О р д е н а   Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й   а к а д е м и и   н а у к

А.А.Белов, А.С.Вергазов, Н.Н.Калиткин

Контроль точности при численном интегрировании жестких систем

Москва — 2020

Белов А. А., Вергазов А. С., Калиткин Н. Н.

Контроль точности при интегрировании жестких систем

Абстракт оставили на потом.

Ключевые слова: потом

Belov A. A., Vergazov A. S., Kalitkin N. N.

Accuracy control in stiff systems integration

Abstract will be written later.

Key words: later

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проекты 09-01-12081-офи_м, 13-01-00493-а и 16-01-00158-а.

Оглавление

1. Методы численного интегрирования жестких систем

1.1. Длина дуги. Численное интегрирование задачи Коши для жестких систем обыкновенных дифференциальных уравнений (ОДУ) является одной из очень трудных задач вычислительной математики. Формально эта задача имеет следующий вид:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, t), \quad 0 \le t \le T, \quad \mathbf{u}(0) = \mathbf{u}^0. \qquad (1)$$

Здесь t – скаляр, $\mathbf{u}(t) = \{u_m(t), 1 \le m \le M\}$ и $\mathbf{f}(\mathbf{u}, \mathrm{t}) = \{f_m(\mathbf{u}, t), 1 \le m \le M\}$ – векторные функции, а M – размерность системы.

Задачу традиционно считают жесткой, если $T\|\mathbf{f}(\mathbf{u}, t)\| \gg 1$ хотя бы в части отрезка $t \in [0, T]$. Решению жестких задач посвящена обширная литература, наиболее подробный обзор которой дан в монографии [*Хайрер, Ваннер*].

Различают два типа прикладных задач. Первый тип – это так называемые большие, или многопроцессные, задачи. Они описываются системами уравнений в частных производных, к которым подключена одна или несколько систем ОДУ. Примером могут служить задачи горения и взрыва. В них процесс горения, то есть реакции химических веществ, описывается системой уравнений химической кинетики; это система ОДУ. Выделяющееся тепло приводит к движению вещества. Это движение описывается уравнениями газодинамики, то есть системой уравнений в частных производных.

Расчет уравнений в частных производных гораздо более трудоемок, чем решение ОДУ. Поэтому в больших задачах шаг по времени τ определяется требованиями методов решения уравнений в частных производных. Этот же шаг τ вычислитель вынужден использовать для решения сопутствующей системы ОДУ. В этом случае естественным аргументом для решения задачи (1) является время t.

Второй класс задач содержит только систему ОДУ. Для этого класса задач оказывается выгоднее выбрать другой аргумент – длину дуги интегральной кривой в многомерном пространстве. Длина дуги $l(t)$ определяется соотношением

$$dl^2 = dt^2 + d\mathbf{u}^2 \equiv dt^2 + \sum_{m=1}^{M} du_m^2 = (1 + \sum_{m=1}^{M} f_m^2) dt^2. \qquad (2)$$

Если выбран новый аргумент l, то старый аргумент t становится его функцией:

$$t(l) \equiv u_0(l); \tag{3}$$

таким образом, число неизвестных функций становится $M+1$. Формально новую функцию можно включить в систему (1), добавив туда уравнение $du_0/dt \equiv dt/dt = 1$.

Напомним, что расширенная система ОДУ содержит в правых частях аргумент *t*. Его в правых частях можно заменить на новую функцию: $t \equiv u_0$. Тогда расширенная система формально не будет содержать аргумента *t* в правых частях, то есть станет автономной.

Перейдем к новому аргументу *l*. Для этого в расширенной системе (1) заменим *dt* на *dl* с помощью соотношения (2). Получим следующую систему размерности $M+1$:

$$\frac{du_m}{dl} = \frac{f_m(u_0, u_1, \ldots, u_M)}{\sqrt{\sum_{m=0}^{M} f_m^2(u_0, u_1, \ldots, u_M)}}, \quad 0 \le m \le M; \quad f_0(u_0, u_1, \ldots, u_M) \equiv 1. \tag{4}$$

При аргументе *l* сумма квадратов правых частей (4) равна 1, то есть норма правой части никогда не бывает большой. Это облегчает численное интегрирование системы. Такой прием полезен даже для нежестких ОДУ, а для жестких он кардинально облегчает решение задачи. Переход к длине дуги и различные преимущества этого метода подробно описаны в монографии [*Кузнецов*]. Заметим, что правые части (4) не содержат аргумента *l*, то есть эта система является автономной.

Переход к длине дуги практически не увеличивает трудоемкость одного шага численных расчетов. Практически всегда основное время расчетов уходит на вычисление правых частей f_m, которые могут быть достаточно сложными функциями своих аргументов. Переход от (4) к (1) включает лишь несколько дополнительных арифметических операций, трудоемкость которых невелика. Размерность системы увеличивается на 1, что так же мало существенно для прикладных задач, где *M* обычно довольно велико.

1.2. Выбор шага. Расчет с постоянным шагом по времени τ или по длине дуги *h* обычно невыгоден. Шаг целесообразно уменьшать там, где решение быстро меняется, то есть правые части ОДУ велики. На участках слабого изменения решения шаг можно увеличивать. В [1] подробно описаны алгоритмы автоматического выбора шага, принятые в мировой литературе. Традиционно используют два основных метода выбора шага. В первом методе каждый шаг выполняют по некоторой схеме $(p+1)$-го порядка точности, в которую вложена схема *p*-го порядка точности. Результат вложенной схемы берут в качестве ответа. По разности результатов двух схем выбирают величину следующего шага. Во втором методе шаг τ рассчитывают повторно, разбив его на два шага величиной $\tau/2$. По разности этих расчетов вычисляют новый шаг.

На основе этих методов написано много пакетов программ. Большинство из них хорошо работают на нежестких задачах. Однако проверка этих пакетов на тестах с известными точными решениями показывает, что реальная точность расчетов лишь по порядку величины близка к запросу пользователя. Она обычно оказывается в несколько раз хуже или лучше. Поэтому в расчетах прикладных задач, где ответ неизвестен, пользователь не может быть вполне уверен в достижении требуемой точности.

Намного хуже ситуация для случая жестких задач. Для них расчеты на тестах показали [*литература*], что реальная точность может быть в $100-1000$ раз хуже заявленной. Это относится даже к таким тщательно выверенным программам, как пакеты Гира или программы Дормана-Принса dopri5. Кроме того, на жестких задачах возможны “срывы” шага: иногда на участках слабо меняющегося решения программа без видимых причин уменьшает шаг в $100-1000$ раз. Затем шаг постепенно увеличивается, но снова срывается; это может повторяться много раз. В [*Пошивайло, НН, Белов*] был предложен принципиально другой алгоритм автоматического выбора шага. Он основан на использовании длины дуги и кривизны κ интегральной кривой в многомерном пространстве. Интуитивно понятно, что чем больше кривизна κ, тем меньше должен быть шаг h. Но каким должен быть алгоритм, связывающий эти две величины?

1.3. Оптимальный шаг. Напомним определение кривизны кривой ***u(t)*** в $(M+1)$-мерном пространстве с координатами $\{t, u_1, u_2, \ldots, u_M\}$. Касательная к этой кривой определяется через производную $d\mathbf{u}/dt$. Деля на длину этого вектора, получим единичный вектор направления касательной:

$$\mathbf{n} = \frac{d\mathbf{u}}{dt} \Big/ \left\| \frac{d\mathbf{u}}{dt} \right\|_2 . \tag{5}$$

Кривизна определяется как производная вектора направления касательной по длине дуги:

$$\kappa = \frac{d\mathbf{n}}{dl}, \kappa = \|\kappa\|_2 . \tag{6}$$

Добавлено примечание ([AV1]): Как в MathType сделать каппа жирным?

Таким образом, кривизна является вектором даже в случае плоской кривой, то есть одного ОДУ; это радиус-вектор окружности, имеющей касание второго порядка с кривой ***u(t)***. Наряду с этим говорят о скалярной кривизне *κ*, которая равна величине радиуса этой окружности.

В ранних работах [*Пошивайло*] с помощью многих численных экспериментов была подобрана неплохая эвристическая закономерность: $h \sim \kappa^{-0.5}$. Однако в пакетах прикладных задач был использован постоянный шаг по длине дуги и несколько семейств разностных схем с порядком точности от 1 до 4. Эти пакеты позволили провести расчеты некоторых прикладных задач, например – образование окислов фосфора и серы при горении различных топлив.

Задача теоретического нахождения зависимости $h(\kappa)$ крайне сложна. Из общих соображений понятно, что результат должен зависеть от того, какая именно разностная схема используется для интегрирования системы ОДУ. Однако удалось найти такой случай [*диф. ур.*], для которого теоретически обосновывается формула выбора оптимального шага. Пусть схема интегрирования имеет точность $\mathrm{O}(h)$; это может быть явная или неявная схема Эйлера или явно-неявная L1-устойчивая схема Розенброка. Тогда

$$h_{opt}(\kappa) = const \cdot \kappa^{-2/5} \tag{7}$$

Качественный вид этой формулы совпадает с ранее найденным эвристическим видом, а оптимальный показатель степени лишь слабо отличается от эвристического.

Как часто оказывается, “чистая” теоретическая формула (7) для своего практического применения требует “кухонных” поправок.

Во-первых, ясно, что при $\kappa = 0$ она дает $h = \infty$. Формально это правильно: если на некотором участке кривой $\kappa = 0$, этот участок есть прямая, то есть ***u(t)*** является линейной функцией. А для линейной функции любая численная схема дает точный ответ при сколь угодно большом шаге.

На практике $\kappa = 0$ лишь в отдельных точках кривой (например, в точках перегиба), а попадание узла расчетной сетки именно в эту точку имеет нулевую вероятность. Однако возможно попадание счетного узла в малую окрестность точки перегиба и получение неприемлемо большого шага *h*. Надо ввести такую поправку, чтобы в любом случае число интервалов было не меньше некоторого разумного $N_{\min}$.

Во-вторых, надо дать разумное определение константы в (6). В нее должна быть включена некоторая интегральная нормировка по длине дуги, обеспечивающая желательное количество интервалов сетки $N_{\max}$.

Исходя из этого, в [*дифф ур*] был предложен следующий алгоритм. Пусть требуется решить задачу (4) на отрезке $[0, L]$ с априорно заданными $N_{\max}$, $N_{\min}$. Несколько видоизменим формулу (7) с учетом сделанных замечаний:

$$h(l)=\left[\frac{N_{\min}}{L}+\frac{N_{\max}\kappa^{2/5}}{\int_0^L \kappa^{2/5}(l')dl'}\right]^{-1}. \quad (8)$$

В точках с очень малой кривизной $\kappa(l)\approx 0$ она дает $h(l)\approx L/N_{\min}$. В точках с очень большой кривизной формула (8) переходит в (7); при этом константа такова, что $\int h^{-1}(l)dl \approx N_{\max}$. При любых $\kappa(l)$ не может получиться $h=0$ или $h=\infty$. Это свидетельствует о разумности формулы (8).

1.4. Двухэтапная стратегия. Численный расчет может называться хорошим, если он гарантирует пользователю требуемую точность. Для нежестких задач, когда в качестве аргумента выбирают время *t*, этого нетрудно добиться. Расчет до заданного момента *T* проводят на равномерной сетке сначала с некоторым умеренным числом шагов *N*. Затем увеличивают *N* в 2 раза и проводят новый расчет. При этом узлы t_n первой сетки точно совпадают с четными узлами удвоенной сетки. В совпадающих узлах находят разность двух сеточных решений $\mathbf{u}(t)$ и вычисляют норму разности (обычно это нормы L2 или C). В этом случае для оценки погрешности применим метод Ричардсона [*Ричардсон 28г.; Марчук, Шайдуров; книга НН 78г.; Хайрер, Ваннер; Калиткин, Альшины 2005г.*]. Если заданная точность не будет достигнута, то процедуру удвоения сетки повторяют.

Однако для жестких задач равномерная сетка непригодна. Для того, чтобы шаг равномерной сетки стал достаточно малым на участках быстрого изменения решения, нужны огромные числа узлов *N*. Даже переход к аргументу *l* не позволяет использовать равномерные сетки.

Существуют так называемые квазиравномерные сетки [*книга НН, книга НН и Альшиных*]. Для них разработана процедура удвоения *N*, позволяющая пользоваться методом Ричардсона. Но для построения такой сетки нужно заранее знать поведение всей функции $\mathbf{u}(l)$ и строить густую сетку там, где функция быстро меняется. Однако до начала решения задачи мы не знаем, где расположены такие участки. Поэтому в [*лит-ра*] была предложена двухэтапная процедура сгущения сеток. С учетом новейших работ она формулируется следующим образом.

Первый этап. Для начала расчета выбираем некоторые разумные значения $N_{\min}$ и $N_{\max}$, входящие в формулу шага (8). В эту формулу входят так же *L* и интеграл. До начала расчетов они неизвестны, поэтому задаются их правдоподобные оценки. После этого проводят расчет с выбором шага согласно (8) и продолжают его до тех пор, пока $u_0(t_n)\equiv t_n$ не превысит значение *T*. Полученный номер узла есть *N* для построенной сетки l_n. При этом автоматически вычисляется значение полной длины дуги для данной сетки

$L = l_N$. Попутно в ходе расчета вычисляют интеграл в (8) по какой-нибудь квадратурной формуле; допустима даже формула левых прямоугольников.

Найденное в этом расчете значение N может не совпадать с исходным значением $N_{\max}$. Разумеется, вычисленное значение L и интеграла в (8) не будут совпадать с теми величинами, которые использовались для начала расчета.

Затем удвоим значения $N_{\min}$, $N_{\max}$ и построим сгущенную примерно вдвое сетку. В этом расчете будем использовать значения интеграла и L, найденные на предыдущей сетке. Практика показывает, что при этом N окажется примерно вдвое большим, чем на предыдущей сетке. Однако новые значения интеграла и L могут сильно отличаться от предыдущих значений, поскольку на первой сетке они были выбраны произвольно. Достаточно очевидно, что новая сетка будет примерно вдвое подробнее предыдущей, однако ее четные узлы не будут совпадать с узлами предыдущей сетки. Поэтому оценка точности по правилу Ричардсона невозможна.

Снова удвоим $N_{\min}$, $N_{\max}$ и построим третью сетку. На третьей сетке значение N уже почти удваивается по сравнению со второй сеткой, а значения интеграла и L будут близки к соответствующим величинам второй сетки. Значения четных узлов третьей сетки так же будут существенно ближе к узлам второй сетки. Однако правило Ричардсона применять по-прежнему нельзя.

Эту процедуру удвоения сетки повторяем до тех пор, пока четные узлы новой сетки не станут достаточно близкими к узлам предыдущей сетки. Различные критерии близости сеток сравнивались в [*лит-ра*]; наилучшим был признан следующий критерий. Обозначим шаги исходной сетки через $h_n = l_n - l_{n-1}, 1 \le n \le N$. Шаги удвоенной сетки обозначим через $\hat{h}_n, 1 \le n \le \hat{N}$; на практике $\hat{N} \approx 2N$, хотя точного равенства может не быть. Шагу h_n в удвоенной сетке соответствует сумма шагов $\hat{h}_{2n-1} + \hat{h}_{2n}$. Будем считать сетки близкими, если

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\sqrt{\xi_n} - \frac{1}{\sqrt{\xi_n}}\right)^2} \le \eta, \quad \xi_n = \frac{\hat{h}_{2n-1} + \hat{h}_{2n}}{h_n}; \tag{9}$$

разумную величину η надо подбирать на основе практических расчетов. При этом значения интеграла и L на этих сетках будут совпадать с приемлемой точностью. При выполнении критерия (9) первый этап заканчивается.

Второй этап. Будем считать, что последняя сетка первого этапа уже достаточно хорошо адаптирована к участкам быстрого изменения решения. Поэтому дальнейшие сгущения сеток будем производить так, чтобы можно было применять метод Ричардсона для оценки погрешности. Для этого надо проводить сгущения так, чтобы последовательность удвоенных сеток была квазиравномерной. Простейший способ такого сгущения имеет следующий вид.

Пусть исходная сетка имеет узлы $l_n, 0 \le n \le N$ и шаги $h_n = l_n - l_{n-1}$. Пусть интервал сетки $[l_{n-1}, l_n]$ является внутренним: $2 \le n \le N-1$. Тогда узлы этого интервала являются четными узлами новой сетки $\hat{l}_{2n-2}$ и $\hat{l}_{2n}$, а шаг h_n делится на два шага новой сетки $\hat{h}_{2n-1}$ и $\hat{h}_{2n}$ по следующим формулам:

$$\hat{h}_{2n-1} = h_n \frac{\sqrt[4]{h_{n-1}}}{\sqrt[4]{h_{n-1}} + \sqrt[4]{h_{n+1}}}, \quad \hat{h}_{2n} = h_n \frac{\sqrt[4]{h_{n+1}}}{\sqrt[4]{h_{n-1}} + \sqrt[4]{h_{n+1}}}, \quad 2 \le n \le N-1. \tag{10}$$

Для левого граничного интервала шаг h_1 делится по формулам

$$\hat{h}_1 = h_1 \frac{\sqrt{h_1}}{\sqrt{h_1} + \sqrt{h_2}}, \quad \hat{h}_2 = h_1 \frac{\sqrt{h_2}}{\sqrt{h_1} + \sqrt{h_2}}. \tag{11}$$

Правый граничный шагh_Nделится следущим образом:

$$\hat{h}_{2N-1} = h_N \frac{\sqrt{h_{N-1}}}{\sqrt{h_{N-1}} + \sqrt{h_N}}, \quad \hat{h}_{2N} = h_N \frac{\sqrt{h_N}}{\sqrt{h_{N-1}} + \sqrt{h_N}}. \tag{12}$$

При таких формулах сгущения сетки число интервалов точно удваивается, а полная длина дуги L не меняется.

Можно доказать, что описанный способ удвоения сетки порождает последовательность квазиравномерных сеток. На такой последовательности правомерно однократное применение метода Ричардсона: по каждой паре соседних сеток можно находить разности функций в совпадающих узлах $l_n = \hat{l}_{2n}$, вычислять нормы этих разностей и определять асимптотически точное значение погрешности по известному порядку точности формулы интегрирования.

Однако этот способ имеет одно ограничение: нельзя пользоваться рекуррентным методом Ричардсона, то есть повышать порядок точности, используя 3 и более соседние сетки.

2. Апробация на тестах

2.1. Требования к тестам. Общепринятым способом апробации различных численных методов является расчет тестовых задач. Хороший тест должен удовлетворять нескольким требованиям. Во-первых, он должен содержать типичные трудности, на преодоление которых ориентирован исследуемый численный метод. Во-вторых, у него должно существовать легко реализуемое точное решение. Обычно под этим подразумевают, что точное решение при любых значениях аргументов легко вычисляется с любой требуемой точностью. Реально для этого нужно, чтобы решение достаточно просто выражалось через элементарные функции; запись через специальные функции нежелательна, так как их реализация с любым требуемым числом значащих цифр далеко не всегда доступна. В-третьих, желательно, чтобы тест содержал один или несколько параметров, которыми можно регулировать его жесткость и другие качественные свойства.

При наличии такого теста апробация метода несложна. Проводят расчет исследуемым методом, причем этот метод выбирает сетку по своим естественным правилам. В узлах этой сетки вычисляют точное решение и непосредственно находят точное значение погрешности, то есть разности между сеточным и точным решением. При этом можно вычислить любую норму погрешности. Это будет достоверная оценка.

Задачи, в которых решение известно только в отдельных реперных точках, непригодны в качестве теста. Во-первых, они не позволяют найти нормы ошибки. Во-вторых, для сравнения с ними приходится проводить решение на таких сетках, некоторые узлы которых попадают в реперные точки. Для большинства алгоритмов такое построение сеток неестественно.

Для обыкновенных дифференциальных уравнений с аргументом t сравнительно легко конструируются различные тесты, в том числе для случаев высокой и сверхвысокой жесткости. Однако при переходе к аргументу l проблема существенно усложняется. Недостаточно, чтобы в элементарных функциях выражалось $\mathbf{u}(t)$. Нужно наличие решения в элементарных функциях так же для $\mathbf{u}(l)$. При этом все формулы не только прямого вычисления решения, но и вычисления обратных функций так же должны выражаться в элементарных функциях, причем достаточно просто. В противном случае, вычисление погрешностей при переходе от одного аргумента к другому будет практически невозможным.

До сих пор такие тесты не были построены. Нам удалось сконструировать два требуемых теста.

2.2. Гиперболический тест. Рассмотрим тестовое уравнение, содержащее один свободный параметр:

$$\frac{du}{dt} = f(u) \equiv sh(\lambda u), \quad \lambda > 0, \quad u(0) = u^0 > 0. \tag{13}$$

Точное решение этой задачи имеет следующий вид:

$$u(t) = \frac{1}{\lambda} \ln \frac{1+B(t)}{1-B(t)}, \quad B(t) = e^{\lambda t} th(\lambda u^0 / 2). \tag{14}$$

Подставляя выражение (14) в формулу (2) и интегрируя, получим выражение для длины дуги:

$$l(t) = \frac{1}{\lambda} \ln \frac{sh[\ln(1+B(t)) - \ln(1-B(t))]}{sh(\lambda u^0)}. \qquad (15)$$

Далее увидим, что жесткость задачи (13) быстро увеличивается с ростом λ. В задачу (13) нетрудно ввести еще один параметр — множитель α в правой части. Однако это не представляет интереса, так как заменой $\tilde{u} = u/\alpha$ и $\tilde{\lambda} = \alpha\lambda$ задача сводится форме (13).

Исследуем качественный вид решения (14). Функция *u(t)* положительна и монотонно возрастает. Она определена на конечном отрезке, поскольку

$$u(t) \to +\infty \text{ при } t \to t_* = \frac{1}{\lambda} \ln(cth(\lambda u^0/2)) > 0. \qquad (16)$$

Длина дуги при этом так же неограниченно возрастает: $l(t) \to +\infty$ при $t \to t_*$. Качественный вид решения *u(t)* изображен на рис. 1. Решение имеет продолжение влево на полуось $-\infty < t \le 0$.

Рис. 1. (Подпись: точное решение (14); <символ кружок> - начальная точка)

Перейдем в уравнении (13) к переменной *l*. Учтем, что $1 + f^2(u) = ch^2(\lambda u)$. Тогда форма записи (4) примет следующий вид:

$$\frac{du}{dl} = th(\lambda u), \quad \frac{dt}{dl} = \frac{1}{ch(\lambda u)}, \quad u(0) = u^0, \quad t(0) = 0, \quad 0 \le l \le +\infty. \qquad (17)$$

Для первого уравнения системы нетрудно написать точное решение:

$$u(l) = \frac{1}{\lambda} \ln(A(l) + \sqrt{A^2(l)+1}), \quad A(l) = e^{\lambda l} sh(\lambda u^0), \quad 0 \le l \le +\infty. \qquad (18)$$

Время-функция так же явно выражается через длину дуги:

$$t(l) = \frac{1}{\lambda} \ln \frac{th[(1/2)\ln(A(l) + \sqrt{A^2(l)+1}}{th(\lambda u^0/2)}, \quad 0 \le l \le +\infty. \qquad (19)$$

Качественный вид функции *u(l)* изображен на рис. 2. Интересно сопоставить эту зависимость с рис. 1. На рис. 1 решение резко возрастает при $t \to t_*$. Тем самым, численный расчет на этом участке возможен только очень малым шагом τ. Однако на рис. 2 этому участку соответствует правая часть графика, где наклон $du/dl \approx 1$. Тем самым, при аргументе l эту область решения можно проходить большим шагом h.

Рис. 2. (Подпись: точное решение (18), (19))

Приведем еще некоторые явные формулы, из которых особенно важно выражение кривизны:

$$t(u) = \frac{1}{\lambda} \ln \frac{th(\lambda u/2)}{th(\lambda u^0/2)}, \tag{20}$$

$$l(u) = \frac{1}{\lambda} \ln \frac{sh(\lambda u)}{sh(\lambda u^0)}, \tag{21}$$

$$\kappa = \frac{u_{tt}}{(1+u_t^2)^{3/2}} = \frac{\lambda sh(\lambda u)}{ch^2(\lambda u)}. \tag{22}$$

Кривизна немонотонно меняется вдоль интегральной кривой на рис. 1 или рис. 2. В левой части обоих графиков она возрастает, затем достигает максимума при условиях

$$\max \kappa = \frac{\lambda}{2}, \quad u_{extr} = \frac{1}{2\lambda} \ln\left(3 + 2\sqrt{2}\right), \quad t_{extr} = \frac{1}{2\lambda} \ln\left(\frac{1}{3+2\sqrt{2}} \frac{ch(\lambda u^0)+1}{ch(\lambda u^0)-1}\right). \tag{23}$$

и затем монотонно убывает до 0 в правой части графиков. Отсюда хорошо видна связь параметра λ с максимальной кривизной интегральной кривой и жесткостью задачи.

Данный тест является уникальным. Во-первых, в нем все значимые величины выражаются друг через друга с помощью несложных комбинаций элементарных функций. Во-вторых, выбирая большие значения λ, можно сделать задачу сколь угодно жесткой.

Трудность задачи. Трудность определяется тем числом шагов *N*, которое необходимо сделать при оптимальном выборе шага. Из формулы оптимального шага (8) нетрудно понять, что это число шагов пропорционально входящему в нее интегралу

$$I = \int_0^L \kappa^{2/5}(l')dl' \quad (24)$$

Величина этого интеграла зависит не только от кривизны *κ(l),* но и от отрезка интегрирования $[0, L]$. Такой критерий определения трудности можно применять для любых тестовых или прикладных задач.

Для гиперболического теста зависимость *κ(l)* можно получить, подставляя формулу (18) в формулу (22):

$$\kappa(l) = \frac{\lambda A(l)}{1 + A^2(l)} = \frac{\lambda e^{\lambda l} sh(\lambda u^0)}{1 + e^{2\lambda l} sh^2(\lambda u_0)} \quad (25)$$

Подстановка величины (25) в интеграл (24) и замена переменных $\zeta = e^{(2/5)\lambda l}(sh(\lambda u^0))^{2/5}$ дают

$$I = \int_0^L \frac{\lambda^{2/5} e^{(2/5)\lambda l} sh^{2/5}(\lambda u^0)}{(1 + e^{2\lambda l} sh^2(\lambda u^0))^{2/5}} dl = \frac{5}{2}\lambda^{-3/5} \int_{\zeta_0}^{\zeta_L} \frac{d\zeta}{(1+\zeta^5)^{2/5}}. \quad (26)$$

Пределы интегрирования $0 < \zeta_0 < \zeta_L < +\infty$, поэтому последний интеграл в (26) не превышает 2. С учетом множителя $\lambda^{-3/5}$ перед интегралом видно, что данный тест нельзя назвать по настоящему трудным. Однако более хорошего теста с такими уникальными выражениями через элементарные функции найти не удалось.

2.3. Тригонометрический тест. Упомянем еще один тест, в котором явно вычисляется длина дуги и все имеющиеся величины выражаются через элементарные функции как аргумента *t*, так и аргумента *l*. Приведем набор формул этого теста (текстовые пояснения опускаем, поскольку они аналогичны гиперболическому тесту):

$\frac{du}{dt} = tg(\lambda u)$(27)

(лямбда > 0)

$u(t) =$(28)

(29)

(30)

(31)

(32)

(33)

(34)

Качественный вид решения при аргументе *t* изображен на рис. 3, а при аргументе *l* – на рис. 4. Поскольку мы положили$\lambda > 0$, то решение u(t) монотонно возрастает на конечном промежутке времени от 0 до t_* = lambda^-1… При этом функция достигает значения u_* тождественно = u(t_*) = pi / (2 lambda), а производная в этой точке u_t(t_*) = +inf. Решение не имеет продолжения за точку t_*.

Если u^0 → 0, то t_* → +inf и полная длина дуги l_* → +inf, хотя u_* остается конечным. Последнее объясняется тем, что при u^0 → 0 правая часть f(u^0) → 0, и движение вдоль интегральной кривой начинается очень медленно, так что полное время движения стремится к бесконечности; соответственно стремится к бесконечности длина дуги.

Отметим, что можно рассмотреть случай lambda < 0. В этом случае решение u(t) будет монотонно убывать, асимптотически стремясь к 0 при t→ +inf. Это решение несколько напоминает известный тест Далквиста.

Тригонометрический тест не позволяет получить задачи произвольно высокой жесткости. Поэтому он менее интересен для тестирования методов, чем гиперболический тест.

3. Численные расчеты.

Дальше:

~~Послать Белову чтобы он внес правки и поставия лит-ру.~~

~~Формулы тестов с синусом и тангенсом.~~

Рисунки: с разными лямбда ERK1, ERK2, ERK4. + для лямбда 10^4 ERK1+ERK2+ERK4 из одной точки.

Графики решения для трех лямбд в одном масштабе. Тоненькой линией вся кривая, маркерами или жирной – расчетный участок, где кривизна от 1 до 1.

Решение в координатах u(t), u(l)?, t(l)?

Библиографический список

ϰ
ϰ