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

Научные руководители:
Фалейчик Борис Викторович
кандидат физ.-мат. наук,
доцент кафедры выч. мат.

Бондарь Иван Васильевич
магистр физ.-мат. наук,
ассистент кафедры выч. мат.

Допущена к защите
«___» _____ 2015 г.
Зав. кафедры вычислительной математики
кандидат физико-математических наук, доцент
Мандрик Павел Алексеевич

Минск, 2015

РЕФЕРАТ

ПРИНЦИП УСТАНОВЛЕНИЯ, ЖЁСТКИЕ ЗАДАЧИ, МЕТОД РУНГЕ-КУТТЫ, СОБСТВЕННЫЕ ЗНАЧЕНИЯ, СПЕКТРАЛЬНЫЙ РАДИУС, ПЕРЕОБУСЛОВЛИВАНИЕ, ПОДАВЛЕНИЕ КОМПАНЕНТ, УСКОРЕНИЕ СХОДИМОСТИ

Объект исследования – методы решения жёстких задач.

Цель работы – разработка вычислительного алгоритма для решения жёстких дифференциальных задач на основе принципа установления.

Методы исследования – методы численного анализа.

Результатом является алгоритм решения жёстких дифференциальных задач с ускореной сходимостью, в основе которого лежит принцип установления.

Областью применения является решение задач математической физики.

РЭФЕРАТ

Дыпломная праца, 27 с., 0 мал., 0 табл., 8 крыніц.

ПРЫНЦЫП УСТАЛЯВАННЯ, ЖОРСТКІЯ ЗАДАЧЫ, МЕТАД РУНГЕ-КУТТЫ, УЛАСНЫЯ ЗНАЧЭННІ, СПЕКТРАЛЬНЫ РАДЫУС, ПЕРААБУМОЎЛЕННЕ, ПАДАЎЛЕННЕ КАМПАНЕНТ, ПАСКАРЭННЕ ЗБЕЖНАСЦІ

Аб'ект даследавання – метады рашэння жорсткіх задач.

Мэта працы – распрацоўка вылічальнага алгарытму для рашэння жорсткіх задач на аснове прынцыпа ўсталявання.

Метады даследавання – метады лікавага аналізу.

Вынікам з'яўляецца алгарытм рашэння жорсткіх дыферэнцыяльных задач з паскоранай збежнасцю, у аснове якога ляжыць прынцып усталявання.

Вобласцю выкарыстання з'яўляецца рашэнне задач матэматычнай фізікі.

REFERAT

Содержание

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* [1].

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 указаной 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 сходящися 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 $S = \dfrac{\max\limits_{1\le k\le n} |Re\lambda_k|}{\min\limits_{1\le k\le n} |Re\lambda_k|}$ 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

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

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 пятидесятых 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 [12].

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

Обширные численные эксперименты с жесткими задачами впервые провели Энрайт, Халл и Линдберг (1975). Их набор задач STIFFDETEST является стандартом для проверки качественности численных методов решения жестких систем [1]. В данном разделе приводятся некоторые тестовые задачи, которые в дальнейшем будут использованы в численном эксперименте.

1.3.1 HIRES: Модель дифференциации растительной ткани

HIRES – это химическая реакция с участием восьми реагентов была предложена Шефером (1975) для объяснения "роста и дифференциации растительной ткани независимо от фотосинтеза при высоких уровнях светового облучения". Готтвальд (1977) предложил использовать ее в качестве тестового примера [1].

Данный пример – типичный случай биохимической модели «умеренной» размерности (современные модели, например, фотосинтеза включают сотни уравнений подобного типа). Хотя данная модель является умеренно жесткой, тем не менее, ее лучше решать с помощью методов, предназначенных для решения жестких систем ОДУ [2].

Соответствующие уравнения имеют вид:

$$\begin{aligned}
y_1' &= -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007,\\
y_2' &= 1.71y_1 - 8.75y_2,\\
y_3' &= -10.03y_3 + 0.43y_4 + 0.035y_5,\\
y_4' &= 8.32y_2 + 1.71y_3 - 1.12y_4,\\
y_5' &= -1.745y_5 + 0.43y_6 + 0.43y_7,\\
y_6' &= -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7,\\
y_8' &= -y7';
\end{aligned} \tag{1.3}$$

$$y_1(0) = 1, y_2(0) = y_3(0) = ... = y_7(0) = 0, y_8(0) = 0.0057,$$

и для выдачи были выбраны значения $x_{out} = 312.8122$ и 421.8122.

1.3.2 ROBER: Модель химических реакций Робертсона

Один из первых и самых популярных примеров жесткой системы ОДУ принадлежит Робертсону (1966) и имеет вид, типичный для моделей химической кинетики – в правой части системы стоят полиномы второй степени от концентраций.

ROBER – реакция Робертсона имеет вид:

$$\begin{aligned} y_1' &= -0.04y_1 + 10^4 y_2 y_3, \\ y_1' &= 0.04y_1 - 10^4 y_2 y_3 - 3 \cdot 10^7 y_2^2, \\ y_3' &= 3 \cdot 10^7 y_2^2; \end{aligned} \tag{1.4}$$

$$y_1(0) = 1, y_2(0) = 0, y_3(0) = 0.$$

Обычно эту задачу рассматривали на отрезке $0 \leq x \leq 40$, пока Хайндмарш не обнаружил, что многие программы терпят неудачу, если x становится очень большим (например, 10^11). Причина заключается в том, что как только компонента численного решения y_2 случайно становится отрицательной, онап стремится к $-\infty$, и выполнение программы прекращается из-за переполнения. Поэтому для выдачи выбирают значения:

$$x_{out} = 1, 10, 10^2, 10^3, ..., 10^11$$

1.3.3 OREGO: Модель Филда–Нойса «орегонатор»

OREGO – орегонатор, знаменитая модель с периодическим решением, описывающая реакцию Белоусова-Жаботинского:

$$\begin{aligned} y_1' &= 77.27(y_2 + y_1(1 - 8.375 \cdot 10^{-6} y_1 - y_2)), \\ y_1' &= \frac{1}{77.27}(y_3 - (1 + y_1)y_2), \\ y_3' &= 0.161(y_1 - y_3), \end{aligned} \tag{1.5}$$

$$y_1(0) = 1, y_2(0) = 2, y_3(0) = 3, \qquad x_{out} = 30, 60, 90, ..., 360.$$

На то, что система жесткая, указывают большие различия в константах скоростей реакций – есть процессы «быстрые», и есть «медленные». Так как переменные системы — концентрации ($HBrO_2$, Br^- и $Ce(IV)$ соответственно), то начальные условия для системы выбирают положительными и, как правило, достаточно близкими к 0.

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 3.

2.1 Линейная задача

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

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

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:

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

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

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

- блочная матрица вида

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

Как правило, (2.3) решается методом Ньютона или с помощью его модификаций. Однако, данных подход требует значительных вычислительных затрат при факторизации матрицы Якоби в случае ее большой размерности. В этой же работе рассматривается альтернативный подход, основанный на процессах установления [5].

Рассмотрим вспомогательное уравнение

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

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

$$\begin{array}{lllll}
\alpha_{21} & & & & \\
\alpha_{31} & \alpha_{32} & & & \\
... & ... & ... & & \\
\alpha_{\sigma 1} & \alpha_{\sigma 2} & ... & \alpha_{\sigma\sigma-1} & \\
\hline
\beta_1 & \beta_2 & ... & \beta_{\sigma-1} & \beta_\sigma
\end{array} \tag{2.6}$$

В результате применения явного метода Рунге-Кутты,, получаем семейство методов, именуемых обобщенными итерациями Пикара.

Пусть ω - шаг по фиктивному времени.

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

Учитывая специфику интегрирования уравнения установления (2.5), нужно выбрать ω, $\{\alpha_{ij}\}_{i,j=1}^{\sigma}$, $\{\beta_i\}_{i=1}^{\sigma}$. Один из способов выбора коэффициентов предложен в [4]. В частности, используя тот факт, что процесс (2.7) может быть записан в виде

$$k^{l+1} = R_\sigma(\omega G)k^l + P(\omega, G), \tag{2.8}$$

где R_σ – полином степени σ, многочлен степени σ, называемый функцией устойчивости вспомогательного метода, P – некоторый многочлен от матрицы.

Многочлен R_σ определяет свойства устойчивости итерационного метода. В нашем случае, он определяет свойства сходимости итерационного процесса. В частности, область устойчивости

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

должна содержать в себе спектр матрицы ωG [4].

Запишем многочлен устойчивости в виде

$$R_\sigma(z) = 1 + \sum_{j=1}^{\sigma} a_j z^j,$$

где $z \in \mathbb{C}$. Коэффициенты $\{a_i\}_{i=1}^{\sigma}$, $a_i \in \mathbb{R}$ будем выбирать таким образом, чтобы минимизировать функцию

$$F(a_1, ..., a_\sigma) = \int_0^1 \int_{\pi-\alpha}^{\pi+\alpha} |R_\sigma(\rho e^{\mathrm{i}\varphi})|^2 d\varphi d\rho \tag{2.9}$$

Здесь α - некоторый заранее заданный угол, определяющий область устойчивости полинома R_σ [6], [7]. Вообще, α нужно выбирать исходя из представлений о спектре матрицы G, поскольку чем лучше будет приближен спектр, тем более эффективным будет метод. Параметр ω выбирается так, чтобы спектр матрицы ωG полностью содержался в области устойчивости. Для этого достаточно положить ω равным спектральному радиусу матрицы G.

Имея сконструированный многочлен устойчивости, мы можем восстановить матрицу Бутчера вспомогательного метода, используя подход, описанный в [1], [8]. Более подробно выбор коэффициентов вспомогательного метода описан в [11].

2.2 Нелинейная задача

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

Рассмотрим систему нелинейных дифференциальных уравнений

$$\begin{aligned} &y'(t) = f(t, y(t)), \\ &y(t_0) = y_0, \\ &t \in [t_0, t_0 + \tau], \quad \tau > 0 \\ &y_0 \in \mathbb{R}^N, \quad y : [t_0, t_0 + \tau] \to \mathbb{R}^N, \\ &f : \mathbb{R}^N \to \mathbb{R}^N. \end{aligned} \tag{2.10}$$

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

$$\begin{aligned} &Y = e \otimes y_0 + \tau (A \otimes I) F(t_0, Y), \\ &y_1 = e \otimes y_0 + \tau (b^T \otimes I) F(t_0, Y). \end{aligned} \tag{2.11}$$

Здесь $e = (1, ..., 1)$, $e \in \mathbb{R}^s$, $Y = (Y_1, ..., Y_s)^T$, $F(t, Y) =$
$= (f(t + c_1\tau, Y_1), ..., f(t + c_s\tau, Y_s))^T$. Уравнение установления для (2.11) имеет вид

$$Y(\theta)' = \tau (A \otimes I) F(t_0, Y(\theta)) - Y(\theta) + e \otimes y_0 = \tilde{r}(Y(\theta)). \tag{2.12}$$

Соответствующий ему процесс установления имеет вид аналогичный (2.7):

$$\begin{aligned} &Y^{l+1} = \Phi(Y^l), \\ &\Phi(Y) = Y + \omega \sum_{p=1}^{\sigma} \beta_p K_p(Y), \\ &K_p(Y) = \tilde{r}(t_0, Y + \omega \sum_{q=1}^{p-1} \alpha_{pq} K_q(Y)). \end{aligned} \tag{2.13}$$

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 (2.12) были по модулю меньше 1.

2.3 Спектральные свойства и скорость сходимости итерационного процесса

Как уже было сказано, процесс (2.13) представим в следующем виде:

$$k^{l+1} = R_\sigma(\omega G)k^l + P(\omega, G), \tag{2.14}$$

где R_σ – многочлен перехода (функцией устойчивости), P – функция, точный вид которой несущественен в данном случае.

Мы уже определились, что многочлен перехода во многом определяет свойства сходимости итерационного процесса, и положив ω равным спектральному радиусу матрицы G, получили, что спектр матрицы ωG будет полностью содержаться в области устойчивости. Таким образом, спектр матрицы ωG имеет определяющее влияние на сходимость итерационного процесса (2.13).

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

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

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

$$\nu_{ij} = \tau\mu_j\lambda_i - 1, \tag{2.15}$$

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 становится 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 2.4.

Проследим как будет изменяться ошибка на l-ой итерации:$\varepsilon^l = k^* - k^l$. Учитывая (2.14), 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, $\varkappa = \rho(J)\rho(J^{-1})$ (ρ здесь - спектральный радиус). Чем больше эта величина, тем медленнее будет сходиться итерационный процесс.

2.4 Переобусловливание

Решение системы линейных алгебраических уравнений вида

$$Ax = b, \tag{2.16}$$

где A – $(n \times n)$-матрица коэффициентов, является центральной задачей многих численных моделей, и часто решение такой системы является наиболее трудоемкой частью вычислений. В вычислительной математике для ускорения сходимости итерационных методов решения таких систем часто применяют подход, известный как *переобусловливание*.

Как известно, переобусловливание – это процесс преобразования системы (2.16) в систему с улучшенными свойствами сходимости итерационного процесса. Вообще говоря, целью переобусловливания является уменьшение числа обусловленности матрицы: чем меньше число обусловленности матрицы, тем быстрее сходится итерационный процесс [13].

Один из частных приемов такого подхода заключается в решении умноженной слева исходной системы СЛАУ на некоторую невырожденную матрицу:

$$MAx = Mb, \tag{2.17}$$

причем такой прием имеет смысл только в том случае, если число обусловленности матрицы коэффициентов MA меньше, чем у исходной матрицы.

Матрица M, с помощью которой осуществляется такое преобразование, называется *переобусловливателем.*

Решатели с переобуславливанием обычно эффективнее, чем использование простых решателей в случае больших и особенно в случае разреженных матриц. Итерационные решатели с переобуславливанием могут использовать безматричные методы, в которых матрица коэффициентов A не хранится отдельно, а доступ к ее элементам происходит через произведения матриц-векторов [16].

Переобусловленную слева систему также можно записать в виде

$$M(Ax - b) = 0. \tag{2.18}$$

Вообще говоря, хороший переобусловливатель M должен удовлетворять следующим требованиям:

- Переобусловленная система должна легко решаться.
- Переобусловливатель должен легко вычисляем и применим [13].

Первое свойство означает, что итерационный процесс должен быстро сходиться, в то же время, второе свойство говорит о том, что каждая итерация не должна быть слишком затратна. Заметим, то эти два тербования находятся в конкуренции между собой, и необходимо всегда соблюдать некий баланс между ними.

Так наиболее легко вычисляемым переобусловливателем будет являться $M = I$. Очевидно, что применяя такой переобусловливатель, мы получим исходную СЛАУ, и смысла в таком переобусловливателе нет. Другая крайность, выбор в качестве переобусловливателя обратной матрицы исходной матрицы коэффициентов, т.е. $M = A^{-1}$. В этом случае итерационный процесс будет наиболее быстро сходится, т.к. будет получено оптимальное число обусловленности 1, требующее одной итерайции, чтобы процесс сошелся. Однако, поскольку нахождение обратной матрицы – задача не тривиальная и более трудоемкая, чем решение исходной системы, построение переобуславлевателя такого вида на практике не целесообразно.

Как можно заметить, построение эффективного переобусловливателя может быть совсем не тривиальной задачей. Способы построения переобуславливателей в контексте процессов установления и приводятся в данной главе.

2.4.1 Первый способ

Чтобы избежать описанного в главе 2.3 нежелательного явления, можно осуществить операцию переобусловливания [9], [10] умножив систему (2.3)

слева на $A^{-1} \otimes I$ (здесь $I \in \mathbb{R}^N \times \mathbb{R}^N$). Тогда система (2.3) примет вид:

$$(\tau I_s \otimes J - A^{-1} \otimes I_N)k + \tilde{g} = 0, \tag{2.19}$$
$$\tilde{g} = (\tilde{g}_1, \tilde{g}_2, .., \tilde{g}_s)^T, \quad \tilde{g}_i = (A^{-1} \otimes I_N)(f(t_0 + c_i\tau) + Jy_0), \quad i = 1, .., s,$$
$$k = (k_1, k_2, .., k_s)^T, \quad k_i \in \mathbb{R}^N,$$

здесь I_s и I_N – единичные матрицы размерности $s \times s$ и $N \times N$ соответственно. Также стоит отметить, что точные решения (2.3) и (2.19) совпадают. По свойствам кронекеровского произведения, собственные значения матрицы

$$\tilde{G} = (\tau I_s \otimes J - A^{-1} \otimes I_N)$$

равны

$$\tilde{\nu}_{ij} = \tau\lambda_i - \frac{1}{\mu_j}.$$

Здесь отсутствует преобразование вращения, выполняются только масштабирование и параллельный перенос. Если все μ_j имеют положительные вещественные части (а это справедливо для большинства применяемых на практике методов Рунге–Кутты), то полностью отсутствует опасность выхода спектра матрицы $\tilde{G}$ за пределы левой комплексной полуплоскости.

Кроме вышеописанной пользы, операция переобуславливания также позволяет сократить объем вычислений, а также позволяет экономить память при хранении $\tilde{G}$ в виде разреженной матрицы [11].

2.4.2 Второй способ

Еще один подход к переобусловливанию связан с приближением обратной матрицы. Для этого используем особый тип переобуславливателей, известный как полиномиальный переобуславливатель $M = p_k(A)$ [14]:

$$p_k(A)Ax = p_k(A)b, \tag{2.20}$$

или

$$p_k(A)(Ax - b) = 0, \tag{2.21}$$

где $p_k(A)$ – многочлен от матрицы k-ой степени следующего вида:

$$p_k(A) = c_kA^k + c_{k-1}A^{k-1} + ... + c_1A + c_0. \tag{2.22}$$

Уже упоминалось, что наилучшим в смысле скорости сходимости итерационного процесса переобусловливателем является обратная матрица коэффициентов системы алгебраических уравнений. Вместо точного вычисления обратной матрицы, что, очевидно, не целесообразно, попробуем найти какое-то ее приближение, используя приведенный выше полиномиальный переобусловливатель (2.22), т.е. попытаемся найти приближение обратной функции от матрицы.

Как известно, для того, чтобы определить какую-либо функцию от матрицы, достаточно определить ее на спектре матрицы. Предположим, что у нас имеется некоторая квадратная матрица A размерности N с полным набором собственных значений $\lambda_{i i=1}^{N}$. Также для простоты изложения предполагаем, что собственные значения все различны и среди них нету нулевых значений. Тогда произвольную функцию от матрицы A можно записать в виде:

$$f(A) = \sum_{i=1}^{N} f(\lambda_i) \prod_{j \neq i} \frac{A - \lambda_j}{\lambda_i - \lambda_j} \tag{2.23}$$

В частности, для обратной функции

$$f_{inv}(x) = x^{-1}$$

имеем

$$f_{inv}(A) = \sum_{i=1}^{N} f(\lambda_i^{-1}) \prod_{j \neq i} \frac{A - \lambda_j}{\lambda_i - \lambda_j}. \tag{2.24}$$

Если нам известны все собственные значения матрицы коэффициентов A, то переобуславливатель

$$M = f_{inv}(A) \tag{2.25}$$

будет в точности равен обратной матрице для матрицы A. Отметим, что, как правило, спектр матрицы A неизвестен, но можно оценить границы, в которых находятся собственные значения. Предположим, нам известны числа a, b, такие, что

$$a \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N \leq b,$$

где $\lambda_1, ..., \lambda_N$ – упорядоченные по возрастанию собственные значения матрицы A. Тогда можно приблизать функцию $f_{inv}(x)$ на отрезке $[a, b]$ каким либо способом. Например, проинтерполировав многочленом Лагранжа по чебышовским узлам на этом отрезке. Пусть q – степень интерполяционного многочлена. Узлы $p_{i i=0}^{q}$, распределенные на отрезке $[a, b]$ вычисляются по известным формулам:

$$p_i = \frac{b+a}{2} + \frac{b-a}{2} \cos(\frac{\pi(2i+1)}{2q+2}), i = \overline{1, q} \tag{2.26}$$

В качестве переобусловливателя тогда будетвыступать многочлен

$$p_q(A) = \sum_{i=1}^{N} f(p_i^{-1}) \prod_{j \neq i} \frac{A - p_j}{p_i - p_j}. \tag{2.27}$$

.

В численом эксперименте в качестве переобусловливателя будем использовать наименее уклоняющийся от обратной функции $\frac{1}{1-t}$ на промежутке $(\frac{-1}{\gamma}, \frac{1}{\gamma})$ полином $f(t)$. Очевидно, что

$$f(t) = \gamma F(\gamma t),$$

где $F(t)$ есть полином, наименее уклоняющийся от функции $\frac{1}{\gamma - t}$ на промежутке $(-1, 1)$. Подход ускорения итерационных процессов, основанный на использовании функции $f(t)$, в литературе известен как универсальный алгоритм, наилучший в смысле второго критерия [15].

Полином степени $s-1$, удовлетворяющий последнему требованию, был известен еще Чебышеву. Именно,

$$F_{s-1}(t) = -\frac{2\alpha^{\frac{s}{2}}}{(1-\alpha)^2}\frac{1}{\gamma - t}\left[T_s(t) - 2\sqrt{\alpha}T_{s-1}(t) + \alpha T_{s-2}(t)\right] + \frac{1}{\gamma - t}, \quad (2.28)$$

где

$$\alpha = (\gamma - \sqrt{\gamma^2 - 1})^2, T_s(t) = \cos s \arccos t.$$

Таким образом,

$$f_{s-1}(t) = \frac{1}{1-t} - \frac{2\alpha^{\frac{s}{2}}}{(1-\alpha)^2}\frac{1}{1-t}\left[T_s(\gamma t) - 2\sqrt{\alpha}T_{s-1}(\gamma t) + \alpha T_{s-2}(\gamma t)\right]. \quad (2.29)$$

Применяя изложенный выше подход к процессам установления нужно учитывать некоторые особенности. Из условий применимости методом установления следует, что все собственные значения матрицы находятся в левой комплексной полуплоскости в полукруге единичного радиуса, то в качестве переобусловливателя применяем полином (2.29), "сдвинутый"по действительной оси на промежуток (-2, 0), т.е.

$$\begin{aligned} &p_{q-1}(t) = -f_{q-1}(t-1), t \in (-2, 0), \\ &p_{q-1}(t) \approx \frac{1}{t} \end{aligned} \quad (2.30)$$

Также заметим, что шаг по фиктивному времени для переобусловленной задачи будет, вообще говоря, не таким, как для исходной задачи, поэтому нужно получить какие-то оценки для максимального собственного значения переобусловленной задачи. Для этих целей можно использовать степенной метод.

Плюсам данного подхода также является тот факт, что матрицу $p_q(A)$ не обязательно вычислять и хранить в памяти компьютера. Рекомендуется сначала вычислять коэффициенты $p_q(t)$, а затем организовать экономичные вычисления решения как линейную комбинацию векторов-матриц по схеме Горнера.

2.5 Подавление компонент

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

Разложение ошибки по базису собственных векторов представимо в следующем виде:

- на k-ом шаге

$$r^k = Ax^k - b = \sum_{i=1}^{N} \alpha_i^k \xi_i \tag{2.31}$$

- на $k+1$-ом шаге

$$r^{k+1} = \sum_{i=1}^{N} \alpha_i^k R(\omega\lambda_i)\xi_i \tag{2.32}$$

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

$$\begin{aligned} r^k &= \alpha_k^m \xi_m + \varepsilon^k, \\ r^{k+1} &= \alpha_m^k R(\omega\lambda_m)\xi_m + \varepsilon^{k+1}, \end{aligned} \tag{2.33}$$

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

В предположении, что

$$\begin{aligned} \alpha_k^m \xi_m &\gg \varepsilon^k, \\ \alpha_m^k R(\omega\lambda_m)\xi_m &\gg \varepsilon^{k+1}, \end{aligned} \tag{2.34}$$

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

Следовательно,

$$\frac{r_k^j}{r_j^k} \approx R(\omega\lambda_m), j = \overline{1, N} \tag{2.35}$$

Пусть

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

.

Тогда

$$\begin{aligned} A\widetilde{x} - b &= 0 \\ A(x^{k+1} + \delta^{k+1}) - b &= 0 \\ Ax^{k+1} + \delta^{k+1} - b &= 0 \\ r^{k+1} + A\delta^{k+1} &= 0 \end{aligned} \tag{2.36}$$

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

$$\begin{aligned}\delta^{k+1} &= -A^{-1}r^{k+1} = -A^{-1}(\alpha_m^k R(\omega\lambda_m)\xi_m + \varepsilon^{k+1}) = \\ &= -\alpha_m^k R(\omega\lambda_m)A^{-1}\xi_m - A^{-1}\varepsilon k + 1 = \\ &= -\frac{1}{\lambda_m}\alpha_m^k R(\omega\lambda_m)\xi_m - A^{-1}\varepsilon^{k+1} = \\ &= -\frac{1}{\lambda_m}r^{k+1} - A^{-1}\varepsilon^{k+1} \approx -\frac{1}{\lambda_m}r^{k+1}\end{aligned} \tag{2.37}$$

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

$$R_\sigma(x) \approx R_\sigma(0) + R'_\sigma(0)x \approx 1 + xR'_\sigma(0) \tag{2.38}$$

Тогда,

$$R_\sigma(\omega\lambda_m) \approx 1 + R'_\sigma(0)\omega\lambda_m \approx \frac{r_j^{k+1}}{r_j^k}, j = \overline{1, N} \tag{2.39}$$

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

$$\lambda_m \approx \frac{{}^{r_j^{k+1}}\!/_{r_j^k} - 1}{\omega R'_\sigma(0)}, j = \overline{1, N} \tag{2.40}$$

Таким образом, получаем возможность уточнить текущее приближение следующим способом:

$$\widetilde{x} \approx x^{k+1} + \frac{1}{\lambda_m}r^{k+1}. \tag{2.41}$$

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

Также отметим, что определение, когда именно нужно производить уточнение решения, требует дополнительных исследований. При проведении численного эксперимента пробовалась различная частота применения операции продавления компонент, результаты чего описаны в главе 3.

Глава 3
Численный эксперимент

3.1 Тестовая задача

3.2 Результаты численного эксперимента

Литература

1. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи./ Пер. с англ. — М.: Мир, 1999. — 685 с.

2. Холодов А. С., Лобанов А. И., Евдокимов А.В. Разностные схемы для решения жестких обыкновенных дифференциальных уравнений в пространстве неопределенных коэффициентов — М.: МФТИ, 2001. — С. 45-46.

3. Деккер К., Вервер Я. Устойчивость методов Рунге—Кутты для жестких нелинейных дифференциальных уравнений/ Пер. с англ.— М.: Мир, 1988.

4. Фалейчик Б. В., Бондарь И. В. Реализация неявных методов для жестких задач методом установления// Theoretical and Applied Aspects of Cybernetics. Proceedings of the International Scientific Conference of Students and Young Scientists – Kyiv: Bukrek, 2011. С. 297-299.

5. Bondar I. V., Faleichik B. V. Iterated Runge-Kutta Methods with Parallelization Capability for Stiff Problems // Theoretical and Applied Aspects of Cybernetics. Proceedings of the 3rd International Scientific Conference of Students and Young Scientists – Kyiv: Bukrek, 2013. P. 336

6. Faleichik B. V. Explicit Implementation of Collocation Methods for Stiff Systems with Complex Spectrum // Journal of Numerical Analysis, Industrial and Applied Mathematics. Vol. 5

7. Фалейчик Б. В., Бондарь И. В. Реализация неявных методов Рунге-Кутты с использованием принципа установления //Аналитические методы анализа и дифференциальных уравнений: Тез. докл. междунар. конф. 12-17 сент. 2011г, Минск, Беларусь. С. 146-147

8. Фалейчик Б. В. Реализация неявных методов для жестких задач с использованием обобщенных итераций Пикара // Тр. 6-й международной конференции «Аналитические методы анализа и дифференциальных уравнений»: в двух томах – Т.1 Математический анализ. — Минск: Институт математики НАН Беларуси, 2012. С. 131–135.

9. Фалейчик Б. В., Бондарь И. В. Реализация неявных методов Рунге-Кутты для больших жестких систем//Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях: XV Республиканская научная конференция студентов и аспирантов “Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях”, 26-28 марта

2012 г.:[материалы]: в 2 ч. Ч.1/редкол. : Демиденко О.М. – Гомель: ГГУ им. Ф. Скорины, 2012. С. 175-176

10. Бондарь И. В. Итерационные процессы установления для жестких линейных задач //Тр. 69-й ежегодной научной конференции студентов и аспирантов БГУ: допущено в печать.

11. Бондарь И. В. Итерационные процессы установления для жестких задач //Республиканский конкурс научных работ студентов высших учебных заведений: допущено в печать.

12. Фалейчик Б. В. Вычислительные алгоритмы решения жестких задач на основе процессов установления // Труды института математики НАН Беларуси. – 2004. – Т. 12, № 1. – С. 45-48.

13. Benzi M. Preconditioning Techniques for Large Linear Systems: A Survey // Journal of Computational Physics. Vol.182, 2002. P. 418-477.

14. Chen K. Matrix Preconditioning Techniques and Applications // Cambridge University Press, 2005, P. 195-197.

15. Фадеев, Д.К., Фадеева, В.Н. Вычислительные методы линейной алгебры // М:Физматгиз, 1963. – С. 570-572.

16. Интернет адрес: https://ru.wikipedia.org/wiki/Предобуславливание