УДК 519.622.2

НАСТРОЙКА СОСТАВНОГО АЛГОРИТМА РЕШЕНИЯ ЖЕСТКИХ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

С.О. Травин
ФИЦ ХФ РАН им. Н.Н. Семенова
Россия, 119991, Москва, ул. Косыгина, 4
E-mail: TravinSO@Yandex.ru

П.И. Михеев
МГТУ им. Н.Э. Баумана (НИУ)
Россия, 105005, Москва, 2-я Бауманская ул., 5, стр. 1
E-mail: petr_miheev@mail.ru

О.Б. Громов
ВНИИНМ им. А.А. Бочвара
Россия, 123098, Москва, ул. Рогова, д. 5а.
E-mail: olbgromov@bochvar.ru

А.А. Быков
ООО «Русатом МеталлТех»
Россия, 123098 Москва, ул. Рогова, д. 5а.
E-mail: andalekbykov@rosatom.ru

Ключевые слова: дифференциальное уравнение, химическая кинетика, прямая задача, жесткие системы, метод Розенброка, метод Адамса.

Аннотация: Представленная работа содержит в себе описание построения и отладки составного алгоритма интегрирования жестких систем дифференциальных уравнений. Предлагается на «разгонном участке», задающем стартовые значения для дальнейшего применения многошаговых алгоритмов, использовать метод Розенброка, использующий неявную схему счета, подразумевающую вычисление и инвертирование якобиана на каждом шаге. При этом временные затраты на подготовку этого этапа сравнительно невелики, поскольку разгонных шагов в методе Адамса всего четыре.

1. Введение

В химической кинетике особый и ниспадающий интерес вызывают, т.н. жесткие системы уравнений, в которых реакционная способность различных компонентов, а значит и их квазистационарная концентрация могут отличаться на много порядков.

Существуют различные определения жесткости системы, но мы будем придерживаться точки зрения Ч. Гира [1]: «Хотя общепринято говорить о «жестких дифференциальных уравнениях», какое-либо уравнение само по себе не является жестким, жесткой может быть конкретная задача Коши для этого уравнения, причем в определенных областях, размеры которых зависят от начальных значений и допустимой погрешности». И в этом смысле все сложности построения алгоритмов интегрирования таких систем сводятся не столько к выбору расчетной схемы, сколько к выбору оптимального шага разностной схемы, достаточно малого, чтобы обеспечить устойчивость и необходимую точность решения, при этом не превращая его в томительное ожидание прохождения миллиардов выполнений внутреннего цикла.

В монографии [2] отмечено, что несмотря на рост быстродействия ЭВМ, сложность задач, встречающихся в практике, опережает развитие вычислительной техники, поэтому проблема создания эффективных численных методов решения задачи Коши для жестких систем остается по-прежнему актуальной.

Десятилетия исследований продемонстрировали, что универсального алгоритма, пригодного для решения любого класса жестких задач, создать еще никому не удалось и, по-видимому, его просто не существует. И в каждом конкретном случае приходится выбирать компромисс между потерей машинного времени и потерей точности и/или устойчивости решения. Стало понятно, что основная проблема выбора состоит во взвешивании преимуществ и недостатков явных и неявных разностных схем.

Явные схемы отличаются очевидной простотой, не требуют ни решения систем уравнений, ни итерационных процедур, но на этом все их «преимущества» заканчиваются. Единственным средством борьбы с жесткостью системы остается лишь более мелкое дробление шага, что в ряде случаев остается приемлемым, учитывая вычислительную дешевизну каждого шага.

Напротив, при реализации L-устойчивых неявных численных схем на каждом шаге решается линейная система алгебраических уравнений. При большой размерности исходной задачи общие вычислительные затраты фактически полностью определяются временем декомпозиции Якобиана. С другой стороны, количество элементарных шагов в таких схемах может оказаться на порядки меньше, чем в случае явных схем.

2. Выбор модели и тактика построения сольвера

Для тестирования возможностей и выявления проблемных мест того или иного метода прежде всего необходимо выбрать подходящую модель. Подчеркнем еще раз, что, по нашему мнению, поиск универсального идеального метода в отрыве от решаемой задачи представляется бессмысленным, поскольку его просто не существует. По этой причине мы выбрали модель, зарекомендовавшую себя своей жесткостью и лишь затем подбирали удачный способ ее решения.

Надо заметить, что изобилие публикаций, посвященных рассмотрению тех или иных аспектов решения прямой задачи химической кинетики сослужило дурную службу и заметно затруднило возможность нахождения реально работающих рекомендаций, способствующих построению эффективной методики интегрирования и воплощению ее в кодах в той или иной программной среде. Напротив, преобладающее число литературных источников, причем не только учебников и монографий, но и оригинальных статей в периодических научных журналах, содержат в себе столь общий набор формул, что его имплементация в кодах практически обречена на неуспех. Если в поисковой системе Google Scholar задать запрос «direct problem of chemical kinetics», то полученный ответ будет содержать свыше четырех миллионов литературных ссылок, из которых самодостаточными для самостоятельной программной реализации оказываются единицы. Так или иначе, нам удалось найти пригодные для практического воплощения литературные источники лишь с некоторым трудом. Также для целей настоящей работы оказались малопригодны пособия, апеллирующие к программным средам с уже встроенными сольверами, типа Matlab, MathCAD, CHEMCAD и т.д.

Модель химических реакций Робертсона является одним из первых и самых популярных примеров «жесткой» системы обыкновенных дифференциальных уравнений; она впервые опубликована в 1966 году [3] и в химической нотации имеет вид:

$$A \overset{k_1}{\rightarrow} B \qquad k_1 = 0.04, \qquad W_1 = k_1[A]$$
$$B + B \overset{k_2}{\rightarrow} C + B \qquad k_2 = 3 \cdot 10^7, \qquad W_2 = k_2[B][B]$$
$$B + C \overset{k_3}{\rightarrow} A + C \qquad k_3 = 1 \cdot 10^4, \qquad W_3 = k_3[B][C]$$

Вот как о ней высказался сам Робертсон: «Когда уравнения представляют поведение системы, содержащей ряд быстрых и медленных реакций, прямое интегрирование этих уравнений становится затруднительным».

Модель Робертсона, описывается дифференциальными уравнениями

$$\dot{y}_1 = -k_1 y_1 + k_2 y_2 y_3, \qquad y_1 \equiv [A]$$
$$\dot{y}_2 = \ \ k_1 y_1 - k_2 y_2 y_3 - \ \ k_3 y_2 y_2, \qquad y_2 \equiv [B]$$
$$\dot{y}_3 = \ \ k_3 y_2 y_2 = k_3 y_2^2, \qquad y_3 \equiv [C]$$

Эта задача - самая жесткая из известных (мера жесткости $М_ж = 10^{15}$). Трудность ее решения в постановке Робертсона обусловлена не только сосуществованием быстрых и медленных реакций, но также и большим интервалом интегрирования (от 0 до 10^{11} с), на протяжении которого размер шага увеличивается более чем на 14 порядков. Большинство стандартных сольверов, особенно основанных на явном методе вычислений, оказались малопригодными для решения этой задачи.

Очевидно, напрашивается использование неявного метода Розенброка, однако, такой подход требует вычисления и инверсии Якобиана на каждом шаге и, кроме того, требуется многократное вычисление правой части f в промежуточных точках, которые при переходе к новой точке повторно не используются, что ожидаемо приводит к относительно медленной реализации решения. Напротив, можно ожидать высокой производительности от многошаговых методов Адамса, но для них неизбежен сравнительно медленный «разгонный» участок для накопления необходимого числа шагов, определяемого порядком метода. В итоге мы пришли к композитному решению, в котором 5-шаговому методу Адамса предшествуют четыре шага по методу Розенброка.

3. Алгоритм Розенброка

В работе [4] разработан L-устойчивый трехстадийный метод Розенброка третьего порядка точности для решения жестких задач, для которого построено неравенство контроля точности вычислений, основанное на оценке аналога глобальной ошибки, которая осуществляется с привлечением ранее вычисленных стадий, что позволяет выбирать величину шага интегрирования фактически без увеличения вычислительных затрат. Во избежание путаницы мы изменили термин «Шаг» в описании алгоритма на термин «Процедура», чтобы отличать его от шага интегрирования. Кроме того, мы до предела упростили набор процедур (5) – (10), связанных с регулированием шага. В общем случае это существенное искажение метода, но применительно к тестовой задаче мы убедились, что оно не влечет неприятных последствий.

Так на каждом n-ом шаге интегрирования выполняется следующая последовательность действий:

Процедура 1. Вычисляется матрица Якоби $f_n' = \partial f(y_n)/\partial y$.

Процедура 2. Формируется матрица $D_n = E + ahf'_n$, где E – единичная матрица, h – шаг интегрирования, a = 0,435866521508459 – специально вычисленный для этого метода числовой коэффициент.

Процедура 3. Вычисляются стадии $k_1,\ k_2,\ k_3$ по формулам, приведенным в [4].

Процедура 4. Вычисляется оценка ошибки на n-м шаге $\varepsilon_n(1)$.

Процедуры 5-10. Если ошибка превышает заранее заданный порог точности осуществляется дробление шага пополам и возврат к пересчету по процедуре 3. Если же ошибка с трехкратным запасом не достигает установленного порога, осуществляем удвоение шага. Для диапазона ошибки $porog/3 \le \varepsilon \le porog$ шаг остается прежним.

Процедура 11 Переход к следующему шагу интегрирования.

Указанный набор процедур был без особых сложностей воплощен в коде в среде VBA-Excel и подтвердил заявленную в [4] устойчивость. При этом следует отметить, что по быстродействию он примерно в 30 раз уступает алгоритму, предоставленному нам Е.А. Новиковым примерно 30 лет назад. Что для первых четырех шагов роли не играет.

4. Многошаговый алгоритм Адамса

Подробно метод Адамса, на первый взгляд, описан в [5]. Рекомендуемым подходом является использование сочетания «предиктор-корректор». На первой стадии используется явный метод вычисления предиктора, т.е. получение предварительного значения вектора $y^*_{n+1} \equiv y^*(t+h)$ на n+1 шаге исходя из уже вычисленных значений на предыдущих пяти шагах по формуле Адамса-Бэшфорта:

$$y^*_{n+1} = y(t) + \frac{h}{720}\left[1901f\big(y(t)\big) - 2774f(y(t-h) + 2616f(y(t-2h) - 1274f(y(t-3h) + 251f(y(t-4h)\right]$$

Следующим действием, является эффективная подстановка предиктора в качестве аргумента для вычисления скорректированного выражения y_{n+1} по формуле Адамса-Молтона:

$$y_{n+1} = y(t) + \frac{h}{720}\left[251f\big(y^*(t+h)\big) + 646f(y(t) - 264f(y(t-h) + 106f(y(t-2h) - 19f(y(t-3h)\right]$$

К нашему удивлению, описанный в [5] подход «предиктор-корректор» хотя и заработал с первого раза после составления кодов для VBA-Excel и показал высокие скоростные характеристики, но сразу же продемонстрировал склонность к неустойчивости (рис. 1). В самом учебнике на тему устойчивости метода приводится понятная по содержанию и выдающаяся по бесполезности Теорема, согласно которой «Для сходимости многошагового метода необходимо и достаточно, чтобы метод был стабилен». Как будет видно из дальнейшего именно выбор величины шага оказывает определяющее влияние не только на устойчивость алгоритма, но и на вычислительные затраты.

5. Составной алгоритм

Как следует из самой сути подхода «предиктор - корректор», перезапуск метода с более мелким шагом требует повторного накопления пяти (для выбранного порядка метода) стартовых точек, что влечет за собой некоторые дополнительные вычислительные затраты, но переводит алгоритм в устойчивый режим.

Эмпирически было подобрано пороговое значение различий между предиктором и корректором в (1-2) %. Увеличение величины порога ведет к недопустимо грубым искажениям результата интегрирования, тогда как движение в сторону уменьшения приводит к неоправданному возрастанию затрат машинного времени.

Убедившись в наличии «срыва» устойчивости при решении задачи Робертсона мы пошли по пути коррекции подхода «предиктор-корректор», а именно, в случае существенного различия между предикторным и корректорным значениями, осуществляли дробление шага пополам и возврат к значениям, полученным на предыдущем шаге (рис. 2).

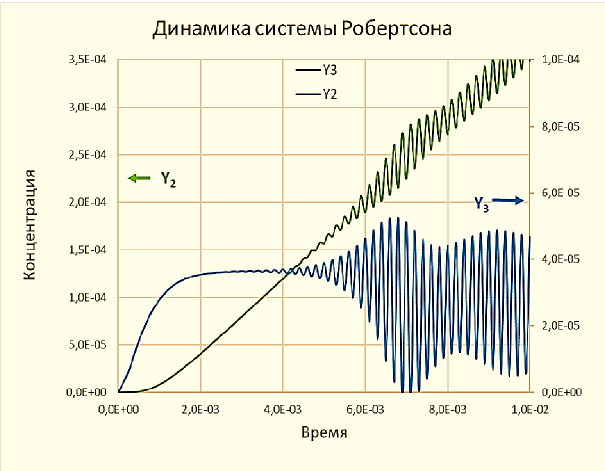

Рис.1. Метод Адамса-Бэшворта-Молтона демонстрирует неустойчивость. Шаг интегрирования 0,0001 с. Затраченное машинное время 0,65 с.

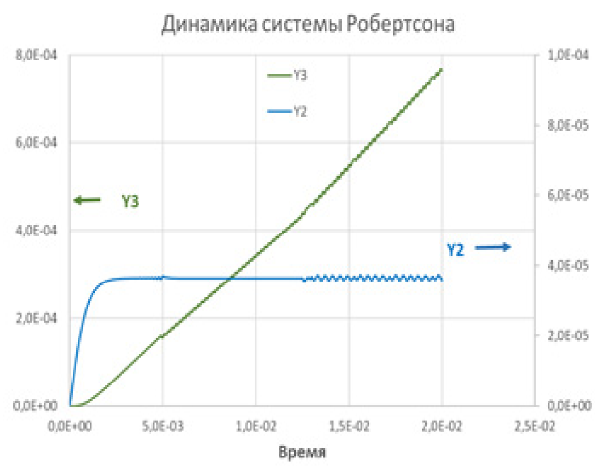

Рис.2. Тот же метод с авторегулированием шага интегрирования. Затрачено 6,92 сек. машинного времени.

6. Заключение

Проведенный сравнительный анализ различных алгоритмов решения СОДУ показал неоспоримое преимущество неявных методов над явными, а многошаговых над одношаговыми. При этом само по себе использование «устойчивых» алгоритмов не гарантирует устойчивости реализуемого метода в случае неправильного выбора шага.

Следует также отметить, что вычислительные затраты, связанные с регулированием и оптимизацией шага по ходу интегрирования могут привести к потерям времени даже более значительным, по сравнению с изначальным выбором нерегулируемого, но достаточно мелкого шага.

Список литературы

1. Хайрер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. М.: Мир, 1999, 685 с.
2. Новиков Е.А. Явные методы для жестких систем. Новосибирск: Наука СО РАН, 1997. 195 с.
3. Robertson H.H. The solution of a set of reaction rate equations // Numerical analysis: an introduction. 1966. Vol. 17, No 8. P. 182.

4. Новиков Е.А., Захаров А.А. Алгоритм переменного шага с применением метода типа Розенброка третьего порядка точности // Вестник Тюменского ГУ. Физико-математическое моделирование. Нефть, газ, энергетика. 2015. Т. 1, № 1 (1). С. 146-154.
5. Гулевич Д.Р., Залипаев В.В. Численные методы в физике и технике. СПб.: УИТМО, 2020. 211 с.