

Баллистический метод с использованием гипер-дуальных чисел

Олифер В. И.

Баллистический метод (иногда также называемый методом стрельбы или методом начальных параметров) не решает краевую задачу для уравнения Эйлера в исходной постановке, а сводит ее к последовательности более простых задач, а именно – задач Коши с особым образом сформулированными начальными условиями.

Рассмотрим двухточечную краевую задачу второго порядка:

$$y''(t) = \gamma \cdot y'(t) + \alpha[\beta \cdot y(t) - \psi(t)] + \varphi'(t), \quad a \leq t \leq b \quad (1)$$

с граничными условиями: $y(a) = y_a$, $y(b) = y_b$.

Здесь: штрих означает производную по t ; t – вещественный параметр (например, время); α , β и γ – некоторые вещественные числа; $\varphi(t)$ и $\psi(t)$ – произвольные аналитические дифференцируемые функции.

Для решения уравнения (1) баллистическим методом [1] исходную краевую задачу надо заменить на соответствующую задачу Коши, которая в нашем случае имеет вид:

$$\begin{cases} y'(t) = \varphi(t) - \alpha \cdot x(t) + \gamma \cdot y(t); \\ x'(t) = \psi(t) - \beta \cdot y(t); \end{cases} \quad \text{при } y(a) = y_a, \quad x(a) = x_a \quad (2)$$

Далее нужно найти такое решение (2) при котором $y(b) = y_b$ (цель). В этом случае, параметром стрельбы служит значение x_a , изменение которого будет влиять на $y(b)$. Каждое решение системы (2) зачастую называют выстрелом. По результатам выстрела производится корректировка параметра стрельбы x_a и осуществляется новый выстрел. Этот процесс продолжается до тех пор, пока не будет поражена цель. Основным элементом этого процесса является процедура, обеспечивающая сам выстрел т.е. решение задачи Коши. Одним из старейших методов интегрирования дифференциальных уравнений является метод разложения в ряд Тейлора. В этом случае расстояние до цели $[a, b]$ разбивается узлами $(t_0, t_1, \dots, t_n; t_0 = a, t_n = b)$ на отрезки (обычно равные Δt). При достаточно малом шаге интегрирования Δt для $(i + 1)$ -го узла имеем

$$\begin{aligned} y_{i+1} &= y(t_i + \Delta t) \approx y_i + y'_i \Delta t + \frac{1}{2} y''_i \Delta t^2 + \frac{1}{6} y'''_i \Delta t^3 + \dots, \\ x_{i+1} &= x(t_i + \Delta t) \approx x_i + x'_i \Delta t + \frac{1}{2} x''_i \Delta t^2 + \frac{1}{6} x'''_i \Delta t^3 + \dots \end{aligned} \quad (3)$$

Значения производных с использованием (2) определяются следующими соотношениями:

$$\begin{aligned} y_i^{(m)} &= \varphi_i^{(m-1)} - \alpha \cdot x_i^{(m-1)} + \gamma \cdot y_i^{(m-1)}, \\ x_i^{(m)} &= \psi_i^{(m-1)} - \beta \cdot y_i^{(m-1)}, \end{aligned} \quad (4)$$

где показатель в круглых скобках указывает на порядок производной, а $m = 1, 2, 3, \dots, M$ – максимальный порядок производных, удерживаемых в разложении Тейлора.

Процесс решения осуществляется последовательным перебором узлов t_0, t_1, \dots, t_n . Для узла t_0 входными данными служат $y_0 = y_a$ и $x_0 = x_a$. По формулам (4) и (3) определяются y_1 и x_1 , которые передаются узлу t_1 в качестве входных данных, и т. д. по цепочке. Результат работы предпоследнего узла содержит значения $y(b)$ и $x(b)$, которые используются для оценки точности выстрела и корректировки значения параметра стрельбы x_a .

Линейность уравнения (1) обеспечивает линейную зависимость между x_a и $y(b)$. В этом случае необходимо сделать только два пристрелочных выстрела для определения параметра стрельбы, обеспечивающего поражение цели: $x_a = (y_b - y_1(b))/(y_2(b) - y_1(b))$, где $y_1(b)$ и $y_2(b)$ результаты первого и второго выстрелов с разными значениями x_a соответственно.

Главным недостатком метода разложения в ряд Тейлора является то, что для получения производных (4) необходимо знать производные функций $\varphi(t)$ и $\psi(t)$. Очевидно, что если $\varphi(t)$ и $\psi(t)$ – сложные функции, то получение их производных может представлять собой весьма сложную и утомительную задачу. Однако, последняя может быть сравнительно легко решена с использованием автоматического дифференцирования (АД) на основе гипер-дуальных чисел 3-го класса [2]. Откликом функций от гипер-дуальных чисел 3-го класса являются точные (с точностью вычислительного устройства) значения самой функции и ее 1-й, 2-й и 3-й производных в заданной точке. В этом случае в разложениях (3) будут удерживаться только первые четыре члена т.е. значение функции и величины 1-й, 2-й и 3-й производных.

Компьютерная реализация

Компьютерная реализация компактных дуальных чисел класса 3 Tdn3 (truncated-dual number 3) на языке SWIFT операционной системы macOS дана в [2]. В выше цитированной работе дан код процедуры решения задачи Коши Taylor с использованием созданного типа данных Tdn3. Автор настоящей статьи посчитал нецелесообразным дублировать указанный код в тексте этой работе. Однако он был использован в численном эксперименте. Для последнего была создана процедура, осуществляющая вышеописанный баллистический метод, Ballistics3 на языке SWIFT операционной системы macOS (см. Приложение 1).

Численный эксперимент

Для проведения численного анализа предлагаемого подхода была рассмотрена краевая задача (1) с $\alpha = \beta = 1$, $\gamma = 0$, $\varphi(t) = 2e^{3t}$, $\psi(t) = e^{3t}$, $y(0) = 3$, $y(1) = 3$ и $0 \leq t \leq 1$.

Соответствующая задача Коши имеет вид:

$$\begin{cases} y'(t) = 2e^{3t} - x(t); \\ x'(t) = e^{3t} - y(t); \end{cases} \quad \text{при } y(0) = 3, \quad x(0) = x_a \quad (5)$$

При одиннадцати расчетных точках первый ($x_a = 0$) и второй ($x_a = 1$) выстрелы дают $y_1(1) \approx 16.3526995$ и $y_2(1) \approx 15.1775423$ соответственно. Тогда параметр стрельбы, удовлетворяющий условию $y(1) = 3$, равен $x_a \approx 11.362479$. Точное аналитическое решение имеет вид $y(t) = A \cdot e^{-t} + B \cdot e^t + 0.625 \cdot e^{3t}$ с $A = 6.8113357977216706 \approx 6.811$ и $B = -4.4363357977216706 \approx -4.436$.

В табл. 1 приведены результаты расчета для 11-ти расчетных узлов, полученные по процедуре Ballistics3.

t/y	1-й выстрел $x_a = 0$	2-й выстрел $x_a = 1$	3-й выстрел $x_a \cong 11.362479$	Точное решение
0.0	3.0	3.0	3.0	3.0
0.1	3.24283333	3.14266666	2.10469166	2.10397122
0.2	3.58610614	3.38477114	1.29844137	1.29705677
0.3	4.05703869	3.75252117	0.59696466	0.59499118
0.4	4.69245028	4.28170338	0.02534713	0.02288855
0.5	5.54203900	5.02095270	-0.37879323	-0.38159156
0.6	6.67280900	6.03616901	-0.56099967	-0.56393328
0.7	8.17504578	7.41648134	-0.44412691	-0.44690831
0.8	10.1703805	9.28230061	0.07959154	0.07736525
0.9	12.8226729	11.7961906	1.15928787	1.15817888
1.0	16.3526995	15.1775423	2.99999999	3.0

Таблица 1. Результаты расчета

Графическое представление траекторий выстрелов приведены на рис. 1. Нетрудно видеть, что третий выстрел практически совпадает с точным решением.

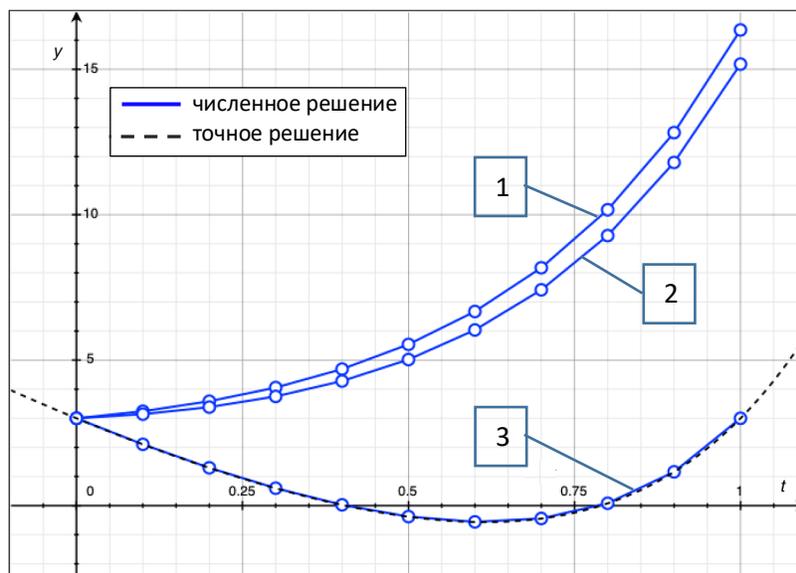


Рис.1. Результаты расчета: 1-траектория 1-го выстрела; 2- траектория 2-го выстрела; 3- траектория точного выстрела.

Количество пристрелочных выстрелов можно сократить если воспользоваться следующим подходом.

Краевая задача, соответствующая системе (5), имеет вид

$$y''(t) = y(t) + 5e^{3t}, \quad 0 \leq t \leq 1 \quad (6)$$

с граничными условиями $y(0) = y(1) = 3$.

Пусть $y(t) = y_1(t) + k \cdot y_2(t)$, тогда из правого граничного условия получим $y(1) = y_1(1) + k \cdot y_2(1)$ и $k = (y(1) - y_1(1))/y_2(1) = (3 - y_1(1))/y_2(1)$. Из левого граничного условия имеем следующее соотношение $y(0) = y_1(0) + k \cdot y_2(0) = 3$.

Для определения $y_1(t)$ и $y_2(t)$ рассмотрим две задачи Коши

$$\begin{cases} y_1'(t) = 2e^{3t} - x_1(t); \\ x_1'(t) = e^{3t} - y_1(t); \end{cases} \quad \text{при } y_1(0) = 3, \quad x_1(0) = \varphi(0) = 2 \quad (7)$$

и

$$\begin{cases} y_2'(t) = -x_2(t); \\ x_2'(t) = -y_2(t); \end{cases} \quad \text{при } y_2(0) = 0, \quad x_2(0) = -1 \quad (8)$$

Начальные условия для (7) и (8) были определены из соотношений $y_1(0) = 3$, $y_1'(0) = 0$ и $y_2(0) = 0$, $y_2'(0) = 1$. Тогда решив (7) и (8) получим решение краевой задачи (6):

$$y(t) = y_1(t) + y_2(t) \cdot (3 - y_1(1))/y_2(1) \quad (9)$$

Таким образом, для получения траектории поражающего выстрела достаточно выполнить всего лишь два пристрелочного выстрела (7) и (8), а затем объединить их результаты используя (9). Процедура `Ballistics2`, реализующая этот подход дана в Приложении 1.

Результаты, полученные по процедурам `Ballistics3` и `Ballistics2`, полностью совпадают.

Точность баллистического метода напрямую зависит от точности решения задачи Коши. Используемый же здесь метод численного интегрирования задачи Коши, основанный на автоматическом дифференцировании и гипер-дуальных числах, обеспечивает решение с точностью вычислительного устройства.

Приложение 1

Тип данных `Tdn3` и процедура `Taylor()` приведены в [2].

```
//----- Ballistics solution for y''(t) = alpha[beta*y(t) - psi(t)] + phi'(t)
//          phi, psi - predefined functions
//          a, b - real numbers
//          t0, tn - start and end of integration interval
//          n - number of parts into which the interval [t0, tn] is divided
//          y0, yn - initial values
//output: array [(t, y, x)]

func Ballistics3(phi:(Double) -> Tdn3, psi:(Double) -> Tdn3,
  a:Double, b:Double, t0:Double, tn:Double, n:Int,
  y0:Double, yn:Double)->[(t:Double, y:Double, x:Double)]{
  let R0 = Taylor(phi: phi, psi: psi, a: a, b: a, t0: t0, tn: tn, n: n, y0: y0, x0: 0);
  let R1 = Taylor(phi: phi, psi: psi, a: a, b: a, t0: t0, tn: tn, n: n, y0: y0, x0: 1);
  let x_0 = (yn - R0[n].y)/(R1[n].y - R0[n].y);
  return Taylor(phi: phi, psi: psi, a: a, b: b, t0: t0, tn: tn, n: n, y0: y0, x0: x_0);
}
```

```
func Ballistics2( $\varphi$ :(Double) -> Tdn3,  $\psi$ :(Double) -> Tdn3,
                a:Double, b:Double, t0:Double, tn:Double, n:Int,
                y0:Double, yn:Double)->[Double]{
  func  $\varphi$ 0(t:Double)->Tdn3{return Tdn3(re: 0)}
  func  $\psi$ 0(t:Double)->Tdn3{return Tdn3(re: 0)}
  let R0 = Taylor( $\varphi$ :  $\varphi$ ,  $\psi$ :  $\psi$ , a: a, b: a, t0: t0, tn: tn, n: n, y0: y0, x0:  $\varphi$ (0).re);
  let R1 = Taylor( $\varphi$ :  $\varphi$ 0,  $\psi$ :  $\psi$ 0, a: a, b: a, t0: t0, tn: tn, n: n, y0: 0, x0: -1);
  let k = (yn - R0[n].y)/R1[n].y;
  var R:[Double] = [];
  for i in 0...n {R.append(R0[i].y + k*R1[i].y)}
  return R;
}

//predefined functions:
func  $\varphi$ (t:Double)->Tdn3{return 2.0*Tdn3.exp(X: 3.0*Tdn3(re: t, im1: 1, im2: 0, im3: 0))}
func  $\psi$ (t:Double)->Tdn3{return Tdn3.exp(X: 3.0*Tdn3(re: t, im1: 1, im2: 0, im3: 0))}

//get solution using 3 shots:
let RX = Ballistics3( $\varphi$ :  $\varphi$ ,  $\psi$ :  $\psi$ , a: 1, b: 1, t0: 0, tn: 1, n: 10, y0: 3, yn: 3);

//get solution using 2 shots:
let R = Ballistics2( $\varphi$ :  $\varphi$ ,  $\psi$ :  $\psi$ , a: 1, b: 1, t0: 0, tn: 1, n: 10, y0: 3, yn: 3);
```

Литература

1. *Калиткин Н. Н.* Численные методы. - СПб.: БХВ-Петербург, 2011. 592 с.
2. *Олифер В. И.* К численному решению задачи Коши с использованием гипер-дуальных чисел. –URL:
http://viosolutions.amerihomesrealty.com/pdf/К_численному_решению_задачи_Коши_с_использованием_гипер-дуальных_чисел.pdf
(дата обращения: 01.04.2020)

Аннотация

В данной публикации излагается применение расширенных гипер-дуальных чисел в численной реализации баллистического метода. Для решения промежуточной задачи Коши используется метод разложения в ряды Тейлора и компьютерное автоматическое дифференцирование на основе дуальных чисел третьего класса. Описаны основные операции и базовые функции указанного класса дуальных чисел. Представлена процедура последовательного поточечного интегрирования исходной двухточечной краевой задачи второго порядка с использованием автоматического дифференцирования и дуальных чисел третьего класса. Дана компьютерная реализация для языка SWIFT операционной системы macOS. Проведены численные эксперименты на базе полученного программного обеспечения.

Ключевые слова

Баллистический метод, задача Коши, дуальные числа, гипер-дуальные числа, расширение гипер-дуальных чисел, автоматическое дифференцирование, ballistic method, Cauchy problem, dual numbers, hyper-dual numbers, extension of hyper-dual numbers, automatic differentiation

24 апреля 2020 г.