дифференциального уравнения зависит от решения линейно и почему?
12. Может ли матрица линеаризованной по методу Ньютона системы стать не положительно определенной? Что можно сделать в этом случае?

ЛАБОРАТОРНАЯ РАБОТА № 3

РЕШЕНИЕ ГАРМОНИЧЕСКИХ ЗАДАЧ

Цель работы. Equation Section (Next)

Разработать программу решения гармонической задачи методом конечных элементов. Провести сравнение прямого и итерационного методов решения получаемой в результате конечноэлементной аппроксимации СЛАУ.

Теоретическая часть.

Постановка задачи

Рассмотрим задачу для уравнения

$$\chi \frac{\partial^2 \mathbf{u}}{\partial t^2} + \sigma \frac{\partial \mathbf{u}}{\partial t} - \operatorname{div}(\lambda \operatorname{grad} \mathbf{u}) = \mathbf{f}. \quad (3.1)$$

в котором правая часть \mathbf{f} представима в виде

$$\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) = \mathbf{f}^s(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sin \omega t + \mathbf{f}^c(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cos \omega t. \quad (3.2)$$

Если остальные параметры рассматриваемого уравнения (3.1) не зависят от времени, то тогда и его решение \mathbf{u} может быть представлено в виде

$$\mathbf{u}(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) = \mathbf{u}^s(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sin \omega t + \mathbf{u}^c(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cos \omega t, \quad (3.3)$$

где \mathbf{u}^s и \mathbf{u}^c – две зависящие только от пространственных координат функции, удовлетворяющие системе уравнений

$$\begin{cases} -\operatorname{div}(\lambda \operatorname{grad} \mathbf{u}^s) - \omega \sigma \mathbf{u}^c - \omega^2 \chi \mathbf{u}^s = \mathbf{f}^s, \\ -\operatorname{div}(\lambda \operatorname{grad} \mathbf{u}^c) + \omega \sigma \mathbf{u}^s - \omega^2 \chi \mathbf{u}^c = \mathbf{f}^c. \end{cases} \quad (3.4)$$

То же самое можно сказать и о решении краевой задачи, если не только \mathbf{f} , но и параметры краевых условий являются гармонически изменяющимися по времени функциями с одной и той же частотой ω :

$$\mathbf{u}_g(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) = \mathbf{u}_g^s(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sin \omega t + \mathbf{u}_g^c(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cos \omega t, \quad (3.5)$$

$$\theta(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) = \theta^s(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sin \omega t + \theta^c(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cos \omega t, \quad (3.6)$$

$$\mathbf{u}_\beta(\mathbf{x}, \mathbf{y}, \mathbf{z}, t) = \mathbf{u}_\beta^s(\mathbf{x}, \mathbf{y}, \mathbf{z}) \sin \omega t + \mathbf{u}_\beta^c(\mathbf{x}, \mathbf{y}, \mathbf{z}) \cos \omega t. \quad (3.7)$$

В этом случае функции \mathbf{u}^s и \mathbf{u}^c должны удовлетворять краевым условиям

$$\mathbf{u}^s|_{S_1} = \mathbf{u}_g^s, \quad \mathbf{u}^c|_{S_1} = \mathbf{u}_g^c, \quad (3.8)$$

$$\lambda \frac{\partial \mathbf{u}^s}{\partial \mathbf{n}} \Big|_{S_2} = \theta^s, \quad \lambda \frac{\partial \mathbf{u}^c}{\partial \mathbf{n}} \Big|_{S_2} = \theta^c, \quad (3.9)$$

$$\lambda \frac{\partial \mathbf{u}^s}{\partial \mathbf{n}} \Big|_{S_3} + \beta (\mathbf{u}^s|_{S_3} - \mathbf{u}_\beta^s) = 0, \quad \lambda \frac{\partial \mathbf{u}^c}{\partial \mathbf{n}} \Big|_{S_3} + \beta (\mathbf{u}^c|_{S_3} - \mathbf{u}_\beta^c) = 0. \quad (3.10)$$

Конечноэлементная аппроксимация

Подробно решение гармонических задач описано в [2, стр. 384]. Выполним конечноэлементную аппроксимацию краевой задачи (3.4), (3.8)–(3.10).

Сначала получим эквивалентную вариационную формулировку. Для этого умножим каждое из уравнений системы (3.4) на пробную функцию \mathbf{v} и применим формулу Грина (интегрирования по частям). В результате получим систему двух вариационных уравнений:

$$\left\{ \begin{aligned} & \int_{\Omega} (\lambda \operatorname{grad} \mathbf{u}^s \operatorname{grad} \mathbf{v} - \omega \sigma \mathbf{u}^c \mathbf{v} - \omega^2 \chi \mathbf{u}^s \mathbf{v}) d\Omega + \int_{S_3} \beta \mathbf{u}^s \mathbf{v} dS = \\ & = \int_{\Omega} \mathbf{f}^s \mathbf{v} d\Omega + \int_{S_2} \theta^s \mathbf{v} dS + \int_{S_3} \beta \mathbf{u}_{\beta}^s \mathbf{v} dS, \\ & \int_{\Omega} (\lambda \operatorname{grad} \mathbf{u}^c \operatorname{grad} \mathbf{v} + \omega \sigma \mathbf{u}^s \mathbf{v} - \omega^2 \chi \mathbf{u}^c \mathbf{v}) d\Omega + \int_{S_3} \beta \mathbf{u}^c \mathbf{v} dS = \\ & = \int_{\Omega} \mathbf{f}^c \mathbf{v} d\Omega + \int_{S_2} \theta^c \mathbf{v} dS + \int_{S_3} \beta \mathbf{u}_{\beta}^c \mathbf{v} dS. \end{aligned} \right. \quad (3.11)$$

Уравнения (3.11) должны выполняться для любой $\mathbf{v} \in \mathbf{H}^1$, удовлетворяющей однородному первому краевому условию на границе S_1 .

Построим конечноэлементную аппроксимацию на основе вариационной формулировки (3.11). Пусть $\{\psi_i\}$ – набор базисных функций ($i = 1 \dots n$). Чтобы получить конечноэлементную СЛАУ, заменим в (3.11) каждую из искомым функций \mathbf{u}^s и \mathbf{u}^c на функции $\mathbf{u}^{s,h} = \sum_{i=1}^n \mathbf{q}_i^s \psi_i$ и $\mathbf{u}^{c,h} = \sum_{i=1}^n \mathbf{q}_i^c \psi_i$, а вместо пробной функции подставим поочередно базисные функции ψ_i :

$$\begin{aligned} & \sum_{j=1}^n \left(\int_{\Omega} (\lambda \operatorname{grad} \psi_i \operatorname{grad} \psi_j - \omega^2 \chi \psi_i \psi_j) d\Omega + \int_{S_3} \beta \psi_i \psi_j dS \right) \mathbf{q}_j^s - \omega \sum_{j=1}^n \left(\int_{\Omega} \sigma \psi_i \psi_j d\Omega \right) \mathbf{q}_j^c = \\ & = \int_{\Omega} \mathbf{f}^s \psi_i d\Omega + \int_{S_2} \theta^s \psi_i dS + \int_{S_3} \beta \mathbf{u}_{\beta}^s \psi_i dS, \quad i = 1 \dots n, \end{aligned} \quad (3.12)$$

$$\begin{aligned} & \sum_{j=1}^n \left(\int_{\Omega} (\lambda \operatorname{grad} \psi_i \operatorname{grad} \psi_j - \omega^2 \chi \psi_i \psi_j) d\Omega + \int_{S_3} \beta \psi_i \psi_j dS \right) \mathbf{q}_j^c + \omega \sum_{j=1}^n \left(\int_{\Omega} \sigma \psi_i \psi_j d\Omega \right) \mathbf{q}_j^s = \\ & = \int_{\Omega} \mathbf{f}^c \psi_i d\Omega + \int_{S_2} \theta^c \psi_i dS + \int_{S_3} \beta \mathbf{u}_{\beta}^c \psi_i dS, \quad i = 1 \dots n. \end{aligned} \quad (3.13)$$

В результате мы получили систему из $2n$ уравнений с $2n$ неизвестными \mathbf{q}_j^s и \mathbf{q}_j^c . Чтобы определить матрицу и вектор правой части полученной конечноэлементной СЛАУ, пронумеруем её уравнения и неизвестные следующим образом. Уравнения из (3.12) и (3.13) будем нумеровать поочередно таким образом, что i -

е уравнение из (3.12) будет иметь номер $2i - 1$, а i -е уравнение из (3.13) – номер $2i$. Соответственно пронумеруем и неизвестные этой системы, которые обозначим \mathbf{q}_j , $j = 1 \dots 2n$ (т.е. неизвестные \mathbf{q}_j^s и \mathbf{q}_j^c системы уравнений (3.12) и (3.13) мы объединяем в один вектор \mathbf{q}): $\mathbf{q}_{2j-1} = \mathbf{q}_j^s$, $\mathbf{q}_{2j} = \mathbf{q}_j^c$, $j = 1 \dots n$.

Обозначим

$$\mathbf{p}_{ij} = \int_{\Omega} (\lambda \operatorname{grad} \psi_i \operatorname{grad} \psi_j - \omega^2 \chi \psi_i \psi_j) d\Omega + \int_{S_3} \beta \psi_i \psi_j dS, \quad (3.14)$$

$$\mathbf{c}_{ij} = \omega \int_{\Omega} \sigma \psi_i \psi_j d\Omega. \quad (3.15)$$

Тогда матрица конечноэлементной СЛАУ будет иметь следующую структуру:

$$\mathbf{A} = \begin{pmatrix} \mathbf{p}_{11} & -\mathbf{c}_{11} & \mathbf{p}_{12} & -\mathbf{c}_{12} & & \mathbf{p}_{1n} & -\mathbf{c}_{1n} \\ \mathbf{c}_{11} & \mathbf{p}_{11} & \mathbf{c}_{12} & \mathbf{p}_{12} & & \mathbf{c}_{1n} & \mathbf{p}_{1n} \\ \mathbf{p}_{21} & -\mathbf{c}_{21} & \mathbf{p}_{22} & -\mathbf{c}_{22} & & \mathbf{p}_{2n} & -\mathbf{c}_{2n} \\ \mathbf{c}_{21} & \mathbf{p}_{21} & \mathbf{c}_{22} & \mathbf{p}_{22} & & \mathbf{c}_{2n} & \mathbf{p}_{2n} \\ & & & & \dots & & \\ & & & & & \dots & \\ \mathbf{p}_{n1} & -\mathbf{c}_{n1} & \mathbf{p}_{n2} & -\mathbf{c}_{n2} & & \mathbf{p}_{nn} & -\mathbf{c}_{nn} \\ \mathbf{c}_{n1} & \mathbf{p}_{n1} & \mathbf{c}_{n2} & \mathbf{p}_{n2} & & \mathbf{c}_{nn} & \mathbf{p}_{nn} \end{pmatrix}.$$

Очевидно, что в этой матрице явно выделяются блоки размера 2×2 вида

$$\mathbf{A}^{ij} = \begin{pmatrix} \mathbf{p}_{ij} & -\mathbf{c}_{ij} \\ \mathbf{c}_{ij} & \mathbf{p}_{ij} \end{pmatrix},$$

для хранения которых достаточно только двух ячеек памяти. Таким образом, для хранения всей матрицы в блочном формате с учетом структуры блока \mathbf{A}^{ij} требуется практически в два раза меньше памяти, чем при хранении её покомпонентно. Поэтому при программной реализации в таких случаях, как правило, используются блочные форматы хранения данных, в том числе и с учетом разреженной структуры матрицы.

Практическая часть.

1. Выполнить конечноэлементную аппроксимацию исходного уравнения в соответствии с заданием. Получить формулы для вычисления компонент матрицы \mathbf{A} и вектора правой части \mathbf{b} .
2. Реализовать программу решения гармонической задачи с учетом следующих требований:
 - язык программирования C++ или Фортран;

- предусмотреть возможность задания неравномерной сетки по пространству, разрывность параметров уравнения по подобластям, учет краевых условий;
 - матрицу хранить в разреженном строчном формате с возможностью регенерации ее в профильный формат;
 - реализовать (или воспользоваться реализованными в курсе «Численные методы») методы решения СЛАУ: итерационный – локально-оптимальную схему или метод сопряженных градиентов для несимметричных матриц с предобуславливанием и прямой – **LU**-разложение или его модификации [2, стр. 871], [3].
3. Протестировать разработанную программу на полиномах первой степени.
 4. Провести исследования реализованных методов для сеток с небольшим количеством узлов 500 – 1000 и большим количеством узлов – порядка 20000 – 50000 для различных значений параметров $10^{-4} \leq \omega \leq 10^9$, $10^2 \leq \lambda \leq 8 \cdot 10^5$, $0 \leq \sigma \leq 10^8$, $8.81 \cdot 10^{-12} \leq \chi \leq 10^{-10}$. Для всех решенных задач сравнить вычислительные затраты, требуемые для решения СЛАУ итерационным и прямым методом.

Варианты заданий.

1. Решить одномерную гармоническую задачу в декартовых координатах, базисные функции – линейные.
2. Решить одномерную гармоническую задачу в цилиндрических координатах, базисные функции – линейные.
3. Решить одномерную гармоническую задачу в цилиндрических координатах с оператором **rotrot**, базисные функции – линейные.
4. Решить одномерную гармоническую задачу в сферических координатах, базисные функции – линейные.
5. Решить двумерную гармоническую задачу в декартовых координатах, базисные функции – билинейные.
6. Решить двумерную гармоническую задачу в цилиндрических координатах, базисные функции – билинейные.
7. Решить двумерную гармоническую задачу в цилиндрических координатах с оператором **rotrot**, базисные функции – билинейные.
8. Решить двумерную гармоническую задачу в полярных координатах, базисные функции – билинейные.
9. Решить трехмерную гармоническую задачу в декартовых координатах, базисные функции – трилинейные.

Контрольные вопросы и задания.

1. Понятие гармонической задачи.
2. Структура получаемой матрицы и вектора правой части конечноэлементной СЛАУ.
3. Условия, при которых задачу можно решать как гармоническую.
4. Возможность решения гармонической задачи как нестационарной.
5. Сравнение решений гармонической и нестационарной задачи.

6. Выдача решения гармонической задачи в произвольный момент времени в произвольной точке.
7. Выдача производной решения гармонической задачи в произвольный момент времени в произвольной точке.
8. Учет краевых условий.
9. Сравнение методов решения СЛАУ, получаемых в результате конечноэлементной аппроксимации гармонической задачи, по вычислительным затратам и возможности получения решения.
10. Блочный формат хранения матриц, получаемых в результате конечноэлементной аппроксимации гармонической задачи.

ЛАБОРАТОРНАЯ РАБОТА № 4

РЕШЕНИЕ НЕСИММЕТРИЧНЫХ СЛАУ

Цель работы. Equation Section 3

Изучить особенности реализации итерационных методов BCG, BCGStab, GMRES для СЛАУ с несимметричными разреженными матрицами. Исследовать влияние предобуславливания на сходимость этих методов.

Теоретическая часть.

Метод GMRES (аббревиатура от **G**eneralized **M**inimum **R**esidual method) и метод бисопряженных градиентов (BCG) описаны в [2, стр. 882-886].

Одна из модификаций метода бисопряженных градиентов с **LU**-предобуславливанием выглядит следующим образом.

Метод бисопряженных градиентов стабилизированный (BCG STAB)

Выбирается начальное приближение \mathbf{x}^0 и полагается:

$$\mathbf{z}^0 = \mathbf{r}^0$$

Далее для $k = 1, 2, \dots$ проводятся следующие вычисления:

$$\begin{aligned}\alpha^k &= \frac{(\mathbf{r}^{k-1}, \mathbf{r}^0)}{(\mathbf{r}^0, \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{z}^{k-1})} \\ \mathbf{p}^k &= \mathbf{r}^{k-1} - \alpha^k \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{z}^{k-1} \\ \gamma^k &= \frac{(\mathbf{p}^k, \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{p}^k)}{(\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{p}^k, \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{p}^k)} \\ \mathbf{y}^k &= \mathbf{y}^{k-1} + \alpha^k \mathbf{z}^{k-1} + \gamma^k \mathbf{t}^k \\ \mathbf{r}^k &= \mathbf{p}^k - \gamma^k \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{p}^k \\ \beta^k &= \frac{\alpha^k (\mathbf{r}^k, \mathbf{r}^0)}{\gamma^k (\mathbf{r}^{k-1}, \mathbf{r}^0)} \\ \mathbf{z}^k &= \mathbf{r}^k + \beta^k \mathbf{r}^{k-1} - \beta^k \gamma^k \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{z}^{k-1}\end{aligned}$$

Практическая часть.

1. Реализовать программу решения СЛАУ большой размерности в разреженном строчном формате в соответствии с заданием.
2. Протестировать разработанную программу на небольших матрицах.
3. Сравнить реализованный метод по вычислительным затратам с методами, используемыми в предыдущей лабораторной работе, на матрицах большой размерности, полученных в результате конечноэлементной аппроксимации в предыдущей лабораторной работе.

Варианты заданий.

1. Реализовать решение СЛАУ методом GMRES без предобуславливания.

2. Реализовать решение СЛАУ методом GMRES с диагональным предобуславливанием.
3. Реализовать решение СЛАУ методом GMRES с **LU**-предобуславливанием.
4. Реализовать решение СЛАУ методом BSG без предобуславливания.
5. Реализовать решение СЛАУ методом BSG с диагональным предобуславливанием.
6. Реализовать решение СЛАУ методом BSG с **LU**-предобуславливанием.
7. Реализовать решение СЛАУ методом BSGSTAB без предобуславливания.
8. Реализовать решение СЛАУ методом BSGSTAB с диагональным предобуславливанием.
9. Реализовать решение СЛАУ методом BSGSTAB с **LU**-предобуславливанием.

Контрольные вопросы и задания.

1. Понятие подпространства Крылова. Базис подпространства Крылова.
2. Ортогонализация Арнольди.
3. Может ли возрасть невязка в методе GMRES? В методе бисопряженных градиентов?
4. Влияние параметра ***m*** на процесс сходимости GMRES. Влияние параметра ***m*** на время решения. Выбор оптимального значения ***m***.

Литература

1. В.П. Ильин. Методы конечных разностей и конечных объемов для эллиптических уравнений // Новосибирск: Изд-во Ин-та математики, 2000. – 345.
2. Соловейчик Ю.Г., Рояк М.Э., Персова М.Г. Метод конечных элементов для решения скалярных и векторных задач // Учебное пособие. – Новосибирск: НГТУ, 2007. – 896 с.
3. М.Г. Персова, М.Э. Рояк, Ю.Г. Соловейчик, А.В. Чернышев. Численные методы решения систем уравнений// Методические указания к выполнению лабораторных работ (№ 2735). – Новосибирск: НГТУ, 2004. – 27 с.