Численные методы решения жестких дифференциальных уравнений. Неявные методы Рунге-Кутты.

Куряков Владимир, БПМИ 188
Кошелева Юлия, БПМИ 188
Квиндт Ева, БПМИ 188

2 сентября 2020 г.

Основные определения i

Строгого общепринятого математического определения жёстких ОДУ нет. Принято считать, что жёсткие системы - это те уравнения решения которых получить намного проще с помощью определённых неявных методов, чем с помощью явных.

Жесткие ОДУ можно решать несколькими численными методами: (m,k)-метод, одностадийная комплексная схема Розенброка, метод конечных суперэлементов. Однако мы приведем анализ их решения при помощи неявного метода Рунге-Кутты.

Метод называется А-стабильным, если его область устойчивости S удовлетворяет условию $\mathbb{C}^- \subset S$, где $\mathbb{C}^-$ - левая половина комплексной плоскости.

Область устойчивости любого явного метода является компактом, то есть они не А-стабильны. Также A-стабильность не

Основные определения ii

гарантирует, что метод будет давать стабильные решения, и что точность решений будет зависеть от порядка метода, поэтому вводят новое понятие:

Метод Рунге-Кутты называется S-стабильным, если любая ограниченная $g : [0, T] \to \mathbb{R}$ имеет ограниченную производную и для любой константы $\lambda_0 > 0 \;\; \exists \; h_0 > 0$ т. ч. численное решение (y_n) уравнения

$$y' = g'(t) + \lambda(y - g(t))$$

удовлетворяет

$$-1 < \frac{y_{n+1} - g(t_{n+1})}{y_n - g(t_n)} < 1$$

обеспечивает $y_n \neq g(t_n)$ для $\forall \; 0 < h < h_0$ и $\forall \; \lambda$ т. ч. $Re(-\lambda) \geq \lambda_0$.

S-стабильные методы являются A-стабильными.

Неявные методы Рунге-Кутты i

Наша цель аппроксимировать решение $y : [0, T] \to \mathbb{R}^d$ задачи Коши

$$y' = f(t, y), \quad y(0) = y_0$$

где $f : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ - непрерывно дифференцируемая функция. Идея метода Рунге-Кутты заключается в получении значения в точке $t_{n+1} = t_n + h$ (h - текущий шаг) аппроксимируя интеграл

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

Мы выбираем точки $c_1, \cdots c_s$ и их веса $b_1, \cdots, b_s$, и приближаем интеграл суммой

$$y(t_{n+1}) = y(t_n) + h \sum_{i=1}^{s} b_i f(t_n + c_i h, y(t_n + c_i h)) + Error$$

где

$$y(t_n + c_i h) = y_n + h \sum_{j=1}^{s} a_{ij} f(t_n + c_i h, y(t_n + c_i h))$$

для заданных коэффициентов a_{ij}.

Введём обозначия $k_i = y(t_n + c_i h)$ и $y(t_n) = y_n$.

Если $a_{ij} = 0$ при $i \leq j$, то k_i зависит только от предыдущих k, и они вычисляются последовательно прямой подстановкой. В таком случае метод называет явным. Иначе нужно решить систему неявных уравнений, откуда и название «неявные методы Рунге-Кутты».

Неявные методы Рунге-Кутты iii

Коэффициенты b_i, c_i и a_{ij} определяют конкретный метод и упорядочиваются в таблицу Бутчера:

$$\begin{array}{c|ccc}
c_1 & a_{11} & \cdots & a_{1s} \\
\vdots & \vdots & \ldots & \vdots \\
c_s & a_{s1} & \cdots & a_{ss} \\
\hline
 & b_1 & \cdots & b_s
\end{array}$$

Явные методы в основном не пригодны для решения жёстких дифференциальных уравнений из-за малой области их абсолютной устойчивости.

Решения неявных уравнений i

Сделаем замену $z_i = k_i - y_n$, чтобы работать с величинами поменьше.

$$z_i = h \sum_{l=1}^{s} a_{il} f(t_n + c_l h, y_n + z_l)$$

Что можно переписать в следующем виде:

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = (A \otimes I_d) \begin{pmatrix} hf(t_n + c_1 h, y_n + z_1) \\ \vdots \\ hf(t_n + c_s h, y_n + z_s) \end{pmatrix} = (A \otimes I_d) F(z)$$

где $F(z)$ - соответствующая матрица с предыдущего шага (для краткости), а $\otimes$ - произведение Кронекера.

Решения неявных уравнений ii

Посмотрим на систему как на функцию от неизвестных z_i:

$$G(z) = z - (A \otimes I_d)F(z)$$

$$G'(z) = I_{sd} - (A \otimes I_d)F'(z)$$

Нам нужно найти её 0, для чего подойдёт метод Ньютона

$$z^{(m+1)} = z^{(m)} - \frac{1}{G'(z^{(m)})}G(z^{(m)})$$

$$\Leftrightarrow$$

$$G'(z^{(m)})(z^{(m+1)} - z^{(m)}) = -G(z^{(m)})$$

Решения неявных уравнений iii

Получили систему линейных уравнений

$$\begin{pmatrix} I_d - ha_{11}J_1 & -ha_{12}J_2 & \cdots & -ha_{1s}J_s \\ -ha_{21}J_1 & I_d - ha_{22}J_2 & \cdots & -ha_{2s}J_s \\ \vdots & \vdots & \ddots & \vdots \\ -ha_{s1}J_1 & -ha_{s2}J_2 & \cdots & I_d - ha_{ss}J_s \end{pmatrix} \begin{pmatrix} \Delta z_1^{(m+1)} \\ \Delta z_2^{(m+1)} \\ \vdots \\ \Delta z_s^{(m+1)} \end{pmatrix} =$$

$$= \begin{pmatrix} -z_1^{(m)} + h\sum_{l=1}^{s} a_{1l} f(t_n + c_l h, y_n + z_l^{(m)}) \\ -z_2^{(m)} + h\sum_{l=1}^{s} a_{2l} f(t_n + c_l h, y_n + z_l^{(m)}) \\ \vdots \\ -z_s^{(m)} + h\sum_{l=1}^{s} a_{sl} f(t_n + c_l h, y_n + z_l^{(m)}) \end{pmatrix}$$

где J_i - якобиан в точке $z_i^{(m)}$

Решения неявных уравнений iv

Матрица коэффициентов этого уравнения имеет размерность $sd \times sd$, следовательно решение системы займёт $O(s^3 d^3)$, что очень дорого делать на каждом шаге метода Ньютона. Поэтому все якобианы заменяются на один, в точке y_n, и матрица коэффициентов получается одна на каждой итерации, что позволяет сделать её LU-разложение, так же за $O(s^3 d^3)$, но зато потом решать систему за $O(s^2 d^2)$ подстановками. Такой метод называется упрощённым методом Ньютона.

После решенияу системы мы переходим к $z^{(m+1)} = z^{(m)} + \Delta z^{(m+1)}$.

В качестве $z^{(0)}$ можно взять 0.

Решение найдено, когда длина вектора $\Delta z^{(m+1)}$ (можно брать разные метрики) становится меньше какой-то величины, обычно заданной юзером.

Примеры использования данного метода

Моделирование цепей с электровакуумными приборами

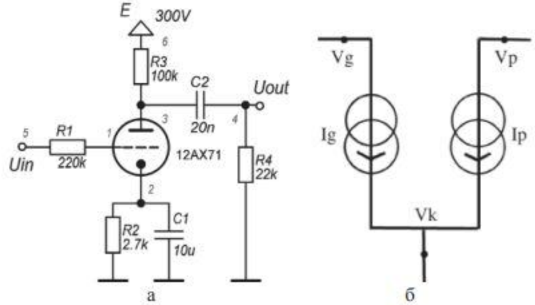


Рис. 1. Типовая схема включения триода и схема его замещения. Цифрами обозначены номера узлов при составлении уравнения

Численное решение уравнения $M_{y'} = f(t, y)$

1. S-стадийный переход

$$(I_s \otimes M)\begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} = h(A \otimes I_n)\begin{pmatrix} f(t_n + c_1 h, x_n + z_1) \\ \vdots \\ f(t_n + c_s h, x_n + z_s) \end{pmatrix}$$

$$x_{n+1} = x_n + \sum_{i=1}^{s} d_i s_i$$

2. Вводим функцию

$$F_0(z) = (I_s \otimes M)z - \begin{pmatrix} f(t_k + c_1 h, x_k + z_k^1) \\ \vdots \\ f(t_k + c_s h, x_k + z_k^s) \end{pmatrix}$$

3. Метод Ньютона

$$W_0(z)\Delta z = (I_s \otimes M)z_k - h(A \otimes I)F(z_k)$$

4. Уравнение нелинейной цепи с триодами

$$f(x) = -Q_x + \begin{bmatrix} -A_t L(H_x) \\ E \end{bmatrix}$$

5. нахождение Якобиана

$$J(x) = -Q - H' \begin{bmatrix} \ldots \\ \delta I_{gk}/\delta x \\ \delta I_{ak}/\delta x \\ \ldots \end{bmatrix}$$

Итерационная матрица

Матрица метода представима в виде: $T^{-1}AT = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_s \end{bmatrix} = \Lambda$

Делаем замену: $\widetilde{z_k} = (T^{-1} \otimes I_n) z_k$, $\widetilde{F} = (T^{-1} \otimes I_s) F$
Тогда итерационная схема преобразуется к виду:

$$\begin{bmatrix} I_n - \lambda_1 J_1 & & \\ & \ddots & \\ & & I_n - \lambda_s J_s \end{bmatrix} \cdot \begin{bmatrix} \widetilde{z_k^1} \\ \vdots \\ \widetilde{z_k^s} \end{bmatrix} = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix} \cdot \begin{bmatrix} \widetilde{z_k^1} \\ \vdots \\ \widetilde{z_k^s} \end{bmatrix} - h \begin{bmatrix} \lambda_1 I_n & & \\ & \ddots & \\ & & \lambda_s I_n \end{bmatrix} \begin{bmatrix} \widetilde{F_1}(z_k) \\ \vdots \\ \widetilde{F_n}(z_k) \end{bmatrix}$$

Вывод

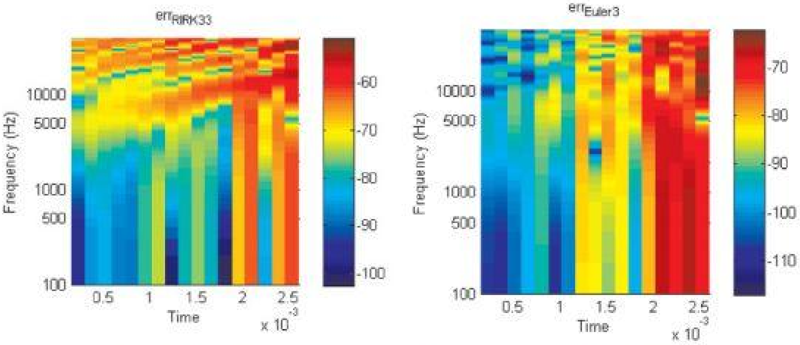


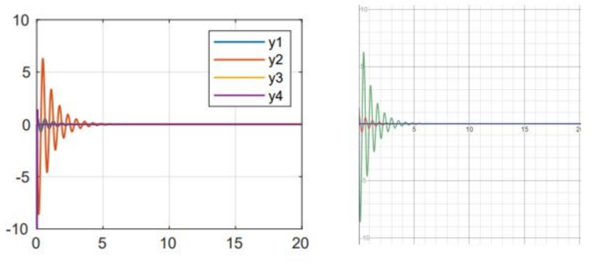
Рис. 4. Сравнение спектра погрешностей метода RIRK33 и ЭПНЭ3 при восьмикратной передискретизации

Реализация

https://github.com/user959/Runge-Kutta-method

Сравнение с реализацией в MatLab

1. $$\begin{cases} y_1' = -y_1 + y_2 \\ y_2' = -100y_1 - y_2 \\ y_3' = -100y_3 + y_4 \\ y_4' = -10000y_3 - 100y_4 \end{cases} \qquad \begin{cases} y_1 = e^{-t}cos(10t) \\ y_2 = -10e^{-t}sin(10t) \\ y_3 = -100e^{-100t}sin(100t) \\ y_4 = e^{-100t}cos(100t) \end{cases}$$



Ode23s

Solution

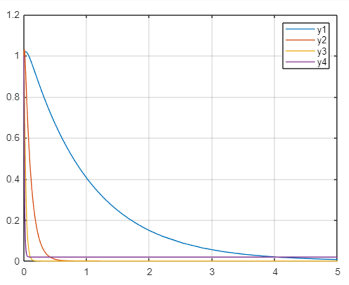
Ошибка вычисления

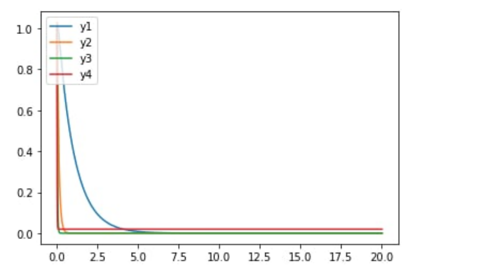
Точка	Разницах в значения y1	Разницах в значения y2	Разницах в значения y3	Разницах в значения y4	Ошибка
4.37(k = exp(-6))	-0.16k	-0.2k	0	-0.2k	8.42exp(-8)
9.44(k = exp(-6))	0	0	0	0.7k	1.69exp(-7)
14.5(k = exp(-6))	0	0	-0.01k	0.31k	7.87exp(-8)
20 (k = exp(-7))	0	0	-0.01k	0.4k	1.16exp(-8)

Точка	Разницах в значения y1	Разницах в значения y2	Разницах в значения y3	Разницах в значения y4	Ошибка
0.7 (k = exp(-12))	-0.03k	-0.13k	0	0	3.56exp(-14)
3.9 (k = exp(-7))	0.02k	-0.1k	0	0	3exp(-9)
7.14 (k = exp(-10))	0	0.38k	0	0	9.51exp(-12)
20 (k = exp(-13))	-0.03k	-0.23k	0	0	5.84exp(-15)

Сравнение реализаций

2. $$\begin{cases} y_1' = -y_1 + y_2^2 + y_3^2 + y_4^2 \\ y_2' = -10y_2 + 10(y_3^2 + y_4^2) \\ y_3' = -40y_3 + 40y_4^2 \\ y_4' = -100y_4 + 2 \end{cases} \qquad \begin{cases} y_1 = ? \\ y_2 = ? \\ y_3 = ? \\ y_4 = \frac{1}{50}e^{-100t}(e^{100t} + 49) \end{cases}$$

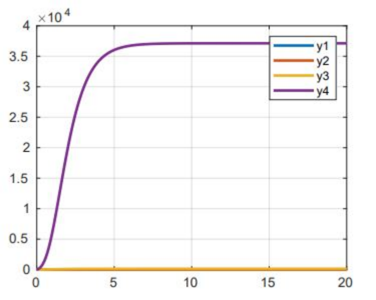


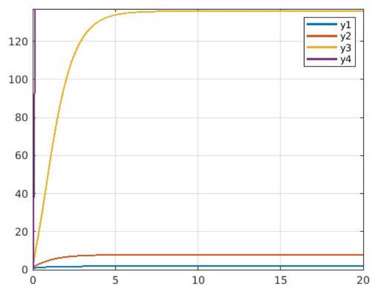


$y_{matlab}(20) = (0.0004; 0.0004; 0.0004; 0.02)$
$y_{self}(20) = (0.0004; 0.0004; 0.0004; 0.02)$

Показательный пример

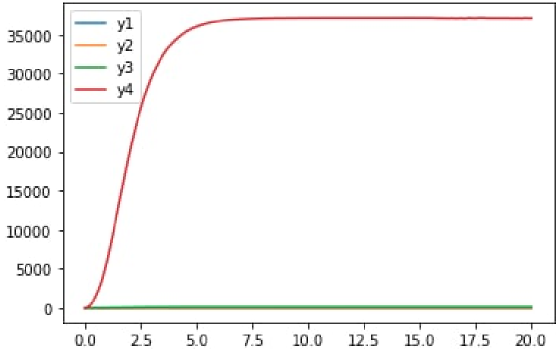
$$3.\ \begin{cases} y_1' = -y_1 + 2 \\ y_2' = -10y_2 + 20y_1^2 \\ y_3' = -40y_3 + 80(y_1^2 + y_2^2) \\ y_4' = -100y_4 + 200(y_1^2 + y_2^2 + y_3^2) \end{cases} \qquad \begin{cases} y_1 = e^{-t}(2e^t - 1) \\ y_2 = ? \\ y_3 = ? \\ y_4 = ? \end{cases}$$





$$y_{ode13s}(20) = (2; 8; 136; 37130) \qquad y_{ode45}(20) = (2; 8; 136; 37128)$$

Реализация



$$y_{self}(20) = (2; 8; 136; 37130)$$

Источники

1. Roger Alexander, Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.'s (SIAM Journal on Numerical Analysis, Vol. 14, No. 6 (Dec., 1977), pp. 1006-1021)
2. E. Hairer, S. P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems (Springer)
3. E. Hairer, G. Wanner, Solving ordinary differential equations II (Springer)
4. Каримов, Андреев, Бутусов, КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ЦЕПЕЙ С ЭЛЕКТРОВАКУУМНЫМИ ПРИБОРАМИ (ТЕХНИЧЕСКИЕ НАУКИ (05.02.00, 05.13.00, 05.17.00, 05.23.00)
5. D. Kressner, Advanced Numerical Analysis (Lecture notes, EPFL)
6. Ю. В. Немировский, А. П. Янковский, Обобщение методов Рунге–Кутты и их применение к интегрированию начально-краевых задач математической физики (Сиб. журн. вычисл. матем., 2005, том 8, номер 1, 57–76)
7. Галанин М.П., Ходжаева С.Р. Методы решения жестких обыкновенных дифференциальных уравнений. Результаты тестовых расчетов
8. Ахмеров Р.Р. Численные методы решения обыкновенных дифференциальных уравнений
9. Josefine Stål, Implementation of singly diagonally implicit Runge-Kutta methods with constant step sizes
10. John C. Butcher, Philippe Chartier, The effective order of singly-implicit Runge-Kutta methods (Numerical Algorithms)