

УДК 519.8

Управление шагом в блочных диагонально-явных методах решения обыкновенных дифференциальных уравнений

О.А. Дмитриева
Донецкий национальный технический университет
dmitriv@r5.dgtu.donetsk.ua

Abstract

Dmitrieva O.A. Step Control in the Sectional Diagonal Implicit Methods of Solving Usual Differential Equations. Control algorithms of integration step for solving Cauchy problem for hard usual differential equations by diagonal implicit methods are considered. The algorithms of choosing the initial step and compressing the obtained results are offered.

Введение

Данная работа является продолжением исследований [1-5], посвященных конструированию блочных методов для решения ОДУ на параллельных компьютерах. Основная идея таких блочных методов заключается в одновременном получении приближений точного решения в равноотстоящих точках блока. Однако при численном решении жестких дифференциальных уравнений возникает необходимость в изменении шага интегрирования, поскольку малые погрешности, полученные вне пограничного слоя, позволяют существенно увеличить шаг интегрирования. При этом, с одной стороны, необходимо обеспечить длину шага, достаточно малую для достижения требуемой точности вычисляемых результатов на пограничном слое, а, с другой стороны, нужно гарантировать достаточно большую длину шага во избежание бесполезной вычислительной работы.

Известные процедуры оценки погрешности и изменения шага (правило Рунге, экстраполяция Ричардсона) имеют столь высокую трудоемкость и так замедляют время получения результата, что о практическом их применении при решении жестких задач не может быть и речи. Тогда необходимо разрабатывать алгоритмы автоматического выбора шага. Но здесь возникает новая сложность. Если для последовательного счета такие подходы, хотя и не являются широко распространенными, но все-таки иногда употребляются [6], то для параллельных реализаций, а тем более для блочных методов такие алгоритмы отсутствуют.

Схемы численного решения

Рассмотрим задачу Коши для обыкновенного дифференциального уравнения

$$y' = f(t, y(t)), \quad y(t_0) = y_0 \quad (1)$$

Один шаг численного решения задачи (1), заданной в виде

$$\frac{c}{b} A$$

или в развернутом виде

$$\begin{array}{c|cccc} c_1 & a_{11} & \cdot & \cdot & a_{1s} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ c_s & a_{s1} & \cdot & \cdot & a_{ss} \\ \hline & b_1 & \cdot & \cdot & b_s \end{array}$$

задается следующими расчетными формулами

$$y_{n+1} = y_n + h * \sum_{i=1}^s b_i k_i \quad (2)$$

$$k_i = f(t_n + c_i h, g_i)$$

$$g_i = y_n + h * \sum_{j=1}^s a_{ij} k_j, \quad i = 1, 2, \dots, s.$$

Для получения оценки погрешности численного решения будем использовать следующую формулу

$$\hat{y}_{n+1} = y_n + h * \sum_{i=1}^s \tilde{b}_i k_i \quad (3)$$

Метод является явным, если матрица $A = \{a_{ij}\}$ обращается в нуль для $j \geq i$, диагонально-явным, если матрица $A = \{a_{ij}\}$ обращается в нуль для $j > i$ или если матрица $A = \{a_{ij}\}$ диагональная, метод является неявным для всех остальных случаев. В этой работе анализ ограничится диагонально - неявными методами (1). Для таких методов s компонент блочного вектора могут быть вычислены параллельно. Следовательно, если доступны s процессоров, то время, необходимое для вычисления одного шага интегрирования, это по большей части время, необходимое для

вычисления s компонентов блока. Оптимальное число процессоров определяется как множество процессоров, для которого количество вычислений компонентов блока (последовательных) на шаг минимальное. В связи с этим нужно отметить, что часто для выполнения блочного метода необходимо менее чем s процессоров.

Диагонально-неявный метод предполагает нижнюю треугольную форму матрицы $A = \{a_{ij}\}$. Если потребовать, чтобы все диагональные элементы матрицы были равны между собой, то это позволит выполнять только одно LU – разложение на шаге интегрирования, а таблица коэффициентов этого метода в общем виде будет выглядеть как [8]

$$\begin{array}{c|cccc}
 c_1 & \gamma & & & \\
 c_2 & a_{21} & \gamma & & \\
 \cdot & \cdot & \cdot & \cdot & \\
 \cdot & \cdot & \cdot & \cdot & \\
 c_s & a_{s1} & a_{s2} & \cdot & \cdot & \gamma \\
 \hline
 & b_1 & b_2 & \cdot & \cdot & b_s
 \end{array} \quad (4)$$

Можно модифицировать приведенную схему и привести ее к виду [8]

$$\begin{array}{c|cccc}
 0 & 0 & & & \\
 c_2 & a_{21} & \gamma & & \\
 \cdot & \cdot & \cdot & \cdot & \\
 \cdot & \cdot & \cdot & \cdot & \\
 c_s & a_{s1} & a_{s2} & \cdot & \cdot & \gamma \\
 1 & b_1 & b_2 & \cdot & \cdot & b_s & \gamma \\
 \hline
 y_{n+1} & b_1 & b_2 & \cdot & \cdot & b_s & \gamma \\
 \mathcal{F}_{n+1} & \mathcal{F}_1 & \mathcal{F}_2 & \cdot & \cdot & \mathcal{F}_s & \gamma
 \end{array} \quad (5)$$

В этом случае метод будет являться $(s+1)$ – стадийным, но его вычислительные затраты совпадут с затратами $(s+1)$ – стадийного метода, так как он обладает явно вычисляемой первой стадией. К тому же у такого метода последняя строка матрицы $A = \{a_{ij}\}$ совпадает с вектором b^T , что дает значительное преимущество методу, который классифицируется как жестко точный по своему построению (FSAL – FirstSameAsLast) и используется для решения жестких и дифференциально-алгебраических систем.

В расчетную схему этого метода кроме численного приближения y_{n+1} можно внести еще и некоторое выражение \mathcal{F}_{n+1} более высокого порядка (более точное, чем y_{n+1}). Последнее может служить для управления погрешностью на каждом шаге. В частности, такая оценка удешевляет выбраковку шагов. Для получения коэффициентов схемы (5) используются следующие соотношения

$$\sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, 2, \dots, p \quad (6)$$

$$\sum_{j=1}^s a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, 2, \dots, s \quad (7)$$

$$\sum_{i=1}^s b_i c_i^{q-1} a_{ij} = \frac{b_j}{q} (1 - c_j^q), \quad j = 1, 2, \dots, s \quad (8)$$

По теореме Бутчера [6], если коэффициенты b , c , a соотношений удовлетворяют условиям (6-8) и при этом $p \leq \eta + \xi + 1$ и $p \leq 2\eta + 2$, то метод имеет порядок p . Для того, чтобы сконструировать неявную схему, удовлетворяющую соотношения (6-8) и имеющую порядок аппроксимации p , выбирают некоторую квадратурную формулу порядка p .

Неявный метод в общем случае для одного дифференциального уравнения требует решения системы из s нелинейных уравнений на каждом временном шаге. Если в качестве (1) понимается система ОДУ, то размерность системы нелинейных уравнений для каждого временного шага будет оцениваться как $n \cdot s$. Один из подходов к решению этой проблемы состоит в использовании методов с нижней треугольной матрицей $A = \{a_{ij}\}$, $i \leq j$, тогда уравнения могут быть решены за s последовательных стадий, причем на каждой стадии нужно найти решение только для системы размерности n . Эффективное решение этой системы — это одна из главных проблем при реализации неявного метода.

Вариация длины шага

В качестве процедуры управления длиной шага можно принять следующую [7]

$$h_{n+1} = \omega * h_n * \left[\frac{\text{error}_n}{\epsilon} \right]^{\frac{1}{4}}, \quad (9)$$

где ω – множитель безопасности, который обычно выбирают равным $0.8, 0.9, (0.25)^{1/(p+1)}, (0.38)^{1/(p+1)}$ чтобы на ближайших итерациях погрешность была приемлема с высокой вероятностью. Если соотношение (9) приводит к слишком быстрому изменению длины шага h , то можно выбрать новую длину шага, например следующим образом:

$$h_1(n+1) = h_n * \min(\text{factmax}, \max(\text{factmin}, \omega * \left[\frac{\text{error}_n}{\epsilon} \right]^{\frac{1}{4}} * (1/(p+1)))) \quad (10)$$

Тогда, если $\text{error} < \epsilon$, два вычисленных шага считаются принятыми, и решение продолжается, исходя из \mathcal{Y}_{n+1} или \mathcal{F}_{n+1} (в последнем случае говорят о методе «локальной экстраполяции»), причем в качестве длины нового шага берется h_{n+1} . В противном случае оба шага отбрасываются, и вычисления повторяются с новой длиной шага h_{n+1} .

Максимальный коэффициент увеличения шага $fact_{max}$ (уменьшения $fact_{min}$) обычно выбирают между 1.5 и 5, что предотвращает слишком сильное увеличение (уменьшение) длины шага. Понятно, что, если его выбрать слишком малым, это может привести и к излишнему увеличению вычислительной работы. Наконец, для шагов, выполняемых непосредственно после отбрасывания забракованных, значение $fact_{max} = 1$. В [9] предлагают, чтобы множитель безопасности ω зависел как от числа ньютоновских итераций на текущем шаге $k_{N_{new}}$, так и от максимального числа ньютоновских итераций k_{max} , например, так:

$$\omega = 0.9 * \frac{2k_{max} + 1}{2k_{max} + 2k_{N_{new}}} \quad (11)$$

Чтобы сократить количество LU-разложений матрицы для нового значения шага при условии, что матрица Якоби не вычислялась заново размер шага h_{n+1} , определенный согласно (9), не вышел из интервала

$$c_1 * h_n \leq h_{n+1} \leq c_2 * h_n, \quad (12)$$

где $c_1 = 1, c_2 = 1.2$, то оставляют значение h_n для длины следующего шага.

Предсказывающая процедура управления

Предсказание размера шага по формуле (9) имеет тот недостаток, что уменьшения размера шага более чем в ω раз невозможно без отказа от шага. При этом выполнение соотношения $h_{n+1} < \omega * h_n$ приводит к тому, что $\|\text{error}\| > 1$. Однако для жестких дифференциальных уравнений при интегрировании в пограничном слое часто требуется стремительное уменьшение размера шага. В этом случае предсказания длины шага обычно определяются как

$$h_{n+1} = \omega * h_n \left(\frac{1}{\|\text{error}_{n+1}\|} \right)^{\frac{1}{p}} \frac{h_n}{h_{n-1}} \left(\frac{\|\text{error}_n\|}{\|\text{error}_{n+1}\|} \right)^{\frac{1}{p}}, \quad (13)$$

где $\|\text{error}_{n+1}\| = C * h_n^4$,

C – константа погрешности.

Алгоритм выбора начального шага

Как правило, длина начального шага задается произвольно и считается, что плохой выбор быстро исправляется механизмом управления длиной шага. Тем не менее, если делается очень плохой выбор и это происходит слишком часто, можно потратить впустую много времени счета. Поэтому ряд авторов [6-9] предложили идеи выбора шага, опирающиеся на гипотезу об элементарных дифференциалах. Т.е., предполагая, что величина шага зависит от задаваемой пользователем точности ε , вида интегрируемой функции $f(t, y(t))$, порядка метода интегрирования p и начальных условий и границ интервала t_a, t_b . В работе предлагается к

использованию следующий алгоритм выбора начального шага

- 1) Вычисляется правая часть (1) $f(t, y(t))$, в начальной точке с координатами $t_0, y(t_0)$.
- 2) Вводится параметр

$$par = \left(\frac{1}{\max(|t_a|, |t_b|)} \right)^{p+1} + \|f\|^{p+1}$$

- 3) Первичное определение шага производится как

$$h = \left(\frac{\varepsilon}{par} \right)^{\frac{1}{p+1}}$$

Из-за того, что очень часто начальные условия находятся в особом положении, где большинство компонент $f(t, y(t))$ может оказаться нулями, к этому алгоритму добавляются еще два шага:

- 4) Выполняется один шаг методом Эйлера с длиной шага h , полученной в 3).
- 5) Шаги 1)–3) повторяются еще раз, но уже с новым начальным значением и выбирается меньшая из двух полученных длин шага.

Приведенный алгоритм хорошо «угадывает» длину начального шага или, по крайней мере, предотвращает очень плохой выбор шага. Использование этого алгоритма при решении ОДУ позволяет значительно сократить число отброшенных шагов при интегрировании.

Плотная выдача результатов интегрирования

Механизм управления длиной шага, описанный выше, выбирает точки, в которых вычисляется решение, в соответствии с требованиями к допустимой локальной погрешности. Часто, однако, требуется вычислить и выдать решение в заданных точках, которые к тому же могут быть расположены достаточно плотно. В частности, это касается конечной точки интервала интегрирования, т.к. заранее непредсказуемые длины шагов, как правило, не позволяют получить решение строго в конечной точке интервала. Одна из возможностей получения решения в таких заданных точках состоит в уменьшении длины шага всякий раз, когда это нужно. Такое многократное уменьшение длины шага от почти оптимальной величины нарушает управление длиной шага и может привести к очень сильному росту времени счета и погрешностей округления. Поэтому представляет интерес использование таких подходов, которые были бы способны выдавать значения решения во всех промежуточных точках

$$t^* = t_n + \theta * h,$$

где $0 < \theta < 1$.

Такие непрерывные методы представляют особенный интерес, если их коэффициенты a_{ij} не

зависят от θ это позволяет на каждом шаге интегрирования вычислить решение в сотнях точек без особых дополнительных затрат.

Существующие подходы, основаны на расширении методов Рунге-Кутты [7], которые подразумевают разработку специальных схем уплотнения под каждый порядок точности. Причем, поскольку построение схем основывается на переопределенных системах уравнения, коэффициенты находятся неоднозначно. К тому же может оказаться, что сформированная таким образом схема не совпадает по точности с исходным методом интегрирования. Второй подход, который используется для уплотнения выдачи результата основывается на локальной интерполяции. Но при этом необходимо дополнительно вычислять правую часть (возможно не один раз)

$$f_{n+1} = f(t_n + h, y_{n+1}),$$

поскольку без дополнительного вычисления в нашем распоряжении нет производной при новом значении y_i . Для увеличения точности используется две возможности: построение интерполяционного многочлена нужной степени (что как раз и влечет за собой дополнительные множественные вычисления правой части). Второй подход использует в качестве f_i производную при значении y_i до интерполяции, т.е. «старое» значение, что снижает точность получения результата.

В данной работе для плотной выдачи предлагаются непрерывные расширения методов [6], которые задаются следующими соотношениями:

$$\begin{aligned} b_1(\theta) &= \frac{11}{3}\theta - \frac{463}{72}\theta^2 + \frac{217}{36}\theta^3 - \frac{29}{9}\theta^4 \\ b_2(\theta) &= \frac{11}{2}\theta - \frac{385}{16}\theta^2 + \frac{661}{24}\theta^3 - 10\theta^4 \\ b_3(\theta) &= -\frac{128}{18}\theta + \frac{20125}{432}\theta^2 - \\ &\quad - \frac{8875}{216}\theta^3 + \frac{250}{27}\theta^4 \\ b_4(\theta) &= -\frac{85}{4}\theta^2 + \frac{85}{6}\theta^3 \\ b_5(\theta) &= -\frac{11}{9}\theta + \frac{557}{108}\theta^2 - \frac{359}{54}\theta^3 + \frac{80}{27}\theta^4 \end{aligned} \quad (14)$$

Эти зависимости используются также и для графического представления решения.

Сравнительная реализация решений тестовой задачи

В качестве тестовой задачи выберем жесткое уравнение

$$\begin{aligned} y' &= -50(y - \cos(t)), \\ y(0) &= 0, 0 \leq t \leq 2, \end{aligned} \quad (15)$$

точное решение которого известно и равно $y(t) = (10e^{-50t} - 50 +$

$$50e^{150t} \cos(t) + e^{150t} \sin(t))/2501]$$

Результаты численного решения этого уравнения явными методами приведены в [6] и характеризуются недопустимыми погрешностями.

Используем для решения диагонально-неявный метод, полученный введением к качеству интерполяционного многочлена полинома Гаусса, обеспечивающего требуемый порядок p и соответствующее порядку минимально возможное число стадий s . Для вывода расчетной схемы зададимся значением порядка $p=4$ и $s=5$. Вид расчетной схемы приведен в таблице 1, в которую ввели еще один дополнительный вектор f_{n+1} , который позволит оценивать погрешность, достигаемую на шаге интегрирования, и управлять ею. Вектор err введем для визуализации полученной погрешности.

Таблица 1

Расчетная схема диагонально-неявного метода с $p=4, s=5$

1/4	1/4				
3/4	1/2	1/4			
11/20	17/50	-1/25	1/4		
1/2	371/1360	-137/2720	15/544	1/4	
1	25/24	-49/48	125/16	-85/12	1/4
y_{n+1}	25/24	-49/48	125/16	-85/12	1/4
f_{n+1}	59/48	-17/96	225/32	-85/12	0
err	-3/16	-27/32	25/32	0	1/4

Результаты реализации уравнения (15) с помощью неявного метода при различных задаваемых предельных погрешностях.

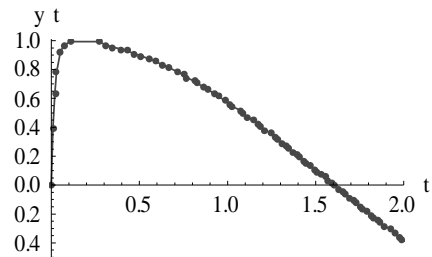
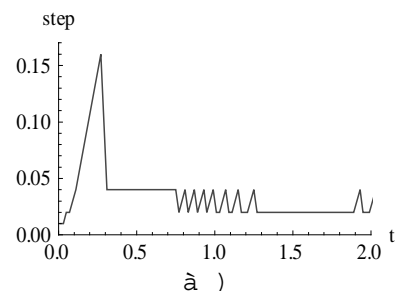


Рисунок 1 – Точное (сплошная линия) и численное решение уравнения (15)



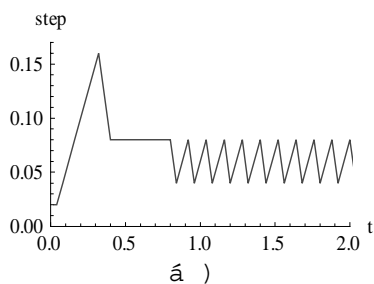


Рисунок 2 – Вариация длины шага

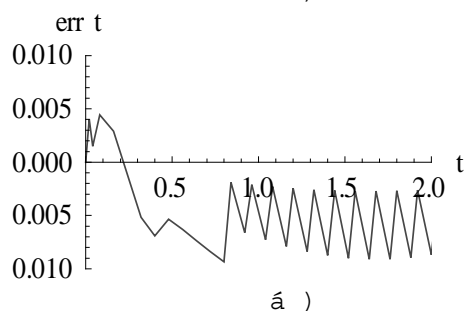
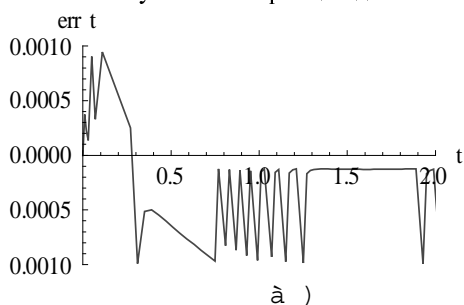


Рисунок 3 – Величины погрешности решения

Заключение

Предложены диагонально-явные методы решения задачи Коши для жестких обыкновенных дифференциальных уравнений и их систем с вариацией шага. В процессе решения осуществляется автоматическая настройка размерности шага интегрирования, обеспечивающая заданную точность.

Получены сравнительные временные характеристики решения тестовой задачи и оценки максимальной погрешности результатов, достигнутых с помощью диагонально-явного метода.

Показано, что эффективность решения задачи Коши для обыкновенных дифференциальных уравнений зависит не только от используемого численного метода и его коэффициентов, но и от выбора начального значения и процедуры управления величиной шага.

Литература

1. Фельдман Л.П., Дмитриева О.А. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений.// Математическое моделирование, том 13, № 7, 2001. – С.66-72.
2. Дмитриева О.А. Об особенностях моделирования линейных динамических систем в многопроцессорных средах// Электронное моделирование, № 2, 2007. С. 63-72.
3. Дмитриева О.А. Анализ параллельных алгоритмов численного решения систем обыкновенных дифференциальных уравнений методами Адамса-Башфорта и Адамса-Моултона.// Математическое моделирование 2000, т. 12, № 5, С. 81-86.
4. Дмитриева О. А. Параллельные алгоритмы обработки систем линейных уравнений/ Материалы XV международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС, 2007) 25-31 мая 2007 г., Алушта – М.: Вузовская книга, 2007. С. 213-215
5. Дмитриева О.А. Устойчивость численного моделирования динамических систем// Материалы VII Международной конференции по неравновесным процессам в соплах и струях. (NPNJ-2008), 25 – 31 мая 2008 г., Алушта – М.: Вузовская книга, 2008. С. 310-313.
6. Хайрер Э., Ваннер. Г. Решение обыкновенных дифференциальных уравнений. Жесткие задачи. - М.: Мир, 1999.- 685с.
7. Хайрер Э., Нерсетт С., Ваннер. Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. - М.: Мир, 1990.- 512с.
8. Скворцов Л.М. Экономичная схема реализации неявных методов Рунге-Кутты.// Журнал вычислительной математики и математической физики, 2008, т.48, № 11, С. 2008-2018.

Поступила в редакцию 15.03.2010