УДК 519.622

ЯВНЫЕ МНОГОШАГОВЫЕ МЕТОДЫ ЧИСЛЕННОГО РЕШЕНИЯ ЖЕСТКИХ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Л. М. Скворцов[1]

Для решения жестких задач предлагается использовать явные многошаговые методы двух типов: адаптивные (нелинейные) методы и стабилизированные методы, имеющие расширенные области устойчивости. Рассмотрены методы 1-го и 2-го порядков с автоматическим выбором шага интегрирования. На тестовых задачах демонстрируется эффективность предложенных методов.

Ключевые слова: обыкновенные дифференциальные уравнения, жесткая задача Коши, явные многошаговые методы.

1. Введение. В настоящее время общепринятым является мнение, что для решения жестких систем обыкновенных дифференциальных уравнений (ОДУ) следует использовать неявные методы, которые позволяют интегрировать такие системы с относительно малыми (по сравнению с явными методами) вычислительными затратами. По этой причине в пакеты прикладных программ включают неявные методы, специально предназначенные для решения жестких задач. Например, в системе математических вычислений MATLAB из семи решателей ОДУ с переменным шагом четыре являются неявными.

В то же время неявные методы не свободны от недостатков, к которым относятся, прежде всего, сложность реализации и необходимость вычисления матрицы Якоби. В тех случаях, когда правая часть содержит разрывы или логические условия, вычисление матрицы Якоби может представлять собой сложную и далеко не тривиальную задачу. Отметим также, что при решении некоторых жестких нелинейных задач применение неявных методов дает неудовлетворительные результаты: они требуют больших вычислительных затрат, обеспечивая при этом весьма низкую точность. Примеры таких задач приведены в [1–3].

Недостатки неявных методов побуждают исследователей разрабатывать специальные явные методы для жестких задач, свободные от этих недостатков. Можно выделить два типа таких методов: нелинейные методы и методы с расширенными областями устойчивости. В статье приводятся детальные описания простых в реализации методов этих двух типов. Вычислительные эксперименты показали, что эти методы могут быть альтернативой неявным методам при решении жестких задач с умеренной точностью.

В [1, 2, 4] были рассмотрены явные нелинейные одношаговые методы, вычислительные затраты которых при решении жестких систем ОДУ значительно меньше, чем при использовании обычных явных методов. В этих методах осуществляется самонастройка параметров расчетной схемы на решаемую задачу с использованием полученных по результатам предварительных стадий покомпонентных оценок собственных значений матрицы Якоби. Поэтому методы такого типа были названы адаптивными.

Опыт применения одношаговых адаптивных методов показал, что они могут быть эффективны при решении умеренно жестких задач с невысокой точностью. Построение одношаговых методов, эффективных при решении жестких задач с высокой точностью, проблематично из-за их низкого стадийного порядка [5, 6], неустойчивости внутренних стадий, а также неустойчивости вложенной формулы, применяемой для оценивания погрешности. Адаптивные методы повышенной точности были построены на основе предложенных в [1, 2] многошаговых формул. Наиболее эффективным оказался явный многошаговый метод переменного порядка [3], который показывает превосходные результаты в широком диапазоне требований к точности. Результаты сравнения этого метода с неявными методами приведены в [3, 7, 8]. Он довольно сложен, поэтому в [3] приведены основные расчетные формулы, но отсутствует детальное описание. Многошаговые методы низких порядков значительно проще, а при умеренных требованиях к

[1] Московский государственный технический университет им. Н.Э. Баумана (МГТУ), факультет специального машиностроения, 2-я Бауманская ул., д. 5, 105005, Москва; зав. лабораторией, e-mail: lm_skvo@rambler.ru

точности они не уступают методу переменного порядка. Мы рассмотрим наиболее простые методы 1-го и 2-го порядков. Отметим, что здесь, как и в [4], указывается реальный порядок метода при решении жестких задач, но для нежестких задач порядок может быть выше. Это имеет принципиальное значение, поскольку многие из предлагавшихся ранее нелинейных методов реально обеспечивают только нулевой порядок для жестких задач. Это показано экспериментально в [1, 2].

Другой тип методов, который будет здесь рассмотрен — стабилизированные методы, имеющие расширенные вдоль вещественной оси области устойчивости. Явные одношаговые методы такого типа успешно применяются для решения жестких задач с распределенным вещественным спектром матрицы Якоби [9–13]. Многочлены устойчивости таких методов построены на основе условия чебышевского альтернанса, поэтому их иногда называют методами Рунге–Кутта–Чебышева. Расширение области устойчивости осуществляется за счет выполнения на каждом шаге интегрирования большого числа внутренних стадий.

В статье предлагается альтернативный подход, основанный на использовании многошаговых схем с двумя стадиями. Длина области устойчивости таких методов вдоль вещественной оси может быть сколь угодно большой, но скорость увеличения этой длины от шага к шагу ограничена. Выбор величины шага осуществляется на основе оценки ошибки и оценки наибольшего по модулю собственного значения матрицы Якоби. В программных реализациях одношаговых методов чебышевского типа (DUMKA, RKC, ROCK) [9–13] требуется дополнительная процедура оценивания наибольшего собственного значения. В предложенных методах (двухшаговом 1-го порядка и трехшаговом 2-го порядка) необходимость в такой процедуре отпадает, поскольку она уже встроена в расчетную схему, не требуя при этом дополнительных вычислений правой части или якобиана.

Как и в [4], мы приводим все расчетные формулы, необходимые для программной реализации рассмотренных методов с автоматическим выбором шага.

2. Адаптивный метод 1-го порядка. Будем решать задачу Коши для системы ОДУ

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

где t — независимая переменная и y — n-мерный вектор. Пусть в процессе интегрирования системы (1) получено численное решение $y_1, \ldots, y_m$ в точках $t_1, \ldots, t_m$, и нам нужно найти решение y_{m+1} в следующей точке $t_{m+1} = t_m + h_m$.

При построении многошаговых адаптивных методов используется та же идея, что и при построении одношаговых методов [1, 2, 4]. Формула шага интегрирования задается так, что при решении модельного уравнения Далквиста $y' = \lambda y$, $y(t_0) = y_0$, численное решение получаем в виде $y_{m+1} = Q(h_m\lambda)y_m$. Функция $Q(z)$ определяется так, чтобы обеспечить правильный характер (затухание или расхождение) численного решения при больших по модулю значениях z и необходимую точность при малых z. Исходя из этого, для методов 1-го и 2-го порядков зададим

$$Q(z) = \begin{cases} 1 + z + \dfrac{z^2}{2} + \dfrac{z^3}{6}, & |z| \leqslant 1.6, \\ 0, & z < -1.6, \\ 1 + 2.23z, & z > 1.6. \end{cases} \tag{2}$$

Константы в формуле (2) выбраны из условия непрерывности функции $Q(z)$. При решении системы (1) формула шага интегрирования реализуется покомпонентно, с использованием покомпонентных оценок наибольшего по модулю собственного значения.

Предложенные в [1–3] многошаговые адаптивные методы являются трехстадийными, а их порядок при решении жестких задач равен числу используемых шагов (это подтверждено экспериментально в [2]). Таким образом, метод 1-го порядка является одношаговым, но для оценивания ошибки будем использовать двухшаговую формулу. Стадии этого метода выполняются по формулам

$$\begin{aligned} f_m &= f(t_m, y_m), \quad & u_1 &= y_m + h_m f_m, & g_1 &= f(t_{m+1}, u_1), \\ & & u_2 &= u_1 + h_m\alpha(g_1 - f_m), & g_2 &= f(t_{m+1}, u_2), \end{aligned} \tag{3}$$

где $0 < \alpha \leqslant 0.5$. Далее вычисляются покомпонентные оценки наибольшего по модулю собственного значения матрицы hA, где A — матрица Якоби. Применяя степенной метод согласно [4], получаем оценку по i-й компоненте в виде

$$z_1^{(i)} = \frac{b}{a}, \quad a = \alpha\big(g_1^{(i)} - f_m^{(i)}\big), \quad b = g_2^{(i)} - g_1^{(i)}. \tag{4}$$

На основе полученных оценок вычисляются настраиваемые параметры

$$c_1^{(i)} = \frac{Q\left(z_1^{(i)}\right) - 1}{z_1^{(i)}}, \quad c_2^{(i)} = \frac{c_1^{(i)} - 1}{z_1^{(i)}}, \tag{5}$$

которые используются в заключительной формуле шага интегрирования

$$y_{m+1}^{(i)} = u_1^{(i)} + h_m c_2^{(i)} \left(g_1^{(i)} - f_m^{(i)}\right), \quad i = 1, \ldots, n. \tag{6}$$

Вычисления по формулам (2), (4), (5) следует организовать так, чтобы исключить переполнение или деление на 0. Для этого, в зависимости от соотношения a и b в (4), следует вычислять оценку собственного значения либо величину, обратную этой оценке. Алгоритм вычисления параметров $c_1^{(i)}$ и $c_2^{(i)}$ по значениям a и b удобно записать в виде следующего фрагмента программы на языке Паскаль:

```
if abs(b) <= 1.6*abs(a) then
begin
    if a<>0 then b:=b/a;
    c2:=1/2+b/6; c1:=1+b*c2;
end else
begin
    a:=a/b;
    if a<0 then c1:=-a else c1:=2.23;
    c2:=a*(c1-1);
end;
```

Этот алгоритм включен в цикл по i от 1 до n, внутри которого вычисляются компоненты вектора y_{m+1}, а также оценка ошибки err на шаге интегрирования.

Оценивание ошибки выполняется по двухшаговой формуле

$$\delta y^{(i)} = \left(1 - c_1^{(i)}\right)\left(u_1^{(i)} - y_m^{(i)} - w\left(y_m^{(i)} - y_{m-1}^{(i)}\right)\right) + h_m c_2^{(i)}\left(g_1^{(i)} - f_m^{(i)} - w\left(f_m^{(i)} - f_{m-1}^{(i)}\right)\right), \quad w = \frac{h_m}{h_{m-1}}, \tag{7}$$

$$\mathrm{err} = \max_i \frac{\left|\delta y^{(i)}\right|}{\mathrm{Atol} + \mathrm{Rtol}\,\max\left(\left|y_m^{(i)}\right|, \left|y_{m+1}^{(i)}\right|\right)}, \tag{8}$$

где Atol — допустимая абсолютная ошибка и Rtol — допустимая относительная ошибка. На первом шаге принимаем $y_{-1} = y_0$, $f_{-1} = f_0$. Нормированная оценка ошибки err используется для определения величины следующего шага по формуле

$$h_{m+1} = w_{\mathrm{new}} h_m, \quad w_{\mathrm{new}} = 0.7\,\mathrm{err}^{-\gamma}, \quad \gamma = \frac{1}{3}. \tag{9}$$

Если $\mathrm{err} > 1$, то шаг считается неудачным и отбрасывается, после чего производится перерасчет с уменьшенным размером шага. Вычисленное согласно (9) значение w_{new} рекомендуется ограничить снизу и сверху (например, значениями $w_{\min} = 0.25$, $w_{\max} = 4$).

Остановимся теперь на выборе параметра α в (3). Значение этого параметра практически не влияет на эффективность решения линейных либо умеренно жестких задач. Однако положение существенно изменяется при интегрировании жестких нелинейных систем. Стадийное значение u_2 характеризуется функцией устойчивости $R_2(z) = 1 + z + \alpha z^2$. При большом значении α возможно значительное отклонение u_2 от траектории решения, что может привести к недостоверности оценок собственных значений. Как следствие, уменьшается шаг интегрирования и увеличиваются вычислительные затраты. Чтобы этого не происходило, следует задавать малое значение α. Для большинства жестких нелинейных задач диапазон оптимальных значений α достаточно широк (например, можно задать $\alpha = 10^{-3}$ либо $\alpha = 10^{-6}$; в обоих случаях точность и вычислительные затраты будут примерно одинаковы). Однако для некоторых задач (в числе которых рассмотренная ниже задача ROBER) существенное преимущество дает более точная настройка, полученная из условия малого значения $R_2(z)$ для наиболее жесткой компоненты. Такая настройка задается формулой

$$\alpha = \min\left\{\alpha_{\max}, \min_i\left\{\left|w z_1^{(i)}\right|^{-1}\right\}\right\}, \tag{10}$$

где $\alpha_{\max} = 0.5$ и $z_1^{(i)}$ — оценки (4), полученные на предыдущем шаге интегрирования.

Формулы (3)–(6) практически совпадают с формулами, приведенными в [4] для одношагового метода 1-го порядка. Отличие состоит в том, что оценивание ошибки осуществляется по двухшаговой формуле (7), а полученная оценка стабилизирована в тех же точках жесткого спектра, что и шаг интегрирования (6). Это позволяет избавиться от колебаний величины шага, вследствие чего вычислительные затраты уменьшаются в несколько раз по сравнению с методом, приведенным в [4]. Таким образом, использование многошаговой формулы оценивания ошибки позволило заметно повысить эффективность одношагового метода. Еще более эффективным может быть метод, в котором формула для шага интегрирования также является многошаговой.

3. Адаптивный метод 2-го порядка. Стадии двухшагового метода 2-го порядка выполняются по формулам:

$$f_m = f(t_m, y_m), \quad w = \frac{h_m}{h_{m-1}}, \quad u_1 = y_m + h_m f_m + \frac{h_m}{2} w(f_m - f_{m-1}), \quad g_1 = f(t_{m+1}, u_1),$$
$$u_2 = u_1 + h_m \alpha \nabla^2 f, \quad g_2 = f(t_{m+1}, u_2).$$

В этих и последующих формулах используются разности назад 2-го порядка, которые обозначим через $\nabla^2 y = (u_1 - y_m) - w(y_m - y_{m-1})$ и $\nabla^2 f = (g_1 - f_m) - w(f_m - f_{m-1})$. На первом шаге принимаем $y_{-1} = y_0$, $f_{-1} = f_0$. Как и в методе 1-го порядка, параметр α следует задавать достаточно малым (например, $\alpha = 10^{-3}$) либо вычислять согласно (10).

Покомпонентные оценки наибольшего по модулю собственного значения получаем в виде $z_1^{(i)} = \dfrac{b}{a}$, $a = \alpha \nabla^2 f^{(i)}$, $b = g_2^{(i)} - g_1^{(i)}$. Эти оценки используются для расчета настраиваемых параметров согласно формулам (5) и $c_3^{(i)} = \dfrac{c_2^{(i)} - 0.5}{z_1^{(i)}}$. Алгоритм вычисления этих параметров по значениям a и b представим в виде

```
if abs(b) <= 1.6*abs(a) then
begin
    if a<>0 then b:=b/a;
    c3:=1/6; c2:=0.5+b*c3; c1:=1+b*c2;
end else
begin
    a:=a/b;
    if a<0 then c1:=-a else c1:=2.23;
    c2:=a*(c1-1); c3:=a*(c2-0.5);
end;
```

Шаг интегрирования выполняется по формуле

$$\delta y^{(i)} = \frac{1 - c_1^{(i)} + w\left(1 - 2c_2^{(i)}\right)}{1 + w} \nabla^2 y^{(i)} + h_m \frac{c_2^{(i)} + 2w c_3^{(i)}}{1 + w} \nabla^2 f^{(i)},$$
$$y_{m+1}^{(i)} = y_m + h_m c_1^{(i)} f_m^{(i)} + w\left(1 - c_1^{(i)}\right)\left(y_m^{(i)} - y_{m-1}^{(i)}\right) + h_m w c_2^{(i)}\left(f_m^{(i)} - f_{m-1}^{(i)}\right) + \delta y^{(i)}, \quad i = 1, \ldots, n,$$

а значения $\delta y^{(i)}$ используются для вычисления нормированной оценки ошибки и размера следующего шага согласно (8) и (9).

4. Стабилизированный метод 1-го порядка. Рассмотрим теперь многошаговые методы, области устойчивости которых расширены вдоль вещественной оси, что позволяет эффективно решать многие жесткие задачи. Двухшаговый метод такого типа запишется в виде

$$\begin{aligned} &\widehat{y}_{m+1} = y_m + h_m f_m, \quad \widehat{f}_{m+1} = f\left(t_{m+1}, \widehat{y}_{m+1}\right), \\ &y_{m+1} = y_m + b_0(y_m - y_{m-1}) + h_m\Big(b_1 f_m + b_2\left(\widehat{f}_{m+1} - f_m\right)\Big), \quad f_{m+1} = f(t_{m+1}, y_{m+1}), \end{aligned} \tag{11}$$

где из условия 1-го порядка имеем $b_1 = 1 - \dfrac{b_0}{w}$, $w = \dfrac{h_m}{h_{m-1}}$. Исследуем устойчивость этого метода. Применяя его для решения уравнения $y' = \lambda y$, получаем разностную схему

$$y_{m+1} = R(z) y_m - b_0 y_{m-1}, \quad R(z) = 1 + b_0 + b_1 z + b_2 z^2, \quad z = h_m \lambda, \tag{12}$$

характеристическое уравнение которой имеет вид

$$\mu^2 - R(z)\mu + b_0 = 0. \tag{13}$$

Согласно определению V.1.1 из [12], точка z принадлежит области устойчивости многошагового метода, если все корни $\mu_i(z)$ характеристического уравнения удовлетворяют неравенству $\left|\mu_i(z)\right| \leqslant 1$ и при этом среди корней нет кратных, равных по модулю 1. Строго говоря, использование уравнения (13) для исследования устойчивости допустимо, если его коэффициенты $R(z)$ и b_0 постоянны. Однако можно воспользоваться методом замороженных коэффициентов, предположив, что эти коэффициенты изменяются достаточно медленно. Такое предположение является обычным при исследовании устойчивости многошаговых методов, коэффициенты которых изменяются при изменении величины шага. В нашем случае коэффициенты характеристического полинома не изменяются при постоянном шаге, а при небольших изменениях величины шага изменяются незначительно. Устойчивость предложенной схемы подтверждается также многочисленными экспериментами.

Устойчивость разностной схемы (12) определяется двумя корнями $\mu_{1,2}(z) = \dfrac{1}{2}\left(R(z) \pm \sqrt{R^2(z) - 4b_0}\right)$ уравнения (13). Область устойчивости задается неравенствами

$$\left|\mu_1(z)\right| \leqslant 1, \quad \left|\mu_2(z)\right| \leqslant 1, \tag{14}$$

при этом оба корня не должны одновременно равняться 1 или -1. Воспользовавшись алгебраическим критерием устойчивости дискретных систем [14], запишем условие (14) в более удобном виде:

$$S_1(z) = 1 - R(z) + b_0 \geqslant 0, \quad S_2(z) = 1 - b_0 \geqslant 0, \quad S_3(z) = 1 + R(z) + b_0 \geqslant 0. \tag{15}$$

Подставив в (15) значение $R(z)$, получим $S_1 = -\left(1 - \dfrac{b_0}{w}\right)z - b_2 z^2$, $S_3 = 2(1 + b_0) + \left(1 - \dfrac{b_0}{w}\right)z + b_2 z^2$.

Пусть $[-l_m, 0]$ — интервал устойчивости на очередном шаге интегрирования, а l_m — его длина. Чтобы было $l_m \geqslant 2$, потребуем выполнения неравенства

$$0 \leqslant b_0 < \min\{w,\, 1\}. \tag{16}$$

Приняв $S_1(-l_m) = 0$, получим длину области устойчивости

$$l_m = \frac{w - b_0}{w b_2}. \tag{17}$$

Чтобы обеспечить положительность $S_3(z)$ на интервале $[-l_m, 0]$, примем

$$\min S_3(z) = S_3\left(-\frac{l_m}{2}\right) = \frac{8w^2(1 + b_0)b_2 - (w - b_0)^2}{4w^2 b_2} = \varepsilon > 0. \tag{18}$$

Удобно задать $\varepsilon = \dfrac{3}{2}(1 - b_0)$, тогда из (17) и (18) получим параметры метода в виде

$$b_0 = w\,\frac{l_m - 2}{l_m + 14w}, \quad b_1 = 1 - \frac{b_0}{w}, \quad b_2 = \frac{b_1}{l_m}; \tag{19}$$

при $l_m = 2$ имеем одношаговый метод второго порядка $\widehat{y}_{m+1} = y_m + h_m f_m$, $y_{m+1} = y_m + \dfrac{h_m}{2}\left(f_m + \widehat{f}_{m+1}\right)$, который используется как стартовый на первом шаге интегрирования. Из (16) и (19) следует, что при $w \leqslant 1$ длина интервала устойчивости может быть сколь угодно большой.

Оценим, насколько быстро может возрастать величина шага без потери устойчивости. При $w > 1$ из (16) и (19) получим

$$w\,\frac{l_m - 2}{l_m + 14w} < 1. \tag{20}$$

Пусть предыдущий шаг был максимального (исходя из условия устойчивости) размера. Тогда величина шага должна расти не быстрее, чем длина интервала устойчивости, что выражается неравенством

$$w \leqslant \frac{l_m}{l_{m-1}}. \tag{21}$$

Максимальная величина очередного шага обеспечивается при $w = \dfrac{l_m}{l_{m-1}}$. Подставив это значение в (20), получим ограничение на увеличение длины интервала устойчивости на одном шаге в виде $l_m < l_{m-1} + 16$, а из (21) получим неравенство, ограничивающее рост величины шага: $w < \dfrac{l_{m-1} + 16}{l_{m-1}}$.

Опишем алгоритм шага интегрирования. На основе оценки ошибки предыдущего шага $\delta y_n = y_n - \widehat{y}_n$ определяем по формуле (8) нормированную оценку ошибки err. Вычисляем

$$\widehat{z}_{m-1} = h_{m-1}\widehat{\lambda}_m, \quad w = \min\left\{0.5\,\mathrm{err}^{-0.5}, \frac{\bigl|\widehat{z}_{m-1}\bigr| + \Delta l}{\bigl|\widehat{z}_{m-1}\bigr|}\right\}, \quad h_m = wh_{m-1}, \quad l_m = \max\left\{2,\, w\bigl|\widehat{z}_{m-1}\bigr|\right\}, \tag{22}$$

где $\widehat{z}_{m-1}$ — текущая оценка наибольшего по модулю отрицательного собственного значения матрицы Якоби, а Δl — максимальное приращение длины интервала устойчивости на одном шаге ($\Delta l < 16$). Определяем параметры метода по формулам (19) и выполняем шаг интегрирования согласно (11).

Как и в адаптивных методах, оценивание границы жесткого спектра осуществляется на основе степенного метода, при этом используется приближенное соотношение $\delta f_m \approx J_m \delta y_m$, $\delta y_m = y_m - \widehat{y}_m$, $\delta f_m = f_m - \widehat{f}_m$, где J_m — матрица Якоби, вычисленная при $t = t_m$, $y = y_m$. Для получения оценки величины $\widehat{\lambda}_m$ сначала вычисляются сглаженные по независимой переменной покомпонентные оценки, среди которых затем выбирается минимальная. Сглаживание производится экспоненциально взвешенным методом наименьших квадратов. Рекуррентные расчетные формулы для получения оценки по i-й компоненте запишутся в виде

$$\widehat{\lambda}_m^{(i)} = \widehat{\lambda}_{m-1}^{(i)} + \frac{\delta y_m^{(i)}}{d_m^{(i)}}\left(\delta f_m^{(i)} - \widehat{\lambda}_{m-1}^{(i)}\delta y_m^{(i)}\right), \quad d_m^{(i)} = \gamma d_{m-1}^{(i)} + \left(\delta y_m^{(i)}\right)^2, \quad \widehat{\lambda}_0^{(i)} = 0, \quad d_0^{(i)} = 0, \quad 0 < \gamma \leqslant 1. \tag{23}$$

Значение γ следует выбирать как компромисс между точностью (чем ближе γ к 1, тем точнее оценка) и скоростью оценивания (чем меньше γ, тем быстрее может изменяться оценка при изменении собственного значения). Окончательную оценку получаем в виде

$$\widehat{\lambda}_m = k \min_i \widehat{\lambda}_m^{(i)}, \tag{24}$$

где значение k следует задавать чуть больше 1.

По результатам решения ряда жестких задач были выбраны следующие значения констант, входящих в формулы (22)–(24): $\Delta l = 8$, $\gamma = 0.9$, $k = 1.1$.

5. Стабилизированный метод 2-го порядка. Трехшаговый метод 2-го порядка зададим в виде

$$\begin{aligned}
&\widehat{y}_{m+1}^{(i)} = y_m + h_m f_m, \quad \widehat{f}_{m+1} = f\bigl(t_{m+1}, \widehat{y}_{m+1}\bigr),\\
&y_{m+1} = y_m + b_0(y_m - y_{m-1}) + c_0\bigl(y_m - (1+w_2)y_{m-1} + w_2 y_{m-2}\bigr) +\\
&\qquad + h_m\Bigl[b_1 f_m + b_2\bigl(\widehat{f}_{m+1} - f_m\bigr) + c_1 f_{m-1} + c_2 w_1\bigl(\widehat{f}_m - f_{m-1}\bigr)\Bigr], \quad f_{m+1} = f(t_{m+1}, y_{m+1}),
\end{aligned} \tag{25}$$

где $w_1 = \dfrac{h_m}{h_{m-1}}$ и $w_2 = \dfrac{h_{m-1}}{h_{m-2}}$. Для обеспечения 2-го порядка должны выполняться условия

$$\frac{1}{w_1}b_0 + b_1 + c_1 = 1, \quad \frac{1+w_2}{2w_1^2 w_2}c_0 - \frac{1}{2w_1^2}b_0 - \frac{1}{w_1}c_1 + b_2 + c_2 = \frac{1}{2}.$$

Кроме того, должны выполняться условия, обеспечивающие заданную длину интервала устойчивости. Как и для метода 1-го порядка, они получаются на основе алгебраического критерия устойчивости дискретных систем. Вывод формул для расчета коэффициентов достаточно сложен, поэтому приведем только окончательные формулы.

Опишем алгоритм одного шага интегрирования. Оценка ошибки предыдущего шага определяется формулой $\delta y_m = y_m - \widehat{y}_m$. Определим по формуле (8) нормированную оценку ошибки err. Вычисляем $\widehat{z}_{m-1} = h_{m-1}\widehat{\lambda}_m$, $w_1 = \min\left\{\dfrac{1}{2}\ \mathrm{err}^{-0.5}, \dfrac{\bigl|\widehat{z}_{m-1}\bigr| + 2}{\bigl|\widehat{z}_{m-1}\bigr|}\right\}$, $h_m = w_1 h_{m-1}$, $l_m = \max\left\{2,\, w_1\bigl|\widehat{z}_{m-1}\bigr|\right\}$, где $\widehat{\lambda}_m$ — оценка наибольшего по модулю отрицательного собственного значения матрицы Якоби, полученная согласно формулам (23) и (24), в которых рекомендуется задать $\gamma = 0.9$ и $k = 1.2$. Определяем параметры

метода по формулам

$$K_1 = \frac{8}{7}\,\frac{14l_m - 27}{l_m - 1}\,, \quad K_2 = \frac{4}{3}\,\frac{12l_m - 23}{l_m - 1}\,,$$

$$c_0 = w_1 w_2\,\frac{K_1 l_m(1+w_1)(K_2 l_m - 8K_2 + 8) + 32w_1(K_1 - 1)(3K_2 - 4)}{K_1 l_m(1+w_2)\big(K_2 l_m + 8w_1 w_2(K_2 - 1)\big) + 32w_1^2 w_2^2(K_1 - 1)(3K_2 - 4)}\,,$$

$$b_0 = w_1 - 16w_1(1 - w_2 c_0)\,\frac{K_1 - 1}{K_1 l_m}\,,$$

$$c_1 = \frac{1}{2}\left(\frac{1+w_2}{w_1 w_2}\,c_0 - \frac{l_m + 2w_1}{w_1 l_m}\,b_0 - w_1\,\frac{l_m - 2}{l_m}\right), \quad b_1 = 1 - \frac{b_0}{w_1} - c_1, \quad b_2 = \frac{b_1}{l_m}\,, \quad c_2 = \frac{c_1}{l_m}$$

и выполняем шаг интегрирования согласно (25). На первых двух шагах принимаем $b_0 = c_0 = c_1 = c_2 = 0$, $b_1 = 1$, $b_2 = 0.5$, т.е. используем одношаговый метод.

6. Результаты решения тестовых задач. Рассмотренные здесь методы были реализованы в виде компьютерных программ, задаваемыми параметрами которых являются допустимая относительная ошибка Rtol, допустимая абсолютная ошибка Atol и величина начального шага h_0. В качестве показателя точности при тестировании методов использовалось значение $\mathrm{scd} = -\lg\left(\max\limits_i \left\{\dfrac{|y_i - \widetilde{y}_i|}{|y_i|}\right\}\right)$, где y_i — точное, а $\widetilde{y}_i$ — численное решение по i-й компоненте в конечной точке интервала интегрирования. Вычислительные затраты оценивались количеством вычислений правой части Nf. Обозначим через AM1 и AM2 адаптивные методы 1-го и 2-го порядков, а через SEM1 и SEM2 стабилизированные методы 1-го и 2-го порядков (SEM — Stabilized Explicit Multistep method).

Для сравнения приводим результаты двух неявных методов 2-го порядка. Первый из них — диагонально неявный метод Рунге–Кутта (обозначим его DIRK2), реализованный в программном комплексе "МВТУ" [8]. Второй — метод Розенброка, реализованный в системе MATLAB под именем ode23s. В приводимых для этих методов значениях Nf учитываются также и вычисления правой части при расчете якобиана, выполняемом посредством численного дифференцирования.

Таблица 1

Результаты решения задачи VDPOL

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	1.32	1 015	1.81	2 753	2.34	7 262	3.53	50 169
AM2	2.42	1 030	3.82	2 822	4.99	7 478	6.69	52 085
SEM1	0.95	9 010	1.40	18 661	1.99	44 969	3.25	302 075
SEM2	3.31	**74 198**	2.66	24 813	2.89	39 680	3.82	236 485
DIRK2	1.99	1 104	2.39	2 190	3.14	4 097	4.64	19 068
ode23s	1.95	1 116	2.58	2 492	3.26	6 584	4.59	44 946

Для тестирования были взяты шесть задач из [12]. Описания пяти из них (VDPOL, OREGO, HIRES, CUSP и BRUSS) даны в [4], где приведены также результаты их решения явными одношаговыми методами (адаптивными, методом Мерсона) и адаптивным многошаговым методом переменного порядка. Значения h_0 и Atol задаем такими же, как в [4]. Результаты решения этих задач приведены в табл. 1–6. Задача BRUSS решалась при двух значениях N (число уравнений $n = 2N$): $N = 100$ (табл. 5) и $N = 500$ (табл. 6). При $N = 500$ задача становится очень жесткой (наибольшее по модулю собственное значение примерно равно $-20\,000$), и для ее решения явным методом Эйлера необходимо около 100 000 вычислений правой части.

Шестая задача — ROBER (модель химической реакции):

$$y_1' = -0.04y_1 + 10^4 y_2 y_3, \quad y_2' = 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, \quad y_3' = 3 \times 10^7 y_2^2,$$
$$y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad 0 \leqslant t \leqslant 10^{11}.$$

На таком интервале эту задачу невозможно решить обычными явными методами, поскольку одно из собственных значений якобиана равно -10^4, а число вычислений правой части для устойчивого решения должно быть величиной порядка 10^{15}. Среди явных методов с этой задачей успешно справились только

адаптивные методы. Результаты при $h_0 = 10^{-6}$ и $\mathrm{Atol} = 10^{-12}\,\mathrm{Rtol}$ приведены в табл. 7. В этой же таблице приведены результаты адаптивного метода переменного порядка, который обозначен как A4 (в программном комплексе "МВТУ" [8] он назван "Адаптивный 4").

Рассмотренные здесь методы относятся к методам невысокой точности, поэтому их не рекомендуется применять при $\mathrm{Rtol} < 10^{-4}$. Это подтверждается приведенными результатами, из которых видно, что при $\mathrm{Rtol} = 10^{-6}$ вычислительные затраты становятся очень большими. Совпадение результатов методов SEM1 и SEM2 в табл. 5 при $\mathrm{Rtol} = 10^{-6}$ объясняется тем, что при малом шаге задача классифицируется как нежесткая, а в этом случае оба метода используют одну и ту же формулу интегрирования. В некоторых случаях при $\mathrm{Rtol} = 10^{-2}$ потребовалось заметно больше вычислений правой части, чем при $\mathrm{Rtol} = 10^{-3}$ (такие результаты выделены в таблицах жирным шрифтом). Причиной такого аномального поведения вычислительных затрат является неустойчивость управления шагом интегрирования, которая иногда возникает при низкой задаваемой точности.

Таблица 2

Результаты решения задачи OREGO

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	0.24	2 091	0.65	4 838	1.17	11 501	2.50	55 945
AM2	1.25	2 479	2.46	5 140	3.68	12 034	5.70	57 940
SEM1	0.00	17 760	0.01	32 331	0.25	66 573	1.43	378 742
SEM2	-0.86	20 739	0.17	28 471	0.98	47 062	2.54	244 919
DIRK2	0.42	1 593	1.08	2 773	1.77	5 408	3.14	23 129
ode23s	0.52	1 501	1.30	3 363	2.10	8 261	3.76	49 905

Таблица 3

Результаты решения задачи HIRES

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	0.34	846	0.93	1 498	1.69	3 325	4.15	20 728
AM2	1.29	1 368	2.02	1 992	2.99	3 875	5.58	22 563
SEM1	1.02	1 481	1.81	3 451	2.56	8 657	3.82	71 232
SEM2	1.44	1 230	2.29	2 785	3.27	7 932	6.39	71 019
DIRK2	1.82	551	2.56	980	3.21	2 000	4.64	19 068
ode23s	2.36	926	3.27	2 515	4.06	6 519	5.47	41 115

Таблица 4

Результаты решения задачи CUSP

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	0.72	1 271	1.73	1 603	2.05	3 707	5.78	29 134
AM2	2.68	**4 187**	3.84	1 601	5.23	4 282	8.16	27 992
SEM1	0.22	2 041	0.70	4 717	1.21	11 804	1.82	42 152
SEM2	2.89	3 561	3.00	5 517	3.50	10 187	4.07	39 024
DIRK2	1.85	5 620	2.62	9 353	3.38	13 676	4.76	44 015
ode23s	1.72	8 479	2.59	15 517	3.61	36 449	5.57	230 127

Сравнение с результатами, приведенными в [4], показывает, что методы AM1 и AM2 заметно превосходят рассмотренные в [4] одношаговые методы A1, A2 и A3. Исключение составляют умеренно жесткие задачи HIRES и BRUSS ($N = 100$), при решении которых одношаговые и многошаговые методы показывают сравнимые результаты. Приведенные в табл. 1–7 результаты позволяют также сравнить адаптивные и стабилизированные методы при решении различных типов задач. Адаптивные методы имеют

преимущество при решении отделимо жестких задач (так называются задачи, у которых жесткий и нежесткий спектры матрицы Якоби достаточно четко разделены [12]). К таким задачам относятся VDPOL, OREGO, CUSP и ROBER. Стабилизированные методы имеют преимущество при решении задач с равномерно заполненным спектром якобиана. К таким задачам относится BRUSS. Задача HIRES занимает промежуточное положение между отделимо жесткими задачами и задачами с распределенным спектром якобиана, поэтому адаптивные и стабилизированные методы показали при ее решении сравнимые результаты.

Таблица 5

Результаты решения задачи BRUSS ($N = 100$)

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	1.58	1 766	1.89	2 305	2.42	4 171	5.57	11 001
AM2	2.43	3 195	3.69	3 238	4.46	4 030	8.11	12 887
SEM1	0.98	637	1.47	1 152	1.97	2 274	6.40	21 477
SEM2	2.48	641	2.57	867	4.15	2 714	6.40	21 477
DIRK2	1.78	2 529	2.37	3 655	3.03	4 105	4.31	6 479
ode23s	1.78	5 892	2.39	11 170	3.06	29 942	4.36	103 937

Таблица 6

Результаты решения задачи BRUSS ($N = 500$)

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	2.00	37 185	2.47	37 955	2.84	40 477	3.46	66 754
AM2	2.20	78 861	3.37	78 758	4.48	78 770	5.70	80 731
SEM1	0.59	1 708	1.08	2 988	1.66	6 235	2.88	28 092
SEM2	1.44	2 176	2.40	3 060	3.37	4 858	5.50	26 030
DIRK2	1.70	11 124	2.39	16 258	3.03	18 507	4.32	22 287
ode23s	1.78	29 092	2.39	55 170	3.05	113 342	4.43	606 912

Таблица 7

Результаты решения задачи ROBER

Метод	$\mathrm{Rtol} = 10^{-2}$		$\mathrm{Rtol} = 10^{-3}$		$\mathrm{Rtol} = 10^{-4}$		$\mathrm{Rtol} = 10^{-6}$	
	scd	Nf	scd	Nf	scd	Nf	scd	Nf
AM1	1.43	2 272	1.90	5 702	2.38	16 269	3.18	152 328
AM2	2.23	**20 304**	3.26	10 780	4.18	16 191	6.22	153 716
A4	1.64	**4 230**	2.43	3 497	4.50	4 060	6.09	4 897
DIRK2	1.89	847	2.48	1 683	3.13	3 539	4.44	16 180
ode23s	1.96	807	2.81	1 731	3.19	4 115	4.73	82 236

При сравнении с неявными методами следует учесть, что предложенные методы значительно проще неявных, что уже является их преимуществом. Сравнение вычислительных затрат при примерно одинаковой фактической точности scd показывает, что метод AM2 превосходит по этому показателю неявные методы на задачах VDPOL, OREGO и CUSP, а метод SEM2 превосходит неявные методы на задачах CUSP и BRUSS. Вычислительные затраты неявных методов при решении больших задач CUSP и BRUSS можно значительно уменьшить, если использовать не численное, а аналитическое вычисление якобиана, но в этом случае пользователь должен сам составить и запрограммировать выражения для расчета элементов матрицы Якоби.

Сравнение методов разных порядков показывает, что методы 1-го порядка AM1 и SEM1 уступают явным и неявным методам 2-го порядка. Показательны результаты метода переменного порядка A4,

приведенные в [4] и в табл. 7 настоящей статьи. В этом методе реализованы многошаговые формулы порядков от 1-го до 5-го (в том числе и формулы, использованные в методах AM1 и AM2). Благодаря этому он показывает хорошие результаты в широком диапазоне требований к точности, не уступая на многих задачах неявным методам. Можно предположить, что усовершенствование методов SEM1 и SEM2 (повышение порядка, переменный порядок) также позволит значительно повысить эффективность методов такого типа. Однако построение таких методов встречает серьезные трудности.

СПИСОК ЛИТЕРАТУРЫ

1. *Скворцов Л.М.* Адаптивные методы численного интегрирования в задачах моделирования динамических систем // Изв. РАН. Теория и системы управления. 1999. № 4. 72–78.
2. *Скворцов Л.М.* Явные адаптивные методы численного решения жестких систем // Матем. моделирование. 2000. **12**, № 12. 97–107.
3. *Скворцов Л.М.* Явный многошаговый метод численного решения жестких дифференциальных уравнений // Журн. вычисл. матем. и матем. физики. 2007. **47**, № 6. 959–967.
4. *Скворцов Л.М.* Простые явные методы численного решения жестких обыкновенных дифференциальных уравнений // Вычислительные методы и программирование. 2008. **9**. 154–162.
5. *Скворцов Л.М.* Точность методов Рунге–Кутты при решении жестких задач // Журн. вычисл. матем. и матем. физики. 2003. **43**, № 9. 1374–1384.
6. *Скворцов Л.М.* Явные методы Рунге–Кутты для умеренно жестких задач // Журн. вычисл. матем. и матем. физики. 2005. **45**, № 11. 2017–2030.
7. *Козлов О.С., Скворцов Л.М.* Тестовое сравнение решателей ОДУ системы MATLAB // Всероссийская научная конференция "Проектирование научных и инженерных приложений в среде MATLAB". М.: Изд-во ИПУ РАН, 2002. 53–60 (http://matlab.exponenta.ru/conf2002/proceedings.php).
8. *Козлов О.С., Скворцов Л.М., Ходаковский В.В.* Решение дифференциальных и дифференциально-алгебраических уравнений в программном комплексе "МВТУ" (http://model.exponenta.ru/mvtu/20051121.html).
9. *Лебедев В.И.* Как решать явными методами жесткие системы дифференциальных уравнений // Вычислительные процессы и системы. Вып. 8. М.: Наука, 1991. 237–291.
10. *Лебедев В.И., Медовиков А.А.* Явный метод второго порядка точности для решения жестких систем обыкновенных дифференциальных уравнений // Изв. вузов. Математика. 1998. № 9. 55–63.
11. *Sommeijer B.P., Shampine L.F., Verwer J.D.* RKC: An explicit solver for parabolic PDEs // J. Comput. Appl. Math. 1997. **88**, N 2. 315–326.
12. *Хайрер Э., Ваннер Г.* Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. М.: Мир, 1999.
13. *Abdulle A.* Fourth order Chebyshev methods with recurrence relation // SIAM J. Sci. Comput. 2002. **23**, N 6. 2041–2054.
14. *Цыпкин Я.З.* Основы теории автоматических систем. М.: Наука, 1977.

Поступила в редакцию
22.09.2008