

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАТИКИ

Кафедра вычислительной математики

Касияник Алексей Леонидович

УСКОРЕНИЕ СХОДИМОСТИ ПРОЦЕССОВ
УСТАНОВЛЕНИЯ. ПЕРЕОБУСЛАВЛИВАНИЕ И
ПОДАВЛЕНИЕ КОМПОНЕНТ

Дипломная работа

Научный руководитель:
Фалейчик Борис Викторович
кандидат физ.-мат. наук,
доцент кафедры выч. мат.

Допущена к защите

«___» _____ 2015 г.

Зав. кафедры вычислительной математики
кандидат физико-математических наук, доцент
Мандрик Павел Алексеевич

Минск, 2015

Содержание

Введение	2
1 Жесткие задачи	4
1.1 Явление жесткости	4
1.2 Трудности, возникающие при численном решении	5
1.3 Примеры жестких задач	7
2 Методы установления	8
2.1 Линейная задача	8
2.2 Нелинейная задача	10
2.3 Спектральные свойства и скорость сходимости итерационного процесса	11
3 Ускорение сходимости	13
3.1 Переобусловливание	13
3.1.1 Первый способ	13
3.1.2 Второй способ	13
3.2 Подавление компонент	13
4 Численный эксперимент	16
4.1 Тестовая задача	16
4.2 Результаты численного эксперимента	16

Введение

Жесткие задачи исследуются примерно со второй половины 20 века. Однако и сейчас сформулировать точное определение жесткости проблематично. Наиболее прагматическая точка зрения вместе с тем была и исторически наиболее ранней (Кертисс и Хиршфельдер, 1952 год): *жесткие уравнения — это уравнения, для которых определенные неявные методы дают лучший результат, обычно несравненно более хороший, чем явные методы*. При этом определенную роль играют собственные значения матрицы Якоби, но важны и такие параметры, как размерность системы, гладкость решения или интервал интегрирования. Более полным является определение данное Ламбертом: *если численный метод с ограниченной областью абсолютной устойчивости, примененный к системе с произвольными начальными условиями вынужден использовать на некотором интервале интегрирования величину шага, которая чрезмерно мала по отношению к гладкости точного решения на этом интервале, тогда говорят, что система является жесткой на этом интервале*.

Как известно, наиболее трудоёмким этапом численного интегрирования жёсткой системы (не)линейных обыкновенных дифференциальных уравнений (ОДУ) размерности n неявным методом является решение на каждом шаге системы (не)линейных уравнений, размерность которой пропорциональна n . В таком случае использование методов ньютоновского типа практически невозможно, а традиционные методы типа простой итерации либо не сходятся, либо сходятся очень медленно. В данной работе рассматриваются способы ускорения методов, основанных на процессах установления, которые применимы в указанной выше ситуации. В работе исследованы два эффективных способа ускорения сходимости решения в процессе решения методами установления: переобусловливание и подавление компонент. Переобусловливание является классическим способом уменьшения «спектрального числа обусловленности» матрицы решаемой задачи, которое существенным образом определяет свойства сходимости итерационного процесса. С помощью операции переобусловливания производится уменьшение числа обусловленности, что положительно влияет на скорость сходимости процесса.

Также было замечено, что компоненты ошибки, которые соответствуют малым собственным значениям, сходятся медленно. Поэтому предположительно подавление ошибки медленно сходящийся компонент может дать существенную прибавку в скорости сходимости итерационного процесса. В результате исследования данного феномена был предложен прием ускорения итерационного процесса, который по аналогу со схожими алгоритмами из известных источников именуется «подавлением компонент». Исследованию описанных проблем, разработке вычислительного алгоритмов, которые бы основывались на идее установления и при этом превосходили в скорости из-

вестные алгоритмы и посвящается данная работа.

Глава 1

Жесткие задачи

Тема численного решения однородных дифференциальных уравнений не нова, и, возможно, несколько удивителен тот факт, что методы, разработанные еще в начале 20 века, до сих пор являются основой наиболее эффективных и распространенных подходов при решении ОДУ. За прошедшее время были достигнуты значительные продвижения в надежности и эффективности этих методов, и большинство существующих типичных научных задач могут быть решены достаточно легко и быстро. Тем не менее есть некоторый класс задач, с которыми классические методы справиться не могут. Такие задачи, называемые «жесткими», слишком важны, чтобы их игнорировать, и слишком трудны, чтобы их решить. Они слишком важны, чтобы их игнорировать, так как они возникают при решении важных физических задач. Они слишком затратные при решении, так как из-за присущей им большой размерности и сложности, классические методы становятся слабо применимы даже несмотря на многократное увеличение мощности современной вычислительной техники. Классические методы решения требуют так много шагов, что ошибки округления могут сделать полученное решение далеким от приемлемого[2]. В этом разделе мы и рассмотрим понятие «жесткости», а также те ключевые проблемы, которые возникают при их решении.

1.1 Явление жесткости

Задачи, называемые жесткими, весьма разнообразны, и дать математически строгое определение жесткости непросто. Поэтому в литературе можно встретить различные определения жесткости, отличающиеся степенью строгости. Сущность же явления жесткости состоит в том, что решение, которое необходимо вычислить, меняется медленно, однако в любой его окрестности существуют быстро затухающие возмущения. Характерное время затухания их называют пограничным слоем. Наличие таких возмущений затрудняет получение медленно меняющегося решения численным способом. При этом жесткими могут быть как скалярные дифференциальные уравнения, так и, что встречается особенно часто, системы обыкновенных дифференциальных уравнений.

Определение 1.1. Система обыкновенных дифференциальных уравнений вида

$$u'(t) = Au(t) \tag{1.1}$$

с постоянной $(n \times n)$ -матрицей A называется жесткой, если:

1. $Re\lambda_k < 0, k = \overline{1, n}$ (т.е. задача устойчива);
2. Отношение $S = \frac{\max_{1 \leq k \leq n} |Re\lambda_k|}{\min_{1 \leq k \leq n} |Re\lambda_k|}$ велико (например, $S > 10$);
3. Промежуток интегрирования велик по сравнению с длиной погранслоя.[Repn]

Число S иногда называют *коэффициентом жесткости* системы.

Поскольку система нелинейных обыкновенных дифференциальных уравнений вида $u'(t) = f(t, u(t))$ может быть в окрестности некоторого известного решения $v(t)$ заменена линейной системой

$$u'(t) = f_u(t, v + \theta(u - v))u + b(t),$$

где f_u - матрица Якоби, а $b(t) = f(t, v) - f_u(t, v + \theta(u - v))v$, то понятие жесткости для нелинейных систем может быть определено аналогично. Заметим, однако, что за пределами класса систем линейных обыкновенных дифференциальных уравнений с постоянной матрицей полагаться на спектр как на источник надежной информации о распространении погрешности уже нельзя[Dekker, Verver].

1.2 Трудности, возникающие при численном решении

Трудность численного решения жестких систем обыкновенных дифференциальных уравнений выражается в нескольких аспектах. При использовании традиционных явных пошаговых методов, основы которых были заложены более века назад, возникают сильные ограничения на длину шага интегрирования.

Рассмотрим в качестве примера систему из двух независимых уравнений

$$\begin{cases} u_1'(t) = -\lambda_1 u_1(t), \\ u_2'(t) = -\lambda_2 u_2(t), t > 0, \lambda_2 \gg \lambda_1 > 0. \end{cases} \quad (1.2)$$

Эта система имеет решение $u(t) = (u_1(t), u_2(t))^T = (u_1^0 e^{-\lambda_1 t}, u_2^0 e^{-\lambda_2 t})^T$. При выписанных условия на λ_1 и λ_2 , очевидно, компонента $u_2(t)$ решения затухает гораздо быстрее, чем $u_1(t)$ и, начиная с некоторого момента t поведение вектора $u(t)$ почти полностью определяется компонентой $u_1(t)$. Однако при решении системы (1.2) численным методом величина шага интегрирования, как правило, определяется компонентой $u_2(t)$, не существенной с точки зрения поведения решения системы. Например, используя явный метод Эйлера,

мы из первого уравнения имеем ограничения на шаг $\tau \leq 2/\lambda_1$, а из второго - $\tau \leq 2/\lambda_2$ и, таким образом, ясно, что для решения системы (1.2) как цельного математического объекта шаг τ ограничен величиной $2/\lambda_2$. Такая же ситуация типична и при решении любой системы обыкновенных дифференциальных уравнений вида (1.1).

Учитывая выше сказанное, можно сделать вывод, что для решения жестких задач наиболее пригодны те численные методы, которые требуют наиболее слабых ограничений на величину шага численного интегрирования из соображений устойчивости. Таким образом, традиционные явные пошаговые методы, основы которых были заложены более века назад, мало пригодны из-за сильных ограничений на длину шага интегрирования, обусловленных неудовлетворительными свойствами устойчивости таких методов.

Начиная с пятидесятих годов двадцатого столетия, для интегрирования жестких задач стали применяться неявные одно- и многошаговые методы, обладающие хорошими свойствами устойчивости. Они позволяют находить приближенное решение жестких задач на достаточно больших шагах. Наиболее эффективными на данный момент считаются неявные коллокационные методы типа Рунге–Кутты. Однако и эти методы обладают существенным недостатком, который состоит в необходимости решения на каждом шаге системы нелинейных уравнений, размерность которой пропорциональна размерности дифференциального уравнения и количеству стадий метода. Поэтому машинная реализация таких методов является весьма громоздкой, и, кроме того, возникающие системы нелинейных уравнений в общем случае могут быть неразрешимы. Применение неявных методов также затрудняется необходимостью вычисления матрицы Якоби.

Стоит затронуть еще один важный момент, не относящийся напрямую к проблеме жесткости. Это вопрос о контроле точности приближенного решения. При пошаговом интегрировании для этих целей обычно используется техника «откатов»: если вычисленная (по правилу Рунге, например) оценка погрешности недостаточно мала, полученное приближение отбрасывается и вычисления повторяются заново с уже меньшей длиной шага. Такой подход, во-первых, не экономичен, так как полностью игнорируется полученное на данном шаге приближенное решение, которое может быть достаточно близким к точному. Вместо того, чтобы уменьшать шаг и повторять такие же вычисления, можно попытаться каким-то образом уточнить уже имеющееся приближение. Во-вторых, несколько «откатов» подряд могут привести к недопустимо малым значениям шага[Faleichik].

В настоящее время наиболее часто для этих целей используют либо неявные методы, либо методы, специально сконструированные для решения задач конкретного вида[Repnikov]. Хорошо применимыми при решении жестких систем являются методы, основанные на процессах установления, которые и рассматриваются в главе 2.

1.3 Примеры жестких задач

Глава 2

Методы установления

В настоящей главе приводятся основные сведения о вычислительном алгоритме, основанном на идее установления. Рассматриваются случаи применения как к линейной системе, так и к нелинейной системе обыкновенных дифференциальных уравнений. Линейный случай является исследуемым в численном эксперименте в главе 4.

2.1 Линейная задача

Рассмотрим задачу Коши для неоднородной линейной системы обыкновенных дифференциальных уравнений (ОДУ):

$$\begin{aligned} y'(t) &= Jy(t) + f(t), \\ y(t_0) &= y_0, \\ t &\in [t_0, t_0 + \tau], \\ y_0 &\in \mathbb{R}^N, \quad y : [t_0, t_0 + \tau] \rightarrow \mathbb{R}^N, \\ J &\in \mathbb{R}^N \times \mathbb{R}^N, \quad \tau \in [0, +\infty). \end{aligned} \tag{2.1}$$

Для нахождения приближения к $y(t_0 + \tau)$, $\tau > 0$ проинтегрируем её произвольным s-стадийным неявным методом типа Рунге-Кутты. Далее этот метод будем называть *базовым методом*. Базовый метод может быть представлен следующей таблицей Бутчера:

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \dots & \dots & \dots & \dots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} \tag{2.2}$$

Здесь $A = (a_{i,j})_{i,j=1}^s$ - так называемая матрица Бутчера базового метода. Тогда

$$y(t_0 + \tau) \approx y_1 = y_0 + \tau \sum_{i=1}^s b_i k_i,$$

где $\{k_i\}_{i=1}^s$ находятся как решение следующей системы линейных алгебраических уравнений (СЛАУ)

$$k_i = J(y_0 + \tau \sum_{j=1}^s a_{ij} k_j) + f(t_0 + c_i \tau).$$

В дальнейшем будем пользоваться матричной записью этой СЛАУ:

$$\begin{aligned} (\tau A \otimes J - I)k + g &= 0, \\ g &= (g_1, g_2, \dots, g_s)^T, \quad g_i = f(t_0 + c_i \tau) + Jy_0, \quad i = 1, \dots, s, \\ k &= (k_1, k_2, \dots, k_s)^T, \quad k_i \in \mathbb{R}^N. \end{aligned} \quad (2.3)$$

Здесь \otimes обозначает кронекеровское произведение матриц[], по определению которого получаем, что

$$G = \tau A \otimes J - I$$

- блочная матрица вида

$$\begin{pmatrix} -1 + \tau a_{11}J & \tau a_{12}J & \dots & \tau a_{1s}J \\ \tau a_{12}J & -1 + \tau a_{22}J & \dots & \tau a_{2s}J \\ \dots & \dots & \dots & \dots \\ \tau a_{s1}J & \tau a_{s2}J & \dots & -1 + \tau a_{ss}J \end{pmatrix}. \quad (2.4)$$

Рассмотрим вспомогательное уравнение

$$k' = (\tau A \otimes J - I)k + g = Gk + g = r(k), \quad (2.5)$$

которое в дальнейшем будем называть уравнением установления.

Очевидно, что точное решение уравнения (2.3) k^* будет являться стационарным решением (2.5). Для этого достаточно, чтобы спектр матрицы G целиком содержался в левой комплексной полуплоскости. Поэтому, если проинтегрировать (2.5) каким-нибудь численным методом, то можно получить приближение к решению (2.3).

Для решения (2.5) будем использовать явный метод Рунге-Кутты, задаваемый таблицей вида

$$\begin{array}{ccccccc} \alpha_{21} & & & & & & \\ \alpha_{31} & \alpha_{32} & & & & & \\ \dots & \dots & \dots & & & & \\ \alpha_{\sigma 1} & \alpha_{\sigma 2} & \dots & \alpha_{\sigma \sigma - 1} & & & \\ \hline \beta_1 & \beta_2 & \dots & \beta_{\sigma - 1} & \beta_{\sigma} & & \end{array} \quad (2.6)$$

Пусть ω - шаг по фиктивному времени. В результате получаем итерационный процесс вида

$$\begin{aligned} k^{l+1} &= \Phi(k^l) \\ \Phi(k) &= k + \omega \sum_{p=1}^{\sigma} \beta_p K_p(k), \\ K_p(k) &= G(k + \omega \sum_{q=1}^{p-1} \alpha_{pq} K_q(k)) + g. \end{aligned} \quad (2.7)$$

Учитывая специфику интегрирования уравнения установления (2.5), нужно выбрать ω , $\{\alpha_{ij}\}_{i,j=1}^{\sigma}$, $\{\beta_i\}_{i=1}^{\sigma}$. Подробно выбор коэффициентов вспомогательного метода описан в [Bondar].

2.2 Нелинейная задача

Рассуждения для нелинейного случая проходят во многом аналогично случаю линейному, поэтому остановимся только на различиях.

Рассмотрим систему нелинейных дифференциальных уравнений

$$\begin{aligned} y'(t) &= f(t, y(t)), \\ y(t_0) &= y_0, \\ t &\in [t_0, t_0 + \tau], \quad \tau > 0 \\ y_0 &\in \mathbb{R}^N, \quad y : [t_0, t_0 + \tau] \rightarrow \mathbb{R}^N, \\ f &: \mathbb{R}^N \rightarrow \mathbb{R}^N. \end{aligned} \tag{2.8}$$

Для интегрирования воспользуемся методом (2.2), причем в отличие от линейного случая применение запишем в симметричном виде:

$$Y_i = y_0 + \tau \sum_{j=1}^s a_{i,j} f(t_0 + c_j \tau, Y_j),$$

$$y(t_0 + \tau) \approx y_1 = y_0 + \tau \sum_{j=1}^s b_j f(t_0 + c_i \tau, Y_j),$$

что в векторной форме представимо как

$$\begin{aligned} Y &= e \otimes y_0 + \tau(A \otimes I)F(t_0, Y), \\ y_1 &= e \otimes y_0 + \tau(b^T \otimes I)F(t_0, Y). \end{aligned} \tag{2.9}$$

Здесь $e = (1, \dots, 1)$, $e \in \mathbb{R}^s$, $Y = (Y_1, \dots, Y_s)^T$, $F(t, Y) = (f(t + c_1 \tau, Y_1), \dots, f(t + c_s \tau, Y_s))^T$. Уравнение установления для (2.9) имеет вид

$$Y(\theta)' = \tau(A \otimes I)F(t_0, Y(\theta)) - Y(\theta) + e \otimes y_0 = \tilde{r}(Y(\theta)). \tag{2.10}$$

Соответствующий ему процесс установления имеет вид аналогичный (2.7):

$$\begin{aligned} Y^{l+1} &= \Phi(Y^l), \\ \Phi(Y) &= Y + \omega \sum_{p=1}^{\sigma} \beta_p K_p(Y), \\ K_p(Y) &= \tilde{r}(t_0, Y + \omega \sum_{q=1}^{p-1} \alpha_{pq} K_q(Y)). \end{aligned} \tag{2.11}$$

Полностью повторить конструирование вспомогательного метода как в случае линейной системы вообще говоря нельзя. Однако если задача позволяет, то можно провести линеаризацию и исследовать спектральные свойства уже

для неё, полностью повторяя приведенные в [Bondar] рассуждения о конструировании вспомогательных методов. Параметр ω полагаем таким, чтобы все собственные значения матрицы Якоби правой части уравнения (2.10) были по модулю меньше 1.

2.3 Спектральные свойства и скорость сходимости итерационного процесса

Процесс (2.11) представим в следующем виде:

$$k^{l+1} = R_\sigma(\omega G)k^l + P(\omega, G), \quad (2.12)$$

где R_σ - многочлен степени σ , называемый многочленом перехода (функцией устойчивости), P - функция, точный вид которой несущественен в данном случае.

Известно, что многочлен перехода во многом определяет свойства устойчивости метода интегрирования ОДУ. В нашем же случае он определяет свойства сходимости итерационного процесса. Мы заинтересованы в том, чтобы вспомогательный метод был устойчив на как можно большей области. В частности, область устойчивости

$$S = \{z \in \mathbb{C} : |R_\sigma(z)| < 1\}$$

должна содержать в себе спектр матрицы ωG [BondarFaleichikKiiv].

Положим ω равным спектральному радиусу матрицы G . Тем самым получаем, что в данном случае спектр матрицы ωG будет полностью содержаться в области устойчивости. Таким образом, спектр матрицы ωG имеет определяющее влияние на сходимость итерационного процесса (2.11).

Предположим, что нам известен спектр исходной матрицы J , и отследим каким может быть спектр результирующей матрицы

$$G = \tau(A \otimes J) - I.$$

По свойству кронекеровского произведения матриц [Dekker], собственные значения $\nu_{i,j}$ матрицы G равны

$$\nu_{ij} = \tau\mu_j\lambda_i - 1, \quad (2.13)$$

где μ_i - собственные значения матрицы A , λ_j - собственные значения матрицы J . То есть, над спектром исходной матрицы системы (2.1) производятся операции масштабирования, поворота и параллельного переноса. Собственные значения матрицы ωG , очевидно, являются смасштабированными на единичный круг собственными значениями G .

Описанные преобразования могут привести к выходу спектра матрицы G за пределы левой комплексной полуплоскости, что по определению плохо – процессы установления становятся неприменимыми. Но даже если выход не произошел, нет гарантии что такая ситуация не возникнет при увеличении шага интегрирования по времени. Чтобы избежать описанного выше нежелательного явления, можно осуществить так называемую операцию переобусловливания. Применение операции переобусловливания к методам установления подробно рассматривается в главе 3.1.

Проследим как будет изменяться ошибка на l -ой итерации: $\varepsilon^l = k^* - k^l$. Учитывая (2.12), получим:

$$\varepsilon^l = R_\sigma(\omega G)\varepsilon^{l-1}$$

Далее предположим, что у матрицы G имеется полный набор собственных векторов $\{\eta^i\}_{i=1}^N$. Тогда

$$\varepsilon^l = \sum_{i=1}^N \varepsilon_i^{l-1} R(\omega G)\eta^i = \sum_{i=1}^N \varepsilon_i^{l-1} R(\omega \nu_i)\eta^i,$$

здесь ν_i – собственное значение, соответствующее η^i . Таким образом,

$$\varepsilon_i^l = R_\sigma(\omega \nu_i)\varepsilon_i^{l-1} = (R(\omega \nu_i))^l \varepsilon_i^0.$$

Проанализируем последнее выражение. R_σ – многочлен перехода, и по построению $R_\sigma(0) = 1$. Учитывая непрерывность R_σ , получим что R_σ близок к 1 когда $\omega \nu_i$ близко к 0. Это значит, что компоненты ошибки, соответствующие малым величинам $\omega \nu_i$, будут уменьшаться медленно. Нетрудно видеть, что характеристикой, в достаточной мере описывающей свойства сходимости, является так называемое “спектральное число обусловленности” матрицы J исходной системы, $\kappa = \rho(J)\rho(J^{-1})$ (ρ здесь - спектральный радиус). Чем больше эта величина, тем медленнее будет сходиться итерационный процесс.

Глава 3

Ускорение сходимости

3.1 Переобусловливание

Переобусловливание - это процесс преобразования условий задачи для ее более корректного решения. Переобусловливание обычно связавно с уменьшением числа обусловленности задачи. Переобусловливаемая задача, как правило, затем решается итерационным методом. В линейной алгебре и вычислительной математике является переобусловливателем для матрицы если у матрицы число обусловленности меньше, чем у A . Также чаще говорят, что это переобусловливатель, чем просто так как точное значение обычно требует больших затрат на вычисление. Поэтому под переобуславливанием часто понимают вычисление точнее произведение вектора-столбца или матрицы векторов-столбцов на что обычно выполняется сложными программными пакетами с использованием итерационных методов, где в конечном итоге не вычисляются точные значения ни для A , ни для b .

3.1.1 Первый способ

3.1.2 Второй способ

3.2 Подавление компонент

Пусть x^0, x^1, \dots, x^k - вектора решений размерности N , полученные в ходе итераций процесса установления, а $\lambda_1 > \lambda_2 > \dots > \lambda_n$ - собственные значения.

Разложение ошибки по базису собственных векторов представимо в следующем виде:

- на k -ом шаге

$$r^k = Ax^k - b = \sum_{i=1}^N \alpha_i^k \xi_i \quad (3.1)$$

- на $k + 1$ -ом шаге

$$r^{k+1} = \sum_{i=1}^N \alpha_i^k R(\omega \lambda_i) \xi_i \quad (3.2)$$

Если λ_i велико, то вклад соответствующей компоненты в ошибку будет невелик. Пусть m - номер собственного значения, соответствующего компоненте с наибольшим вкладом в ошибку, то есть λ_m - наименьшее по модулю

собственное значение. Тогда

$$\begin{aligned} r^k &= \alpha_k^m \xi_m + \varepsilon^k, \\ r^{k+1} &= \alpha_m^k R(\omega \lambda_m) \xi_m + \varepsilon^{k+1}, \end{aligned} \quad (3.3)$$

где ε^k – погрешность, вносимая всеми компонентами помимо m -ой.

В предположении, что

$$\begin{aligned} \alpha_k^m \xi_m &\gg \varepsilon^k, \\ \alpha_m^k R(\omega \lambda_m) \xi_m &\gg \varepsilon^{k+1}, \end{aligned} \quad (3.4)$$

получаем, что величина ε^k значительно меньше вклада в погрешность m -ой компоненты.

Следовательно,

$$\frac{r_k^j}{r_j^k} \approx R(\omega \lambda_m), j = \overline{1, N} \quad (3.5)$$

Пусть

$$\tilde{x} \approx x^{k+1} + \delta^{k+1}$$

Тогда

$$\begin{aligned} A\tilde{x} - b &= 0 \\ A(x^{k+1} + \delta^{k+1}) - b &= 0 \\ Ax^{k+1} + \delta^{k+1} - b &= 0 \\ r^{k+1} + A\delta^{k+1} &= 0 \end{aligned} \quad (3.6)$$

Следовательно, в предположении, что ε^{k+1} мало, получаем

$$\begin{aligned} \delta^{k+1} &= -A^{-1}r^{k+1} = -A^{-1}(\alpha_m^k R(\omega \lambda_m) \xi_m + \varepsilon^{k+1}) = \\ &= -\alpha_m^k R(\omega \lambda_m) A^{-1} \xi_m - A^{-1} \varepsilon^{k+1} = \\ &= -\frac{1}{\lambda_m} \alpha_m^k R(\omega \lambda_m) \xi_m - A^{-1} \varepsilon^{k+1} = \\ &= -\frac{1}{\lambda_m} r^{k+1} - A^{-1} \varepsilon^{k+1} \approx -\frac{1}{\lambda_m} r^{k+1} \end{aligned} \quad (3.7)$$

Получим оценку для λ_m , для чего разложим многочлен перехода $R_\sigma(x)$ в ряд Тейлора:

$$R_\sigma(x) \approx R_\sigma(0) + R'_\sigma(0)x \approx 1 + xR'_\sigma(0) \quad (3.8)$$

Тогда,

$$R_\sigma(\omega \lambda_m) \approx 1 + R'_\sigma(0)\omega \lambda_m \approx \frac{r_j^{k+1}}{r_j^k}, j = \overline{1, N} \quad (3.9)$$

Следовательно, приближенное значение собственного значения λ_m , соответствующего компоненте с наибольшим вкладом в ошибку, можно вычислить следующим образом:

$$\lambda_m \approx \frac{r_j^{k+1}/r_j^k - 1}{\omega R'_\sigma(0)}, j = \overline{1, N} \quad (3.10)$$

Таким образом, получаем возможность уточнить текущее приближение следующим способом:

$$\tilde{x} \approx x^{k+1} + \frac{1}{\lambda_m} r^{k+1}. \quad (3.11)$$

При практической реализации, невзирая на то, что полученные оценки должны выполняться для любых допустимых j , рекомендуется для расчетов выбирать медианное значение ряда $\frac{r_1^{k+1}}{r_1^k}, \dots, \frac{r_N^{k+1}}{r_N^k}$.

Также отметим, что определение, когда именно нужно производить уточнение решения, требует дополнительных исследований. При проведении численного эксперимента пробовалась различная частота применения операции продавливания компонент, результаты чего описаны в главе 4

Глава 4

Численный эксперимент

4.1 Тестовая задача

4.2 Результаты численного эксперимента