Сибирский журнал
индустриальной математики
Июль–сентябрь, 2011. Том XIV, № 3(47)

УДК 519.622.2

ВЫЧИСЛИТЕЛЬНАЯ ПОГРЕШНОСТЬ МЕТОДА ЭЙЛЕРА ПРИ ВЫЧИСЛЕНИЯХ В АРИФМЕТИКЕ С ПЛАВАЮЩЕЙ ТОЧКОЙ

Е. А. Калинина, О. Н. Самарина

Приводится алгоритм, позволяющий найти оптимальное в смысле вычислительной точности число шагов метода Эйлера для решения задачи Коши для системы линейных дифференциальных уравнений с постоянными коэффициентами. Даются численные примеры применения указанного метода для нахождения значения решения задачи Коши в точке и построения решений системы нелинейных обыкновенных дифференциальных уравнений.

Ключевые слова: метод Эйлера, задача Коши, система обыкновенных дифференциальных уравнений, арифметика с плавающей точкой, вычислительная погрешность.

1. Постановка задачи

Во многих случаях для интегрирования систем обыкновенных дифференциальных уравнений (ОДУ) используется метод Эйлера. Применение метода Эйлера имеет смысл в тех задачах, в которых система дифференциальных уравнений (как правило, с очень большим числом уравнений) задается не матрицей, а в виде списка элементов структуры, эквивалентной схемы, эскиза или чертежа конструкции. Такие системы уравнений возникают при проектировании различных АСУ, систем прогнозирования, тренажеров и подобных систем. В качестве примера такой задачи приведем построение аппаратного ускорителя Awsin [1], которое основано на интегрировании системы дифференциальных уравнением с помощью явного метода Эйлера с постоянным шагом.

В этом случае возникает задача определения шага метода Эйлера, при котором погрешность решения будет минимальной. Если взять очень большое число шагов, то при довольно малой погрешности самого метода увеличится вычислительная погрешность из-за накопления ошибок округления [2–7]. При малом числе шагов значительной оказывается погрешность метода. Ввиду достаточно большой сложности задачи [2–5, 8] ранее использовались другие подходы: вероятностный [9], поиск точного решения системы [10]. Однако ошибки округления не являются независимыми случайными величинами (см. [5]).

В настоящей работе приводится алгоритм, позволяющий найти число шагов метода Эйлера, при котором относительная погрешность вычисленного решения минимальна. Показано, что для ряда систем интегрирование с помощью метода Эйлера дает довольно точный результат и является допустимым. Приводятся численные примеры применения рассмотренного метода для нахождения значения решения задачи Коши в точке и для построения решений систем нелинейных ОДУ.

Как известно, системы ОДУ порядков выше первого сводятся к системам ОДУ первого порядка (см., например, [1]). Наиболее важным является случай, когда система ОДУ первого порядка разрешима относительно производных. Для таких систем, имеющих нормальную форму Коши, и применяется метод

Эйлера. В дальнейшем мы будем рассматривать системы линейных уравнений с постоянными коэффициентами.

Будем использовать следующую норму вектора: если $X = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$, то $\|X\| = |x_1| + |x_2| + \cdots + |x_m|$, а также соответствующую матричную норму $\|A\| = \max\limits_{1 \le j \le m} \|A_j\|$, где $A = (A_1, A_2, \ldots, A_m)$ — матрица порядка m.

Рассмотрим задачу Коши для системы линейных дифференциальных уравнений

$$\dot{X} = AX, \quad X(t_0) = X_0, \quad X_0 \neq 0, \tag{1}$$

здесь $A = [a_{ij}]_{i,j=1}^{m}$ — вещественная квадратная матрица порядка m с постоянными элементами, $X(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{pmatrix}$, $X_0 = \begin{pmatrix} x_1^0 \\ \vdots \\ x_m^0 \end{pmatrix}$. Считаем, что все элементы матрицы A и вектора X_0 заданы точно.

Решение задачи (1) имеет вид $X = e^{A(t-t_0)}X_0$. При данном числе шагов n, применяя метод Эйлера в точной арифметике, получим следующее приближенное значение решения в точке $t_1 = t_0 + 1$:

$$\bar{X}(t_1) = \left(E + \frac{1}{n}A\right)^n X_0. \tag{2}$$

Обозначим решение, вычисленное методом Эйлера в арифметике с плавающей точкой, через $\widehat{X}(t_1)$. Обозначим относительную погрешность вычисления j-й компоненты вектора $\widehat{X}(t_1)$ по формуле (2) через $\delta_1(n)$. Норма вектора полной относительной погрешности, компоненты которого равны полным относительным погрешностям компонент вектора $\bar{X}(t_1)$, находится по формуле

$$F(n) = \sum_{j=1}^{m} \left| \frac{x_j(t_1) - \bar{x}_j(t_1)}{\bar{x}_j(t_1)} \right| + \varepsilon mn. \tag{3}$$

Таким образом, требуется найти значение n, при котором достигается минимум функции $F(n)$.

Замечание 1. Здесь $\varepsilon = 2\upsilon$, где υ (unit roundoff) — максимальная относительная погрешность вычислений (так, для float (4 байта) $\varepsilon \approx 1,19 \cdot 10^{-7}$, для double (8 байт) $\varepsilon \approx 2,22 \cdot 10^{-16}$, для long double (10 или 12 байт в зависимости от системы) $\varepsilon \approx 1,08 \cdot 10^{-19}$). Значения ε взяты из стандартного включаемого файла float.h для C-компилятора на архитектуре x86.

Скалярное произведение векторов $U_{m\times 1}$ и $V_{m\times 1}$ необходимо вычислить в так называемой расширенной точности (промежуточные вычисления выполняются точно с вдвое более длинной мантиссой). При этом согласно [2] для вычислительной погрешности справедлива формула

$$|U^TV - fl(fl_e(U^TV))| \le \upsilon|U^TV| + \frac{m\upsilon_e}{1-\upsilon_e}(1+\upsilon)|U|^T|V|,$$

где $|U|$ — вектор с компонентами, равными модулям компонент вектора U, υ_e — максимальная относительная погрешность вычисления с длинной мантиссой, $fl(a)$ — результат представления числа a в арифметике с плавающей точкой на компьютере, $fl_e(a)$ — представление a с длинной мантиссой. При выполнении условия $m\upsilon_e|U|^T|V| \le \upsilon|U^TV|$ вычисленное скалярное произведение почти так же точно, как и округленное его точное значение. Отсюда получаем (3).

Когда хотя бы одна компонента решения $\widehat{X}(t_1)$ близка к нулю, будем пользоваться полной абсолютной погрешностью, которая находится по формуле

$$\Delta(n)=\sum_{j=1}^{m}|x_j(t_1)-\bar{x}_j(t_1)|+\varepsilon mn\sum_{j=1}^{m}|\bar{x}_j(t_1)|. \tag{4}$$

2. Оптимальное число шагов метода Эйлера

Рассмотрим общий случай: ни одна из компонент вектора $\widehat{X}(t_1)$ не обращается в нуль.

Определение. Назовем оптимальным числом шагов метода Эйлера число шагов n_{opt}, при котором значение решения в точке находится с наименьшей полной относительной погрешностью. Оптимальное число шагов метода Эйлера можно найти, воспользовавшись следующей теоремой.

Теорема 1. *Оптимальное число шагов n_{opt} для нахождения значения решения системы дифференциальных уравнений* (1) *при $t_1 = t_0 + 1$ приближенно может быть найдено по формуле*

$$n_{\text{opt}} \approx \sqrt{\left(\left|\frac{\ddot{x}_1(t_1)}{x_1(t_1)}\right| + \left|\frac{\ddot{x}_2(t_1)}{x_2(t_1)}\right| + \cdots + \left|\frac{\ddot{x}_m(t_1)}{x_m(t_1)}\right|\right) \Big/ 2m\varepsilon}. \tag{5}$$

Доказательство. Пусть A_J — жорданова нормальная форма матрицы A, а $T = [t_{ij}]_{i,j=1}^{m}$ — матрица, составленная из векторов канонического базиса, такая, что $A_J = T^{-1}AT$. Тогда равенство (2) можно записать в виде

$$\widehat{X}(t_1) \approx T\Big(E + \frac{1}{n}A_J\Big)^n T^{-1}X_0. \tag{6}$$

Рассмотрим сначала случай различных собственных чисел матрицы A. Пусть A — квадратная матрица порядка m. Предположим, что A имеет l пар комплексно-сопряженных собственных чисел $\mu_1, \bar{\mu}_1, \ldots, \mu_l, \bar{\mu}_l$ и $m - 2l$ вещественных собственных числа $\lambda_1, \lambda_2, \ldots, \lambda_{m-2l}$. Для комплексных собственных чисел будем также использовать тригонометрическую форму $\mu_j = \rho_j(\cos\zeta_j + i\sin\zeta_j)$, $j = 1, 2, \ldots, l$, причем считаем $\sin\zeta_j > 0$. Тогда согласно (6) j-я компонента вектора $X(t_1)$ (решения задачи Коши (1)) будет иметь вид

$$c_1^{(j)}e^{\lambda_1} + c_2^{(j)}e^{\lambda_2} + \cdots + c_{m-2l}^{(j)}e^{\lambda_{m-2l}} + d_1^{(j)}e^{\mu_1} + \bar{d}_1^{(j)}e^{\bar{\mu}_1} + \cdots + d_l^{(j)}e^{\mu_l} + \bar{d}_l^{(j)}e^{\bar{\mu}_l},$$

где постоянные коэффициенты $c_k^{(j)}$, $d_k^{(j)}$, $\bar{d}_1^{(j)}$ зависят от элементов матрицы T. Соответственно j-я компонента вектора $\bar{X}(t_1)$ имеет вид

$$\begin{aligned} &c_1^{(j)}\Big(1+\frac{\lambda_1}{n}\Big)^n + \cdots + c_{m-2l}^{(j)}\Big(1+\frac{\lambda_{m-2l}}{n}\Big)^n \\ &\quad + d_1^{(j)}\Big(1+\frac{\mu_1}{n}\Big)^n + \bar{d}_1^{(j)}\Big(1+\frac{\bar{\mu}_1}{n}\Big)^n + \cdots + d_l^{(j)}\Big(1+\frac{\mu_l}{n}\Big)^n + \bar{d}_l^{(j)}\Big(1+\frac{\bar{\mu}_l}{n}\Big)^n. \end{aligned} \tag{7}$$

$$F(n) = \sum_{j=1}^{m}\left|\frac{c_1^{(j)}e^{\lambda_1} + \cdots + c_{m-2l}^{(j)}e^{\lambda_{m-2l}} + d_1^{(j)}e^{\mu_1} + \bar{d}_1^{(j)}e^{\bar{\mu}_1} + \cdots + d_l^{(j)}e^{\mu_l} + \bar{d}_l^{(j)}e^{\bar{\mu}_l}}{\sum\limits_{k=1}^{m-2l} c_k^{(j)}\left(1+\frac{\lambda_k}{n}\right)^n + \sum\limits_{k=1}^{l}\left(d_k^{(j)}\left(1+\frac{\mu_k}{n}\right)^n + \bar{d}_k^{(j)}\left(1+\frac{\bar{\mu}_k}{n}\right)^n\right)} - 1\right| + \varepsilon mn$$

или

$$F(n) = \sum_{j=1}^{m} \left| \frac{\sum\limits_{k=1}^{m-2l} c_k^{(j)} e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)} e^{\rho_k \cos\zeta_k} \cos(\rho_k \sin\zeta_k + \chi_k^{(j)})}{\sum\limits_{k=1}^{m-2l} c_k^{(j)} \left(1+\frac{\lambda_k}{n}\right)^n + \sum\limits_{k=1}^{l} 2r_k^{(j)} \left(1+\frac{\rho_k^2}{n^2}+2\frac{\rho_k\cos\zeta_k}{n}\right)^{n/2} \cos(\omega_k n + \chi_k^{(j)})} - 1 \right| + \varepsilon mn,$$

где

$$d_k^{(j)} = r_k^{(j)}\big(\cos\chi_k^{(j)} + i\sin\chi_k^{(j)}\big), \quad \omega_k = \arg\Big(1+\frac{\mu_k}{n}\Big) = \arccos\frac{1+\rho_k\cos\zeta_k/n}{\sqrt{1+\frac{\rho_k^2}{n^2}+2\frac{\rho_k\cos\zeta_k}{n}}}$$

с учетом положительности $\sin\zeta_k$, $k = 1, 2, \ldots, l$.

Для доказательства теоремы нам понадобятся следующие формулы:

$$c_k^{(j)}\Big(1+\frac{\lambda_k}{n}\Big)^n = c_k^{(j)} e^{\lambda_k} - c_k^{(j)} e^{\lambda_k}\frac{\lambda_k^2}{2n} + o\Big(\frac{1}{n}\Big),$$

$$d_k^{(j)}\Big(1+\frac{\mu_k}{n}\Big)^n + \bar{d}_k^{(j)}\Big(1+\frac{\bar{\mu}_k}{n}\Big)^n = 2r_k^{(j)} e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + \chi_k^{(j)}\big) - \frac{1}{n} r_k^{(j)}\rho_k^2 e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + 2\zeta_k + \chi_k^{(j)}\big) + o\Big(\frac{1}{n}\Big). \quad (8)$$

Разложим $F(n)$ в сходящийся ряд по степеням $1/n$ до членов второго порядка, используя равенства (8). Получим функцию $\widehat{F}(n)$:

$$\widehat{F}(n) = \frac{1}{2n}\sum_{j=1}^{m} \left| \frac{\sum\limits_{k=1}^{m-2l} c_k^{(j)}\lambda_k^2 e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)}\rho_k^2 e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + 2\zeta_k + \chi_k^{(j)}\big)}{\sum_{k=1}^{m-2l} c_k^{(j)} e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)} e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + \chi_k^{(j)}\big)} \right| + \varepsilon mn.$$

Продифференцируем $\widehat{F}(n)$ по n и приравняем полученное выражение к нулю. Получим приближенное уравнение для нахождения n:

$$n \approx \sqrt{\sum_{j=1}^{m} \left| \frac{\sum\limits_{k=1}^{m-2l} c_k^{(j)}\lambda_k^2 e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)}\rho_k^2 e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + 2\zeta_k + \chi_k^{(j)}\big)}{\sum\limits_{k=1}^{m-2l} c_k^{(j)} e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)} e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + \chi_k^{(j)}\big)} \right| \Big/ 2m\varepsilon}. \quad (9)$$

Осталось заметить, что

$$\frac{\ddot{x}_j(t)}{x_j(t)}\bigg|_{t=t_0+1} = \frac{\sum\limits_{k=1}^{m-2l} c_k^{(j)}\lambda_k^2 e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)}\rho_k^2 e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + 2\zeta_k + \chi_k^{(j)}\big)}{\sum\limits_{k=1}^{m-2l} c_k^{(j)} e^{\lambda_k} + \sum\limits_{k=1}^{l} 2r_k^{(j)} e^{\rho_k\cos\zeta_k}\cos\big(\rho_k\sin\zeta_k + \chi_k^{(j)}\big)}.$$

Тем самым справедливость формулы (5) установлена.

Теперь предположим, что у матрицы A имеются кратные собственные числа. В этом случае нам понадобятся формулы

$$\frac{(n-1)(n-2)\dots(n-p_k+1)}{p_k!n^{p_k-1}}\left(1+\frac{\lambda_k}{n}\right)^{n-p_k} = \frac{e^{\lambda_k}}{p_k!}\left(1-\frac{\lambda_k^2+2p_k\lambda_k+p_k(p_k-1)}{2n}\right)+o\left(\frac{1}{n}\right), \quad (10)$$

$$d_k^{(j)}\frac{(n-1)(n-2)\dots(n-p_k+1)}{p_k!n^{p_k-1}}\left(1+\frac{\mu_k}{n}\right)^{n-p_k} + \bar{d}_k^{(j)}\frac{(n-1)(n-2)\dots(n-p_k+1)}{p_k!n^{p_k-1}}\left(1+\frac{\bar{\mu}_k}{n}\right)^{n-p_k}$$
$$= \frac{2r_k^{(j)}}{p_k!}e^{\rho_k\cos\zeta_k}\cos(\rho_k\sin\zeta_k+\chi_k^{(j)}) - \frac{r_k^{(j)}}{np_k!}e^{\rho_k\cos\zeta_k}\left[\rho_k^2\cos\left(\rho_k\sin\zeta_k+\chi_k^{(j)}+2\zeta_k\right)\right.$$
$$\left.+2p_k\rho_k\cos\left(\rho_k\sin\zeta_k+\chi_k^{(j)}+\zeta_k\right)+p_k(p_k-1)\cos\left(\rho_k\sin\zeta_k+\chi_k^{(j)}\right)\right]+o\left(\frac{1}{n}\right) \quad (11)$$

в обозначениях, введенных ранее. Аналогично рассмотренному случаю получаем, что формула (5) остается справедливой и в случае наличия кратных собственных чисел у матрицы A.

Осталось проверить, что производные функций $F(n)$ и $\widehat{F}(n)$ достаточно близки. Для произвольного простого собственного числа ν матрицы A имеем

$$\left|o'\left(\frac{1}{n}\right)\right| = |\mathfrak{c}e^{\nu}|\left|\frac{8\nu^3+3\nu^4}{12n^3}+o\left(\frac{1}{n^4}\right)\right|,$$

для кратного собственного числа ν кратности p имеем

$$\left|o'\left(\frac{1}{n}\right)\right| = \left|\frac{c}{e^{\nu}}\right|\left|\frac{-3\nu^4-4(3p+2)\nu^3-3p(5p+3)\nu^2+6p(1-p^2)\nu-p(3p^3-10p^2+9p-2)}{12p!n^3} + o\left(\frac{1}{n^4}\right)\right|,$$

откуда сразу же следует требуемое. Скорость убывания $o\left(\frac{1}{n}\right)$ пропорциональна $1/n^3$.

Замечание 2. Заметим, что нормальная форма Жордана матрицы A не вычисляется и нигде в алгоритме не используется.

Пусть $s_j(t) = \operatorname{sign}(\ddot{x}_j(t)/x_j(t))$, $j = 1, 2, \dots, m$. Здесь $\operatorname{sign} y$ — знак числа y.

Так как $\ddot{X}(t) = A^2X(t)$, справедливо следующее утверждение.

Следствие 1. *Оптимальное число шагов n_{opt} для нахождения значения решения системы дифференциальных уравнений* (1) *при $t_1 = t_0+1$ приближенно может быть найдено по формуле*

$$n_{\text{opt}} \approx \sqrt{\frac{1}{2m\varepsilon}\left(\frac{s_1(t_1)}{x_1(t_1)}, \frac{s_2(t_1)}{x_2(t_1)}, \dots, \frac{s_m(t_1)}{x_m(t_1)}\right)A^2\begin{pmatrix}x_1(t_1)\\x_2(t_1)\\\dots\\x_m(t_1)\end{pmatrix}}. \quad (12)$$

Теперь предположим, что элементы матрицы A заданы с относительными погрешностями, не превосходящими δA, а элементы вектора X_0 — с относительными погрешностями, не превосходящими δX_0. В этом случае относительная вычислительная погрешность вычислений для любой компоненты вектора $\bar{X}(t_1)$ равна $(\varepsilon+\delta A)n+\delta X_0$.

Следствие 2. *Оптимальное число шагов n_{opt} для нахождения значения решения системы дифференциальных уравнений* (1) *при $t_1 = t_0+1$ приближенно может быть найдено по формуле*

$$n_{\text{opt}} \approx (1+\delta A)\sqrt{\left(\frac{s_1(t_1)}{x_1(t_1)}, \frac{s_2(t_1)}{x_2(t_1)}, \ldots, \frac{s_m(t_1)}{x_m(t_1)}\right) A^2 \begin{pmatrix} x_1(t_1) \\ \ldots \\ x_m(t_1) \end{pmatrix} \Big/ 2m(\varepsilon+\delta A)} \quad (13)$$

в обозначениях следствия 1, *где δA — максимальная относительная погрешность элементов матрицы A.*

Доказательство следствия 2 аналогично доказательству теоремы 1.

Определение 1. Относительной погрешностью метода Эйлера на шаге (локальной погрешностью усечения) называется величина

$$\left\| \left(\frac{x_j(t_0+1/n) - \bar{x}_j(t_0+1/n)}{\bar{x}_j(t_0+1/n)} \right) \right\|_{j=1,\ldots,m} = \sum_{j=1}^{m} \left| \frac{x_j(t_0+1/n) - \bar{x}_j(t_0+1/n)}{\bar{x}_j(t_0+1/n)} \right|.$$

Определение 2. Относительной вычислительной погрешностью метода Эйлера на шаге (локальной относительной вычислительной погрешностью) называется величина, равная накопленной относительной вычислительной погрешности арифметических операций за выполнение одного шага метода.

Тем самым относительная вычислительная погрешность метода Эйлера на шаге равна в нашем случае $m\varepsilon$.

Теорема 2. *Для оптимального числа шагов n_{opt} относительная погрешность метода Эйлера на шаге совпадает с относительной вычислительной погрешностью на шаге.*

Доказательство. Пусть $\delta\xi$ — вектор относительных погрешностей решения на одном шаге, т. е. j-я компонента $\delta\xi$ равна отношению j-й компоненты вектора $\bar{X}(t_1) - e^{A/n}X_0$ к j-й компоненте $\bar{X}(t_1)$.

С учетом (8), (10), (11) приближенное равенство $\|\delta\xi\| \approx m\varepsilon$ равносильно приближенному равенству (5).

Определение 3. Назовем оптимальным шагом метода Эйлера шаг интегрирования, при котором относительная погрешность метода Эйлера на шаге совпадает с относительной вычислительной погрешностью на шаге.

Замечание 3. При нахождении значения решения задачи Коши (1) в точке $t_1 = t_0 + \tau$ методом Эйлера имеем $\bar{X}(t_1) = (E + A\tau/n)^n X_0$. Таким образом, этот случай сводится к случаю $\tau = 1$ заменой матрицы A на матрицу $A\tau$. Следовательно, достаточно рассмотреть только случай $\tau = 1$.

Из формулы (13) и замечания 3 следует, что погрешность элементов матрицы A влияет не только на значение решения в точке, но и на саму точку, в которой вычисляется решение. Действительно, поскольку матрица системы становится равной $A(1 \pm \delta A)$, то вместо точки $t_0 + 1$ значение решения вычисляется в точке $t_0 + 1 \pm \delta A$.

3. Вычислительный алгоритм

Предложенный здесь алгоритм позволяет не только найти оптимальное число шагов метода Эйлера, но и вычислить значение решения задачи Коши максимально точно.

Воспользуемся методом Банаха (простых итераций). Сначала найдем приближенное значение $X_1(t_1)$, которое получается интегрированием с помощью метода Эйлера с достаточно большим числом шагов n_1. Затем по найденному

значению $X_1(t_1)$ определим следующее значение n_2 числа шагов метода Эйлера и построим новое решение с числом шагов n_2 и т. д. Каждое следующее значение n_{k+1} числа шагов будем находить по формуле

$$n_{k+1} = \left\lceil \sqrt{\left(\frac{s_1^{(k)}(t_1)}{x_1^{(k)}(t_1)}, \ldots, \frac{s_m^{(k)}(t_1)}{x_m^{(k)}(t_1)} \right) A^2 X_k(t_1)/2m\varepsilon} \right\rceil, \; X_k(t_1) = \begin{pmatrix} x_1^{(k)}(t_1) \\ x_2^{(k)}(t_1) \\ \ldots\ldots \\ x_m^{(k)}(t_1) \end{pmatrix}. \quad (14)$$

Здесь для вещественного α величина $\lceil \alpha \rceil$ обозначает наименьшее целое число, которое больше или равно α.

Теорема 3. *Последовательность* (14) *сходится, если* n_1 *достаточно велико.*

Доказательство. Обозначим $\Delta n{=}n_k{-}n_{\rm opt}$ и $\delta X_{\rm total}{=}\left(\dfrac{\Delta x_1}{x_1}, \ldots, \dfrac{\Delta x_m}{x_m} \right)^T$, где $\Delta x_j = x_j(n_k) - x_j(n_{\rm opt}) = x_j(n_k) - x_j$; через $A^{2\,(j)}$ и A_k^2, $j,k = 1,2,\ldots,m$, обозначим соответственно j-ю строку и k-й столбец матрицы A^2. Далее будем пользоваться равенством

$$\frac{1}{x_j + \Delta x_j} = \frac{1}{x_j} - \frac{\Delta x_j}{x_j^2} + o\left(\frac{\Delta x_j}{x_j} \right), \quad j = 1, \ldots, m.$$

Поскольку $\sqrt{a+\Delta a} - \sqrt{a} = \dfrac{\Delta a}{2\sqrt{a}} + o\left(\dfrac{\Delta a}{\sqrt{a}} \right)$, то для оценки близости некоторого n и $n_{\rm opt}$ рассмотрим разность

$$\left(\frac{s_1}{x_1 + \Delta x_1}, \ldots, \frac{s_m}{x_m + \Delta x_m} \right) A^2 \begin{pmatrix} x_1 + \Delta x_1 \\ \vdots \\ x_m + \Delta x_m \end{pmatrix} - \left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) A^2 \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

$$= -\left(\frac{\Delta x_1}{x_1}, \ldots, \frac{\Delta x_m}{x_m} \right) \begin{pmatrix} s_1/x_1 A^{2\,(1)} \\ \vdots \\ s_m/x_m A^{2\,(m)} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

$$+ \left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) (x_1 A_1^2, \ldots, x_m A_m^2) \begin{pmatrix} \Delta x_1/x_1 \\ \vdots \\ \Delta x_m/x_m \end{pmatrix} + o(\|\delta X_{\rm total}\|).$$

По модулю она не превосходит

$$\left\| (1, \ldots, 1) \begin{pmatrix} s_1/x_1 A^{2(1)} \\ \vdots \\ s_m/x_m A^{2(m)} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \right\| \|\delta X_{\rm total}\|$$

$$+ \left\| \left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) (x_1 A_1^2, \ldots, x_m A_m^2) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\| \|\delta X_{\rm total}\| + o(\|\delta X_{\rm total}\|)$$

$$= 4m\varepsilon n_{\rm opt}^2 \|\delta X_{\rm total}\| + o(\|\delta X_{\rm total}\|).$$

Так как

$$(1, \ldots, 1) \begin{pmatrix} s_1/x_1 A^{2(1)} \\ \vdots \\ s_m/x_m A^{2(m)} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) A^2 \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = 2m\varepsilon n_{\rm opt}^2,$$

$$\left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) (x_1 A_1^2, \ldots, x_m A_m^2) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \left(\frac{s_1}{x_1}, \ldots, \frac{s_m}{x_m} \right) A^2 \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = 2m\varepsilon n_{\rm opt}^2,$$

то $|n_k - n_{\text{opt}}| \le \sqrt{2m\varepsilon}\, n_{\text{opt}}\|\delta X_{\text{total}}\| + o(\|\delta X_{\text{total}}\|)$.

Для оценки $\|\delta X_{\text{total}}\|$ воспользуемся равенствами (8), (10), (11). Имеем

$$\|\delta X_{\text{total}}\| \le 2m\varepsilon n_{\text{opt}}^2 \left|\frac{1}{n_{k-1}} - \frac{1}{n_{\text{opt}}}\right| + 2m\varepsilon + o\left(\frac{1}{n_{k-1}}\right) + o\left(\frac{1}{n_{\text{opt}}}\right).$$

Следовательно, при достаточно большом n_k с уменьшением относительной погрешности решения $\|\delta X_{\text{total}k}\|$ следующее значение n_{k+1} становится ближе к n_{opt}, а при приближении n_k к n_{opt} относительная погрешность приближенного решения уменьшается.

Замечание 4. Так как для правой части системы уравнений (1) выполняется условие Липшица с $L = \|A\|$, а кроме того, при $\tau \le 1$ [12]

$$\left\| AX(t) - \frac{X(t+\tau) - X(t)}{\tau} \right\| \le \frac{\tau}{2}\|A\|^2 e^{\|A\|}\|X_0\|,$$

то абсолютная погрешность метода Эйлера при $t_1 = t_0 + 1$ для данного числа шагов n по норме не превосходит $\|\Delta(n)\| \le \frac{\|A\|e^{\|A\|}}{2n}(e^{||A||} - 1)\|X_0\|$. Тем самым n можно выбрать так, чтобы

$$\|\delta\xi\| \min_{j=\overline{1,m}} |x_j(t_1)| \le \left\| \frac{\Delta(n)}{n} \right\| \le \frac{\|A\|e^{\|A\|}}{2n^2}(e^{\|A\|} - 1)\|X_0\| \le m\varepsilon \min_{j=\overline{1,m}} |x_j(t_1)|.$$

Поскольку $\min\limits_{j=\overline{1,m}} |x_j(t_1)| \ge \min\limits_{j=\overline{1,m}} |x_j^0| e^{-\|A\|}$, то если у вектора X_0 нет нулевых компонент, при выборе n_1 исходя из условия

$$n_1 \ge e^{\|A\|} \sqrt{\|A\|(e^{\|A\|} - 1)\|X_0\| \big/ \big(2m\varepsilon \min_{j=\overline{1,m}} |x_j^0|\big)},$$

мы заведомо получим $n_1 \ge n_{\text{opt}}$.

Однако из приведенных примеров ясно, что последовательность (14) во многих случаях сходится и при $n_1 = 1$.

Для найденного оптимального значения числа шагов интегрирования приведем некоторую оценку погрешности вычисления значения решения задачи Коши методом Эйлера в точке, которая сразу следует из теоремы 3.

Утверждение. *Для порядка матрицы A для $m < 42016$ при точности* float ($\varepsilon = 1,19 \cdot 10^{-7}$) *полная погрешность метода составит не более* 1 % *при выполнении условия* $\left(\dfrac{s_1^{(1)}(t_1)}{x_1^{(1)}(t_1)}, \ldots, \dfrac{s_m^{(1)}(t_1)}{x_m^{(1)}(t_1)}\right) A^2 X_1(t_1) < 2m\varepsilon.$

Замечание 5. В приведенном алгоритме рассмотрен общий случай, когда ни одна из компонент вектора $X_k(t_1)$ не обращается в нуль. Если в процессе вычислений какие-то компоненты вектора $X_k(t_1)$ обнуляются, то n_{opt} положим равным $\lceil \sqrt{\|A^2\|/(2m\varepsilon)} \rceil$. Действительно, переходя в этом случае к абсолютной погрешности, применяя формулы (8), (10) и (11), аналогично результату теоремы 1, получаем $n \approx \sqrt{\|\ddot{X}(t_1)\|/(2m\varepsilon\|X(t_1)\|)}$.

Воспользуемся свойствами нормы для получения верхней границы n:

$$\sqrt{\frac{\|\ddot{X}(t_1)\|}{2m\varepsilon\|X(t_1)\|}} = \sqrt{\frac{\|A^2X(t_1)\|}{2m\varepsilon\|X(t_1)\|}} \le \sqrt{\frac{\|A^2\|}{2m\varepsilon}}. \tag{15}$$

Неравенством (15) можно воспользоваться для определения n_1, когда вектор X_0 имеет хотя бы одну нулевую компоненту. В этом случае можно положить $n_1 = \lceil \sqrt{\|A^2\|/(2m\varepsilon)} \rceil$.

Гипотеза. Метод, приведенный в настоящей работе, применим и для систем дифференциальных уравнений более общего вида, в частности для систем ОДУ $\dot{X} = F(t, X)X$, как показывают примеры 2, 3. Однако обоснование этого является вопросом для дальнейших исследований.

4. Численные примеры

Рассмотрим несколько примеров.

Пример 1. Найдем решение задачи Коши $\dot{X} = AX$, $X(0) = X_0$ в точке $t_1 = 1$, где

$$A = \begin{pmatrix} -2 & 25 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 10 & 3 & 3 & 3 & 0 \\ 0 & 0 & 2 & 15 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 15 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 25 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 \end{pmatrix}, \quad X_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

Здесь матрица системы имеет две пары кратных собственных чисел: $\lambda = -2$ и $\lambda = -3$ кратности 2.

Замечание 6. Матрица A взята из [13], где приводится ее спектральный портрет. Ниже приведены вектор приближенного решения $\bar{X}(1)$, найденного с помощью предложенного метода, и вектор точного решения $X(1)$:

$$\bar{X}(1) = \begin{pmatrix} 27396,6 \\ 8060,98 \\ 5965,25 \\ 952,514 \\ 222,53 \\ 2,27395 \\ 0,0497601 \end{pmatrix}, \quad X(1) = \begin{pmatrix} 27442,2104 \\ 8072,047686 \\ 5972,466329 \\ 953,2221479 \\ 222,6731115 \\ 2,274040654 \\ 0,04978706837 \end{pmatrix}.$$

Относительная погрешность найденного решения составляет $0,006215939206$, $n_{\text{opt}} = 7483$, и это значение получается уже на втором шаге алгоритма.

Пример 2. Рассмотрим теперь жесткое дифференциальное уравнение

$$y'(x) = y^2 - y^3, \quad y(0) = 10^{-4}, \quad 0 \le x \le 20000.$$

Результаты вычислений, использующих различные методы, приведены в [14]. На рис. 1 показано решение, полученное с помощью предложенного метода. Это решение в точности совпадает с аналитическим решением данного дифференциального уравнения. Время вычислений составляет 0,10989 с с выводом на печать и менее одного тика таймера без печати, тогда как решение данного уравнения в системе Matlab с помощью различных программ занимает от 0,12 до 1,04 с. Результаты, полученные в [14], более точные, чем те, что получаются при решении стандартными средствами Matlab, однако время вычислений составляет 2,10 и 4,08 с для двух предложенных там методов. Кроме того, найденные решения не совпадают с аналитическим. Тем самым метод, представленный в настоящей статье, является наиболее эффективным. Программа расчета написана на C++ в точности float. Каждая следующая точка

графика определялась как точка, полученная с помощью оптимального шага интегрирования. После этого она становилась начальной точкой для новой задачи Коши с матрицей системы, вычисленной в этой точке.

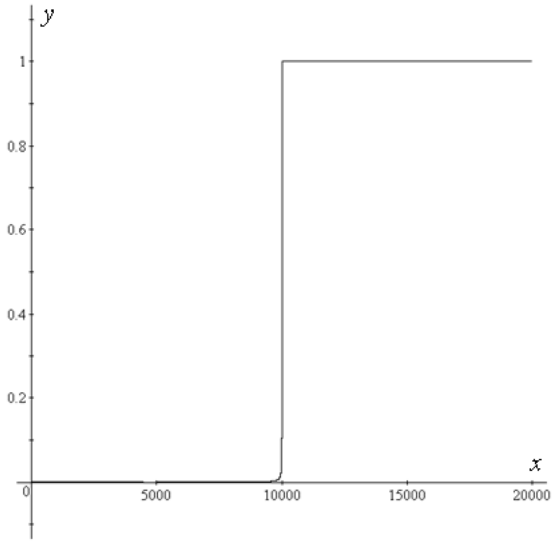

Рис. 1. Решение уравнения $y' = y^2 - y^3$ методом Эйлера

Пример 3. В качестве еще одного примера приведем построение алгебраической кривой $F(x, y) = 0$, где

$$F(x,y) = -3840x^6 + 512x^5y - 2688x^4y^2 + 256x^3y^3 - 624x^2y^4 + 32xy^5 - 48y^6 + 7552x^5 + 6784x^4y - 544x^3y^2 + 416x^2y^3 - 80xy^4 - 272y^5 + 1168x^4 - 9792x^3y + 7280x^2y^2 + 288xy^3 - 1340y^4 - 7200x^3 - 4896x^2y - 216xy^2 - 2520y^3 - 120x^2 + 5616xy - 4164y^2 + 2016x + 2016y + 441.$$

Рассмотрим систему дифференциальных уравнений $\dfrac{dx}{dt} = \dfrac{\partial F}{\partial y}$, $\dfrac{dy}{dt} = -\dfrac{\partial F}{\partial x}$. Перепишем эту систему в виде

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = A(x,y) \begin{pmatrix} x \\ y \end{pmatrix}, \tag{16}$$

где элементы матрицы A следующие:

$$a_{11}(x,y) = 2016/x + 512x^4 - 5376x^3y + 768x^2y^2 - 2496xy^3 + 160y^4 + 6784x^3 - 1088x^2y + 1248xy^2 - 320y^3 - 9792x^2 + 14560xy + 864y^2 - 4896x - 432y + 5616, \text{ если } x \neq 0;$$

$$a_{11}(x,y) = 160y^4 - 320y^3 + 864y^2 - 432y + 5616, \text{ если } x = 0;$$

$$a_{12}(x,y) = -288y^4 - 1360y^3 - 5360y^2 - 7560y - 8328, \text{ если } x \neq 0;$$

$$a_{12}(x,y) = 2016/y - 288y^4 - 1360y^3 - 5360y^2 - 7560y - 8328, \text{ если } x = 0;$$

$$a_{21}(x,y) = -2016/x - 2560x^3y + 10752x^2y^2 - 768xy^3 + 1248y^4 - 27136x^2y + 1632xy^2 - 832y^3 + 29376xy - 14560y^2 + 9792y + 23040x^4 - 4672x^2 - 37760x^3 + 21600x + 240, \text{ если } x \neq 0;$$

$$a_{21}(x,y) = 1248y^4 - 832y^3 - 14560y^2 + 9792y + 240, \text{ если } x = 0;$$
$$a_{22}(x,y) = -32y^4 + 80y^3 - 288y^2 + 216y - 5616, \text{ если } x \neq 0;$$
$$a_{22}(x,y) = -2016/y - 32y^4 + 80y^3 - 288y^2 + 216y - 5616, \text{ если } x = 0.$$

Проинтегрируем ее с помощью предложенного метода. Возможны два варианта использования рассмотренного алгоритма.

1. Фиксированы значения t, в которых матрица системы пересчитывается, а значение решения для данных t находится с помощью оптимального числа шагов.

2. В каждой точке t находится оптимальный шаг интегрирования и значение решения в новой полученной точке, если использовать этот оптимальный шаг. Матрица системы в каждой новой полученной точке пересчитывается.

Чтобы не возникала ситуация overflow, при написании программы на C++ матрица системы была домножена на 10^{-8}, тем самым уменьшился исходный шаг интегрирования. В связи с этим оптимальное число шагов практически во всех случаях стало равно единице, и два предложенных варианта для системы уравнений (16) дали совпадающие решения, которые представлены на рис. 2. Также на рис. 2 показана аналитическая кривая, которая выделена более жирно. Как видно из этого рисунка, кривые отличаются незначительно.

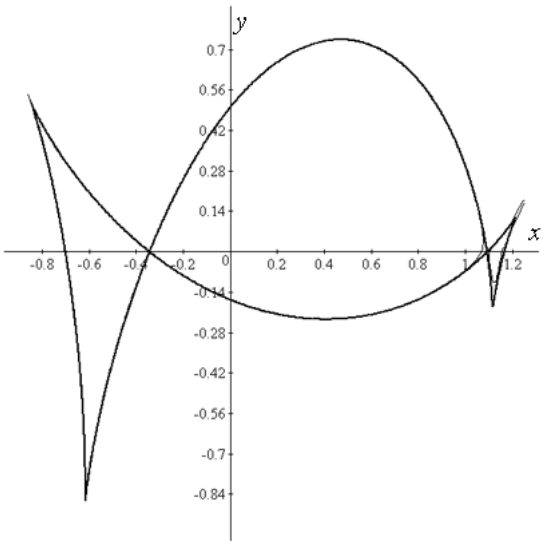

Рис. 2. Кривая $F(x,y) = 0$, построенная с помощью метода Эйлера, и аналитическая кривая

Замечание 7. Система дифференциальных уравнений

$$\frac{dX}{dt} = AX \tag{17}$$

заменой переменных $t = k\tau$, $k \neq 0$ сводится к системе

$$\frac{dX(k\tau)}{d\tau} = kAX(k\tau) \quad \text{или} \quad \frac{dZ}{d\tau} = kAZ, \quad Z(\tau) = X(k\tau), \tag{18}$$

т. е. системе такого же вида с матрицей kA.

Найдя оптимальное число шагов метода Эйлера для системы (17) по формуле (14), мы найдем также пропорциональное ему оптимальное число шагов

для системы (18). Поделив матрицу A на оптимальное для нее число шагов, получим матрицу системы, для которой оптимальным будет один шаг. Сделав его, найдем максимально точное значение решения в новой точке. Поскольку нас интересуют только координаты (x, y) точек кривой $F(x, y) = 0$ на плоскости, то учитывать значение t не нужно. С учетом того, что изначально матрица системы была поделена на 10^8, оптимальное число шагов метода Эйлера получается достаточно большим для того, чтобы иметь возможность воспользоваться теоремами 1 и 3.

Точки интегрирования были взяты приближенно как решения уравнения $F(x, y) = 0$, при этом они выбирались так, чтобы получить все части кривой.

Также система (16) была проинтегрирована с использованием второго варианта алгоритма с большим количеством шагов (порядка 3000000) для начальных данных $(0; 0, 5)$. Результат интегрирования представлен на рис. 3, где приведена и аналитическая кривая, выделенная более жирно. Программа расчета написана на C++ в точности float.

Метод Эйлера является простейшим методом интегрирования задачи Коши для системы ОДУ первого порядка. Для определенных задач существуют более эффективные методы, однако стандартный метод Эйлера довольно широко используется благодаря простоте применения и скорости. Во многих случаях вычисления методом Эйлера проводятся в стандартной процессорной арифметике с плавающей точкой. Так, например, многие задачи биофизики решаются только методом Эйлера [14].

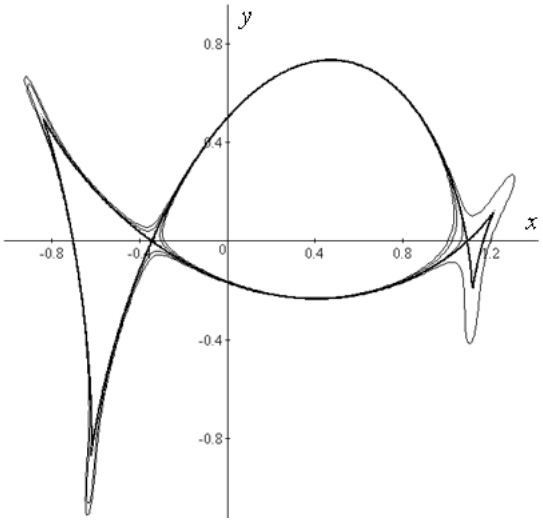

Рис. 3. Кривая $F(x, y) = 0$, полученная интегрированием методом Эйлера с большим количеством шагов, и аналитическая кривая

В настоящей работе предложена модификация метода Эйлера, которая позволяет найти максимально точное решение систем ОДУ с постоянными коэффициентами так, чтобы сумма погрешности метода и вычислительной погрешности была наименьшей. При этом сохраняются все скоростные характеристики метода. Предложенный алгоритм может быть использован для решения различных прикладных задач, в которых требуется довольно высокая точность расчетов при достаточно большой скорости вычислений, в частности при программировании в режиме реального времени.

Приводятся численные примеры использования предложенного метода.

Авторы искренне благодарны рецензенту, полезные и весьма существенные замечания и предложения которого были учтены при подготовке окончательного варианта работы.

ЛИТЕРАТУРА

1. *Lewis D. M.* A compiled-code hardware accelerator for circuit simulation // IEEE Trans. Computer-Aided Design. 1992 V. 11, N 5. P. 555–565.
2. *Higham N. J.* Accuracy and Stability of Numerical Algorithms. Philadelphia: SIAM, 1996.
3. *Golub G. H., Ortega J. M.* Scientific Computing and Differential Equations. Lomdon: Acad. Press, 1992.
4. *Ортега Дж., Пул У.* Введение в численные методы решения дифференциальных уравнений. М.: Наука, 1986.
5. *Björk A., Dahlquist G.* Numerical Mathematics and Scientific Computations. V. 1. Philadelphia: SIAM, 2008.

5. *Демидович Б. П., Марон И. А.* Основы вычислительной математики. Физматгиз, 1963.
6. *Бахвалов Н. С., Жидков Н. П., Кобельков Г. М.* Численные методы. М.: Наука, 1987.
7. *Kahaner D., Moler C., Nash S.* Numerical Methods and Software. Englewood Cliffs: Prentice-Hall, 1989.
8. *Stuart A. M.* Probabilistic and deterministic convergence proofs for software for initial value problems // Numer. Algorithms. 1997. V. 14, N 3. P. 227–260.
9. *Tucker W.* A rigorous ODE solver and Smale's 14th problem // Found. Comput. Math. 2002. V. 2. P. 53–117.
10. *Степанов В. В.* Курс дифференциальных уравнений. М.; Л.: Гостехиздат, 1950.
11. *Ахмеров Р. Р.* Численные методы решения обыкновенных дифференциальных уравнений. Новосибирск: изд. НГУ, 1994.
12. *Годунов С. К.* Лекции по современным аспектам линейной алгебры. Новосибирск: Науч. книга, 2002.
13. *Korhonen T., Tavi P.* Automatic time-step adaptation of the forward Euler method in simulation of models of ion channels and excitable cells and tissue // Simul. Model. Practice and Theory. 2008 V. 16. P. 639–644.

Калинина Елизавета Александровна
Самарина Ольга Николаевна
Санкт-Петербургский госуниверситет
Университетский пр., 35
198504 Петергоф г. Санкт-Петербург
ekalinina69@gmail.com; samarina_o@mail.ru

Статья поступила 30 ноября 2009 г.
Окончательный вариант 12 апреля 2011 г.