

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
БЕЛАРУСЬ**

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра вычислительной математики

Касияник Алексей Леонидович

**МЕТОДЫ УСТАНОВЛЕНИЯ ДЛЯ ЖЕСТКИХ ЗАДАЧ: УСКОРЕНИЕ
СХОДИМОСТИ В ПРОЦЕССЕ РЕШЕНИЯ**

Курсовая работа
студента 4 курса 5 группы

“Допустить к защите”
Руководитель работы
Бондарь Иван Васильевич
ассистент кафедры выч. мат.

Руководитель
Бондарь Иван
Васильевич
ассистент кафедры выч. мат.

“__” _____ 2014 г

Минск 2014

АННОТАЦИЯ

Касияник А.Л. Методы установления для жестких задач: ускорение сходимости в процессе решения: Курсовая работа / Минск: БГУ, 2014. – 22 с.

В курсовой работе рассматриваются различные способы ускорения сходимости в процессе решения с помощью методов установления при решении жестких систем дифференциальных уравнений. На примере тестовой задачи проведен численный эксперимент для линейного случая.

АНАТАЦЫЯ

Касіянік А.Л. Метады ўсталявання для жорсткіх задач: паскарэнне збежнасці ў працэсе рашэння: Курсавая праца / Мінск: БДУ, 2014. – 22 с.

У курсавой працы разглядаюцца розныя падыходы паскарэння збежнасці ў працэсе рашэння с дапамогай метадаў усталявання пры рашэнні жорсткіх сістэм дыферэнцыяльных раўнанняў. На прыкладзе тэставай задачы праведзены лікавы эксперымент для лінейнага выпадку.

ANNOTATION

Kasiyanik A.L. Steadying process methods for stiff systems of differential equations: convergence acceleration in the process of solving: Coursework / Minsk: BSU, 2014. – 22 p.

In the course work considers the various approaches of convergence acceleration in the process of solving stiff systems of differential equations by iterative steadying process. There is numerical experiment for test problem in this work for linear system.

РЕФЕРАТ

Курсовая работа, 22 с., 1 таблица, 9 источников.

Ключевые слова: ПРИНЦИП УСТАНОВЛЕНИЯ, ЖЕСТКИЕ ЗАДАЧИ, МЕТОДЫ РУНГЕ-КУТТЫ, ПЕРЕОБУСЛАВЛИВАНИЕ, СПЕКТРАЛЬНЫЕ СВОЙСТВА, УСКОРЕНИЕ СХОДИМОСТИ

Объект исследования: объектом исследования является методы решения жестких задач.

Цель исследования: разработка вычислительного алгоритма для решения жестких дифференциальных задач на основе принципа установления.

Методы исследования: методы численного анализа.

Результаты: изучен алгоритм решения жестких дифференциальных задач, в основе которого лежит принцип установления; предложены способы ускорения сходимости в процессе решения; проведен численный эксперимент

Область применения: решение задач математической физики.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
РЕФЕРАТ	3
ВВЕДЕНИЕ	5
1 Итерационный процесс установления.....	7
1.1 Линейный случай.....	7
1.2 Нелинейный случай.....	10
2 Спектральные свойства и скорость сходимости итерационного процесса	12
3 Способы ускорения сходимости в процессе решения	14
3.1 Операция переобуславливания	14
3.1.1 Применение операции переобуславливания для процесса установления	16
3.2 Подавление медленно сходящихся компонент	17
4 Численный эксперимент для линейной системы	19
ЗАКЛЮЧЕНИЕ.....	21
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	22

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 методы, основанные на процессе установления, которые применимы в указанной выше ситуации.

Особое внимание в работе посвящено исследованию способов ускорения сходимости решения в процессе решения методами установления. Было замечено, что компоненты ошибки, которые соответствуют малым собственным значениям, сходятся медленно. Поэтому предположительно подавление ошибки медленно сходящихся компонент может дать существенную прибавку в скорости сходимости итерационного процесса. Также известно, что свойства сходимости процесса существенным образом зависят от так называемого «спектрального числа обусловленности»[1] матрицы J исходной системы. Чем больше эта величина, тем медленнее будет сходиться процесс. Классическим способом уменьшения числа обусловленности, и тем самым улучшения спектральных свойств задачи, является применения операции переобуславливания.

Исследованию описанных проблем, разработке вычислительного алгоритма,

который бы основывался на идее установления и при этом превосходил в скорости известные алгоритмы и посвящается данная работа.

1 Итерационный процесс установления

В настоящей главе приводятся основные сведения о вычислительном алгоритме, основанном на идее установления. Рассматриваются случаи применения как к линейной, так и нелинейной системы обыкновенных дифференциальных уравнений. Нелинейный случай является исследуемым в численном эксперименте в главе 3.

1.1 Линейный случай

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

$$\begin{aligned}y'(t) &= Jy(t) + f(t) \\ y(t_0) &= y_0, \\ t &\in [t_0, t_0 + \tau], \\ y_0 &\in \mathbb{R}^n, y: [t_0, t_0 + \tau] \rightarrow \mathbb{R}^n, \\ J &\in \mathbb{R}^n \times \mathbb{R}^n, \tau \in [0, +\infty).\end{aligned}\tag{1.1}$$

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:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \cdots & \cdots & \cdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}\tag{1.2}$$

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

$$y(t_0 + \tau) \approx y_1 = y_0 + \tau \sum_{i=1}^s b_i k_i,\tag{1.3}$$

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

$$k_i = J \left(y_0 + \tau \sum_{j=1}^s a_{ij} k_j \right) + f(t_0 + c_i \tau). \quad (1.4)$$

Более удобной является матричная запись этой СЛАУ, которой и будем пользоваться в дальнейшем:

$$\begin{aligned} (\tau A \otimes J - I)k + g &= 0, \\ g &= (g_1, g_2, \dots, g_s)^T, g_i = f(t_0 + c_i \tau) + J y_0, i = 1, \dots, s, \\ k &= (k_1, k_2, \dots, k_s)^T, k_i \in \mathbb{R}^n. \end{aligned} \quad (1.5)$$

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

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

— блочная матрица вида

$$\begin{pmatrix} -1 + \tau a_{11}J & \dots & \tau a_{1s}J \\ \dots & \dots & \dots \\ \tau a_{s1}J & \dots & -1 + \tau a_{ss}J \end{pmatrix}.$$

Рассмотрим вспомогательное уравнение

$$k' = (\tau A \otimes J - I)k + g = Gk + g = r(k), \quad (1.6)$$

которое в дальнейшем будем называть *уравнением установления*.

Очевидно, что точное решение уравнения (1.5) k^* будет являться стационарным решением (1.6). Для этого достаточно, чтобы спектр матрицы G целиком содержался в левой комплексной полуплоскости. Поэтому, если проинтегрировать (1.6) каким-нибудь численным методом, то можно получить приближение к решению (1.5).

Для решения (1.6) будем использовать явный метод Рунге-Кутты, в дальнейшем называемый *вспомогательным методом*, задаваемый таблицей вида:

$$\begin{array}{ccccccc}
& \alpha_{21} & & & & & \\
& \alpha_{31} & \alpha_{32} & & & & \\
& \dots & \dots & \dots & & & \\
& & & & & & \\
& \frac{\alpha_{\sigma 1}}{\beta_1} & \frac{\alpha_{\sigma 2}}{\beta_2} & \dots & \frac{\alpha_{\sigma \sigma-1}}{\beta_{\sigma-1}} & \frac{\alpha_{\sigma \sigma}}{\beta_{\sigma}} &
\end{array} \quad (1.7)$$

Пусть ω – величина шага по фиктивному времени. В результате получим семейство итерационных процессов вида:

$$\begin{aligned}
k^{l+1} &= \Phi(k^l), \\
\Phi(k) &= k + \omega \sum_{p=1}^{\sigma} \beta_p K_p(k), \quad (1.8) \\
K_p(k) &= G \left(k + \sum_{q=1}^{p-1} \alpha_{pq} K_q(k) \right) + g.
\end{aligned}$$

Выбор ω , $\{\alpha_{ij}\}_{i,j=1}^{\sigma}$, $\{\beta_i\}_i^{\sigma}$ производится учитывая специфику интегрирования уравнения установления (1.6). Подробно выбор коэффициентов вспомогательного метода описан в [2].

1.2 Нелинейный случай

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

Рассмотрим систему нелинейных дифференциальных уравнений

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(t_0) &= y_0, \\ t &\in [t_0, t_0 + \tau], \tau > 0, \\ y_0 &\in \mathbb{R}^n, y: [t_0, t_0 + \tau] \rightarrow \mathbb{R}^n, \\ f &\in \mathbb{R}^n \rightarrow \mathbb{R}^n. \end{aligned} \tag{1.7}$$

Для интегрирования воспользуемся методом (1.2), причем в отличие от линейного случая применение запишем в симметричном виде:

$$y(t_0 + \tau) \approx y_1 = y_0 + \tau \sum_{j=1}^s b_j f(t_0 + c_j \tau, Y_j), \tag{1.8}$$

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

что в векторной форме представимо как

$$Y = e \otimes y_0 + \tau (A \otimes I) F(t_0, Y), \tag{1.9}$$

$$y_1 = e \otimes y_0 + \tau (b^T \otimes I) F(t_0, Y).$$

Здесь $e = (1, \dots, 1), e \in \mathbb{R}^s, Y = (Y_1, \dots, Y_s)^T, F(t, Y) = (f(t + c_1 \tau, Y_1), \dots, f(t + c_s \tau, Y_s))^T$. Для приближенного вычисления решения системы воспользуемся методом утсановления: введем фиктивную переменную $\theta \in (0, +\infty]$ и рассмотрим вспомогательное дифференциальное уравнение

$$Y(\theta)' = \tilde{r}(Y(\theta)).$$

В развернутом виде уравнение установления для (1.7) имеет вид

$$Y(\theta)' = \tau (A \otimes I) F(t_0, Y(\theta)) - Y(\theta) + e \otimes y_0 = \tilde{r}(Y(\theta)). \tag{1.10}$$

Соответствующий ему процесс установления имеет вид аналогичный (1.8):

$$Y^{l+1} = \Phi(Y^l),$$

$$\Phi(Y) = Y + \omega \sum_{p=1}^{\sigma} \beta_p K_p(Y), \quad (1.9)$$

$$K_p(Y) = \tilde{r}(t_0, Y + \sum_{q=1}^{p-1} \alpha_{pq} K_q(Y)).$$

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 [2] рассуждения о конструировании вспомогательных методов.

Параметр ω полагаем таким, чтобы все собственные значения матрицы Якоби правой части уравнения (1.10) были по модулю меньше 1.

2 Спектральные свойства и скорость сходимости итерационного процесса

Процесс (1.8) представим в следующем виде:

$$k^{l+1} = R_{\sigma}(\omega G)k^l + P(\omega, G), \quad (2.1)$$

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 . 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 (1.8).

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

$$G = \tau(A \otimes J) - I. \quad (2.2)$$

По свойству кронекеровского произведения матриц, собственные значения v_{ij} матрицы G равны

$$v_{ij} = \tau \mu_j \lambda_i - 1, \quad (2.3)$$

Где μ_i – собственные значения матрицы A , λ_i – собственные значения матрицы J . То есть, над спектром исходной матрицы системы (1.1) 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 переобуславливания (глава 3.1.1).

Исследуем скорость сходимости итерационного процесса. Для этого проследим, как будет изменяться ошибка на l -ой итерации:

$$\varepsilon^l = k^* - k^l.$$

Учитывая (2.1), получим:

$$\varepsilon^l = R_\sigma(\omega G)\varepsilon^{l-1} \quad (2.4)$$

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

$$\varepsilon^l = \sum_{i=1}^N \varepsilon_i^{l-1} R(\omega G) \eta^i = \sum_{i=1}^N \varepsilon_i^{l-1} R(\omega v_i) \eta^i, \quad (2.5)$$

Здесь v_i – собственное значение, соответствующее η_i . Таким образом,

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

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 ωv_i близко к 0. Это значит, что компоненты ошибки, соответствующие малым величинам ωv_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.

3 Способы ускорения сходимости в процессе решения

3.1 Операция переобуславливания

Переобуславливание – это процесс преобразования условий задачи для ее более корректного численного решения. Переобуславливание обычно связано с уменьшением числа обусловленности задачи. Переобуславливаемая задача, как правило, затем решается итерационным методом.

В линейной алгебре и вычислительной математике P является переобуславливателем для матрицы A , если у матрицы $P^{-1}A$ число обусловленности меньше, чем у A . Также чаще говорят, что $T = P^{-1}$ это переобуславливатель, чем просто P , так как точное значение обычно требует больших затрат на вычисление. Поэтому под переобуславливанием часто понимают вычисление $T = P^{-1}$, точнее произведение вектора-столбца или матрицы векторов-столбцов на $T = P^{-1}$, что обычно выполняется сложными программными пакетами с использованием итерационных методов, где в конечном итоге не вычисляются точные значения ни для P , ни для $T = P^{-1}$.

Переобуславливание используется в итерационных методах при решении СЛАУ вида $Ax = b$, так как скорость сходимости для большинства итерационных линейных решателей увеличивается с уменьшением числа обусловленности в результате переобуславливания. Решатели с переобуславливанием обычно эффективнее, чем использование простых решателей, например, таких, как метод Гаусса 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.

Вместо решения исходной СЛАУ, описанной выше, можно решать переобусловленную систему:

$$AP^{-1}Px = b$$

которую можно решить через

$$AP^{-1}y = b,$$

где y удовлетворяет

$$Px = y$$

или решить переобусловленную слева систему:

$$P^{-1}(Ax - b) = 0$$

В результате получим то же решение, что и в исходной СЛАУ, до тех пор пока матрица переобуславливателя P невырожденная. Наиболее распространенным является переобуславливание слева. Целью переобуславливания является уменьшение числа обусловленности левой или правой переобусловленной СЛАУ $P^{-1}A$ или AP^{-1} соответственно. Переобусловленная матрица $P^{-1}A$ или AP^{-1} почти никогда не формируется отдельно. В место этого операция переобуславливания P^{-1} выполняется только над уже готовыми векторами, которые получаются в результате расчета итерационными методами.

Так как оператор P^{-1} применяется на каждом шаге итерационного линейного решателя, операция P^{-1} должна быть легко вычисляемой (по времени вычисления). Наиболее быстрым переобуславливателем в этом случае будет $P = I$, так как $P^{-1} = I$. Очевидно, что в результате работы такого переобуславливателя мы получим исходную СЛАУ. Другая крайность, выбор $P = A$, что даст $P^{-1}A = AP^{-1} = I$, при этом будет получено оптимальное число обусловленности 1, требующее одной итерации для того, чтобы решение сошлось. Тем не менее в этом случае $P^{-1} = A^{-1}$ и сложность вычисления переобуславливателя сравнима со сложностью решения исходной системы. Поэтому необходимо выбирать P где-то между двумя этими крайними случаями, пытаясь получить минимальное число итераций сохраняя легкость вычисления P^{-1} .

3.1.1 Применение операции переобуславливания для процесса установления

Чтобы избежать описанного в главе 2 нежелательного явления можно применить операцию переобуславливания, умножив систему (1.5) слева на $A^{-1} \otimes I$ (здесь $I \in \mathbb{R}^n \times \mathbb{R}^n$). Тогда система (1.5) примет вид:

$$\begin{aligned} (\tau I_s \otimes J - A^{-1} \otimes I_N)k + \tilde{g} &= 0, \\ \tilde{g} &= (\tilde{g}_1, \tilde{g}_2, \dots, \tilde{g}_s)^T, \tilde{g}_i = (A^{-1} \otimes I_N)(f(t_0 + c_i \tau) + Jy_0), i = 1, \dots, s, \\ k &= (k_1, k_2, \dots, k_s)^T, k_i \in \mathbb{R}^n \end{aligned} \quad (3.1)$$

Здесь I_s и I_N – единичные матрицы размерности $s \times s$ и $N \times N$ соответственно. Также стоит отметить, что точные решения (1.5) и (3.1) совпадают. По свойствам кронекеровского произведения, обобщенные значения матрицы

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

равны

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

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.[1]

3.2 Подавление медленно сходящихся 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

$$r^k = Ax^k - b = \sum_{i=1}^N \alpha_i^k \xi_i \quad (2.1)$$

- на $k+1$ -ом шаге

$$r^{k+1} = \sum_{i=1}^N \alpha_i^k R(\omega \lambda_i) \xi_i \quad (2.2)$$

Если λ_i велико, то вклад соответствующей компоненты в ошибку будет несущественен. Пусть m – номер собственного значения, соответствующего компоненте с наибольшим вкладом в ошибку.

Тогда

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

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

В предположении, что

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

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

$$\begin{aligned} r^k &\approx \alpha_m^k \xi_m, \\ r^{k+1} &\approx \alpha_m^k R(\omega \lambda_m) \xi_m \end{aligned}$$

Следовательно,

$$\frac{r_j^{k+1}}{r_j^k} \approx R(\omega \lambda_m), \quad j = \overline{1, N}. \quad (2.1)$$

Пусть \tilde{x} – точное решение, т.е.

$$\tilde{x} = x^{k+1} + \delta^{k+1}.$$

Тогда

$$0 = A\tilde{x} - b = A(x^{k+1} + \delta^{k+1}) - b = Ax^{k+1} + A\delta^{k+1} - b = r^{k+1} + A\delta^{k+1} = 0$$

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

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

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

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

Тогда,

$$R_\sigma(\omega\lambda_m) \approx 1 + R'_\sigma(0)\omega\lambda_m \approx \frac{r_j^{k+1}}{r_j^k}, \quad j = \overline{1, N} \quad (2.4)$$

Отсюда

$$\lambda_m \approx \left(\frac{r_j^{k+1}}{r_j^k} - 1 \right) / (\omega R'_\sigma(0)), \quad j = \overline{1, N} \quad (2.5)$$

Таким образом, получаем возможность уточнить текущее приближение следующим способом:

$$\tilde{x} \approx x^{k+1} + \frac{1}{\lambda_m} r^{k+1}. \quad (2.6)$$

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 заметим, что определение, когда именно нужно производить уточнение решения требует дополнительных исследований. В при реализации численного эксперимента уточнение проводилось на каждой 20-ой итерации процесса.

Полученный способ подавления медленно сходящихся компонент вектора решений исследуется с помощью численного эксперимента, постановка и результаты которого приведены в главе 3.

4 Численный эксперимент для линейной системы

Проведем численный эксперимент на примере задачи известной тестовой задачи *HIRES*. *HIRES* — эта химическая реакция с участием восьми реагентов, которая 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. Соответствующие уравнения имеют вид:

$$\begin{aligned}y_1' &= -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, \\y_2' &= 1.71y_1 - 8.75y_2, \\y_3' &= -10.03y_3 + 0.43y_4 + 0.035y_5, \\y_4' &= 8.32y_2 + 1.71y_3 - 1.12y_4, \\y_5' &= -1.745y_5 + 0.43y_6 + 0.43y_7, \\y_6' &= -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, \\y_7' &= 280y_6y_8 - 1.81y_7, \\y_8' &= -280y_6y_8 + 1.81y_7. \\y_1(0) &= 1, y_2(0) = y_3(0) = \dots = y_7(0) = 0, y_8(0) = 0.0057.\end{aligned}$$

А для выдачи были выбраны значения $t_{out} = 321.8122$ и 421.8122 .

Выше были представлен способ ускорения сходимости решения путем подавления ошибки медленно сходящихся компонент (п. 3.2). Интересно отследить вычислительные затраты в случаях с подавлением ошибки и без. Эксперимент будем проводить следующим образом: применим неявный метода *Радона* 3-го порядка в качестве базового метода и применим его для решения жесткой нелинейной задачи *HIRES* с использованием уточнения и без. Проведем серию таким экспериментов, задавая разную степень точности.

Как было сказано выше, в качестве базового метода используем следующие неявные методы 2-стадийный метод *Радона* 3-го порядка точности:

$\frac{1}{3}$		$\frac{5}{12}$	$-\frac{1}{12}$
1		$\frac{3}{4}$	$\frac{1}{4}$
		$\frac{3}{4}$	$\frac{1}{4}$

Для каждого эксперимента приведем общее количество итераций, произведенных

методом с уточнением и без. Количество стадий используемого явного вспомогательного метода $\sigma = 20$.

Результаты численного эксперимента проиллюстрированы в следующей таблице:

Точность	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
Количество итераций при решении без уточнения	5002	17507	46118	55719	47852	60144	75411
Количество итераций при решении с уточнением	2288	3766	5607	6950	10517	15834	27236
Ускорение решения	2,18	4,64	6,63	8,02	4,56	3,79	2,77

ЗАКЛЮЧЕНИЕ

В данной курсовой работе приведены общие принципы методов установления для решения жестких задач, рассмотрены способы ускорения сходимости итерационного процесса установления при решении систем дифференциальных уравнений. Проведен вычислительный эксперимент для нелинейной системы.

Таким образом, на основании проведенного вычислительный эксперимент на тестовой задаче HIREs, можно заключить, что практическая реализация способа ускорения итерационного процесса, приведенного в пункте (3.2) даёт ощутимое уменьшение вычислительных затрат при использовании методов, основанных на процессе установления. Однако стоит заметить, что с повышением требуемого порядка точности решения, выигрыш в трудоемкости несколько уменьшается, что хорошо видно при использовании метода РадоПА 3-го порядка точности в приведенном вычислительном эксперименте.

Даже несмотря на то, что описанный в работе способ требует дальнейшего исследования и улучшения, полученные результаты вычислительного эксперимента позволяют значительно улучшить сходимость.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бондарь И. В. Итерационные процессы установления для жестких задач. // Республиканский конкурс научных работ студентов высших учебных заведений Республики Беларусь
2. 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.
3. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. Пер. с англ. — М.: Мир, 1999.
4. Faleichik B. V. Explicit Implementation of Collocation Methods for Stiff Systems with Complex Spectrum // Journal of Numerical Analysis, Industrial and Applied Mathematics. Vol. 5
5. Axelsson, O. Iterative Solution Methods. //Cambridge University Press, 1996.
6. 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. 2011 г, Минск, Беларусь. С. 146-147
7. Фалейчик, Б. В. Вычислительные алгоритмы решения жестких задач на основе процессов установления / Б. В. Фалейчик // Труды института математики НАН Беларуси. - 2004. - Т. 12, № 1. - С. 45-48.
8. Бондарь И. В. Итерационные процессы установления для жестких линейных задач // Тр. 69-й ежегодной научной конференции студентов и аспирантов БГУ.
9. Фалейчик Б. В. Одношаговые методы численного решения задачи Коши : учеб.-метод. пособие / Б. В. Фалейчик. — Минск : БГУ, 2010.— 42 с.