

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ГИПЕР-ДУАЛЬНЫХ ЧИСЕЛ

Олифер В. И.

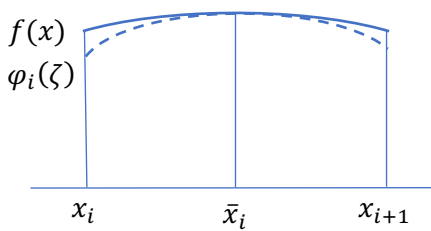
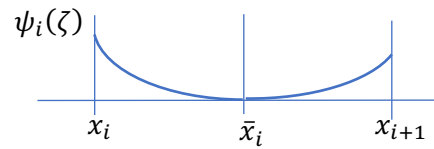
1. Процедура численного интегрирования

Численное интегрирование является приближённым вычислением значения определённого интеграла. Обычно численное интегрирование применяется когда подынтегральная функция не задана аналитически либо её первообразная не выражается через аналитические функции. Далее рассматривается последний случай, т. е. подынтегральная функция задана аналитически, но является настолько сложной, что получение её первообразной в аналитической форме весьма сложно или вообще невозможно. Большинство методов численного интегрирования состоит в следующем. Заданный отрезок интегрирования $[a, b]$ делится опорными точками $x_0, x_1, x_2, \dots, x_n$ на n равных отрезков длиной $\Delta x = 2h = (b - a)/n$. На каждом из этих отрезков $[x_i, x_{i+1}]$ исходная подынтегральная функция $f(x)$ обычно аппроксимируется многочленами той либо иной степени. Например, многочленом третьей степени. В этом случае необходимо привлечь еще одну точку, лежащую между точками x_i и x_{i+1} . Пусть эта точка лежит посередине рассматриваемого i -го отрезка, т. е. $\bar{x}_i = (x_{i+1} - x_i)/2$. Далее разложим исходную подынтегральную функцию $f(x)$ в ряд Тейлора в окрестности этой точки (удерживая только четыре первых члена)

$$f(\bar{x}_i + \zeta) \approx \varphi_i(\zeta) = f(\bar{x}_i) + f'(\bar{x}_i)\zeta + \frac{1}{2}f''(\bar{x}_i)\zeta^2 + \frac{1}{6}f'''(\bar{x}_i)\zeta^3, \quad \zeta \in [-h, h] \quad (1.1)$$

где $f'(\bar{x}_i)$, $f''(\bar{x}_i)$ и $f'''(\bar{x}_i)$ значения первой, второй и третьей производных исходной функции $f(x)$ соответственно.

Нетрудно видеть, что функция $\varphi_i(\zeta)$ гиперболическая и при $\zeta = 0$ равна $f(\bar{x}_i)$. Однако, в общем случае, на концах i -го интервала эта функция не дает значений $f(x_i)$ и $f(x_{i+1})$ (см. рис. 1).

Рис. 1. Графики функций $f(x)$ и $\varphi_i(\zeta)$ Рис. 2. График функции $\psi_i(\zeta)$

Для устранения этого недостатка введем корректирующую функцию

$$\psi_i(\zeta) = (\alpha_{1i}\theta(-\zeta) + \alpha_{2i}\theta(\zeta))\zeta + (\beta_{1i}\theta(-\zeta) + \beta_{2i}\theta(\zeta))\zeta^2 + (\gamma_{1i}\theta(-\zeta) + \gamma_{2i}\theta(\zeta))\zeta^3 \quad (1.2)$$

(см. рис. 2), где $\theta(\zeta)$ - функция Хэвисайда [1]. Значения коэффициентов α_{1i} , β_{1i} , γ_{1i} и α_{2i} , β_{2i} , γ_{2i} определим из условий в концевых точках i -го интервала ($\bar{\varphi}_i(\zeta) = \varphi_i(\zeta) + \psi_i(\zeta)$):

$$\begin{aligned} f(x_i) &= \bar{\varphi}_i(-h), & f'(x_i) &= \bar{\varphi}'_i(-h), & f''(x_i) &= \bar{\varphi}''_i(-h), \\ f(x_{i+1}) &= \bar{\varphi}_i(h), & f'(x_{i+1}) &= \bar{\varphi}'_i(h), & f''(x_{i+1}) &= \bar{\varphi}''_i(h), \end{aligned} \quad (1.3)$$

которые приводят к двум независимым системам алгебраических уравнений третьего порядка относительно α_{1i} , β_{1i} , γ_{1i} и α_{2i} , β_{2i} , γ_{2i} соответственно. Их решение равно ($k = 1, 2$):

$$\begin{aligned} \alpha_{ki} &= (-1)^k [3h^{-1}\Delta f_{ki} - (-1)^k(2f'_{ki} + \bar{f}'_i) + hf''_{ki}/2], \\ \beta_{ki} &= (-1)^k [-(-1)^k 3h^{-2}\Delta f_{ki} + 3h^{-1}f'_{ki} - (-1)^k f''_{ki} - (-1)^k \bar{f}''_i/2], \\ \gamma_{ki} &= (-1)^k [h^{-3}\Delta f_{ki} - (-1)^k h^{-2}f'_{ki} + h^{-1}f''_{ki}/2 - (-1)^k \bar{f}'''_i/6], \end{aligned} \quad (1.4)$$

где:

$$\begin{aligned} \Delta f_{ki} &= f_{ki} - f(\bar{x}_i), & f_{ki} &= f(x_{ki}), & \bar{f}'_i &= f'(\bar{x}_i), & \bar{f}''_i &= f''(\bar{x}_i), & \bar{f}'''_i &= f'''(\bar{x}_i), \\ f'_{ki} &= f'(x_{ki}), & f''_{ki} &= f''(x_{ki}), & x_{1i} &= x_i, & x_{2i} &= x_{i+1} \end{aligned} \quad (1.5)$$

Тогда (1.1) с учетом (1.2) и (1.4) запишется следующим образом

$$\begin{aligned} f(\bar{x}_i + \zeta) &\approx \varphi_i(\zeta) + \psi_i(\zeta) = f(\bar{x}_i) + \\ &+ (f'(\bar{x}_i) + \alpha_{1i}\theta(-\zeta) + \alpha_{2i}\theta(\zeta))\zeta + \\ &+ (f''(\bar{x}_i)/2 + \beta_{1i}\theta(-\zeta) + \beta_{2i}\theta(\zeta))\zeta^2 + \\ &+ (f'''(\bar{x}_i)/6 + \gamma_{1i}\theta(-\zeta) + \gamma_{2i}\theta(\zeta))\zeta^3 \end{aligned} \quad (1.6)$$

Нетрудно проверить, что полученное соотношение определяет две гиперболы сопряженные в точке \bar{x}_i и удовлетворяет условиям в конечных точках i -го интервала (1.3).

Интегрируя (1.6) от $-h$ до h получим

$$S_i = \int_{-h}^h (\varphi_i(\zeta) + \psi_i(\zeta)) d\zeta = \Delta x \cdot f_{0i} + r1 \cdot \Delta f_i + r2 \cdot (f'_{1i} - f'_{2i}) + r3 \cdot (f''_{1i} + f''_{2i}) \quad (1.7)$$

где:

$$\begin{aligned} f_{0i} &= f(\bar{x}_i), & \Delta f_i &= f(x_{1i}) + f(x_{2i}) - 2 \cdot f(\bar{x}_i), \\ f'_{1i} &= f'(x_{1i}), & f'_{2i} &= f'(x_{2i}), & f''_{1i} &= f''(x_{1i}), & f''_{2i} &= f''(x_{2i}), \\ r1 &= \frac{3}{4}h, & r2 &= \frac{1}{4}h^2, & r3 &= \frac{1}{24}h^3 \end{aligned}$$

Как видно, величина S_i зависит от значений исходной функции в точках x_i , \bar{x}_i и x_{i+1} , а также от значений первой и второй производных исходной функции в точках x_i и x_{i+1} .

Суммируя все S_i получим значение интеграла от исходной функции в пределах $[a, b]$:

$$J = \int_a^b f(x) dx \approx \sum_{i=1}^n S_i = \sum_{i=1}^n [\Delta x \cdot f_{0i} + r1 \cdot \Delta f_i + r2 \cdot (f'_{1i} - f'_{2i}) + r3 \cdot (f''_{1i} + f''_{2i})] \quad (1.8)$$

Количество расчетных точек x_i и \bar{x}_i для реализации (1.8) равно $2n + 1$.

При достаточно малом Δx в ряде случаев можно вместо трех расчетных точек x_i , \bar{x}_i и x_{i+1} использовать только две x_i , и x_{i+1} . Тогда, полагая

$$\begin{aligned} f_{0i} &= f(x_i), & \Delta f_i &= f(x_{i+1}) - f(x_i), \\ f'_{1i} &= f'(x_{i+1}), & f''_{1i} &= f''(x_{i+1}), \\ r1 &= \frac{3}{4}\Delta x, & r2 &= \frac{1}{4}\Delta x^2, & r3 &= \frac{1}{24}\Delta x^3 \end{aligned}$$

получим

$$I = \int_a^b f(x) dx \approx \sum_{i=1}^n [\Delta x \cdot f_{0i} + r1 \cdot \Delta f_i - r2 \cdot f'_{1i} + r3 \cdot f''_{1i}] \quad (1.9)$$

В этом случае требуется $n + 1$ расчетных точек x_i .

Как указывалось выше, соотношение (1.8), а также (1.9) требуют вычисления значений исходной функции и ее первой и второй производных в расчетных точках. Далее для их вычисления будем использовать метод автоматического дифференцирования, основанный на использовании усеченных гипер-дуальных чисел.

2. Автоматическое дифференцирование и усеченные гипер-дуальные числа

Автоматическое дифференцирование [2, 3] связано с точным (точностью представления чисел в компьютерной системе) вычислением производных сложных математических функций. Автоматическое дифференцирование (АД) позволяет избежать дублирования функциональности программного кода (изменение кода функции не требует изменения кода ее производной). Для компьютерной реализации АД необходимо создать новый тип данных, перезагрузить базовые математические функции и операции над ними. Если новый тип данных строится на основе усеченных гипер-дуальных чисел [4], то за одно обращение к перезагруженной функции точно вычисляются значения самой функции и ее первой и второй производных.

Согласно [4] усеченное гипер-дуальное число (truncated hyper-dual number) определяется выражением $X = x + x_1 \varepsilon + x_2 \omega$, где x , x_1 и x_2 – вещественные числа, ε и ω – мнимые символы. Пространство усеченных гипер-дуальных чисел отвечает трехмерной алгебре с правилом умножения элементов базиса $\{1, \varepsilon, \omega\}$:

\times	1	ε	ω
1	1	ε	ω
ε	ε	2ω	0
ω	ω	0	0

(2.1)

Таблица 2.1. Правила умножения элементов базиса усечённых гипер-дуальных чисел

Число $x = Re(X) = X.Re$ называется главной частью X , а $x_1 = Im_1(X) = X.Im1$ и $x_2 = Im_2(X) = X.Im2$ – мнимыми частями X .

Алгебраические операции сложения, умножения, обращения и деления (с учетом табл. 2.1) определены по правилам:

$$\begin{aligned}
 A &= a + a_1 \boldsymbol{\varepsilon} + a_2 \boldsymbol{\omega}, & B &= b + b_1 \boldsymbol{\varepsilon} + b_2 \boldsymbol{\omega} \\
 A + B &= a + b + (a_1 + b_1) \boldsymbol{\varepsilon} + (a_2 + b_2) \boldsymbol{\omega}, \\
 A \cdot B &= a \cdot b + (a \cdot b_1 + b \cdot a_1) \boldsymbol{\varepsilon} + (a \cdot b_2 + 2 \cdot a_1 \cdot b_1 + b \cdot a_2) \boldsymbol{\omega}, \\
 A^{-1} &= a^{-1} - a_1 \cdot a^{-2} \boldsymbol{\varepsilon} + (2 \cdot a_1^2 \cdot a^{-3} - a_2 \cdot a^{-2}) \boldsymbol{\omega}, \\
 A/B &= A \cdot B^{-1} = a \cdot b^{-1} + (a_1 \cdot b^{-1} - a \cdot b_1 \cdot b^{-2}) \boldsymbol{\varepsilon} + [2 \cdot (a \cdot b_1^2 \cdot b^{-3} - a_1 \cdot b_1 \cdot b^{-2}) - \\
 &\quad a \cdot b_2 \cdot b^{-2} + a_2 \cdot b^{-1}] \boldsymbol{\omega}
 \end{aligned} \tag{2.2}$$

Функция усеченного гипер-дуального аргумента реализуется выражением

$$F(X) = f(x) + x_1 \cdot f'(x) \boldsymbol{\varepsilon} + (x_2 \cdot f'(x) + x_1^2 \cdot f''(x)) \boldsymbol{\omega} \tag{2.3}$$

При $x_1 = 1$ и $x_2 = 0$ выражение (2.3) принимает вид:

$$F(X) = f(x) + f'(x) \boldsymbol{\varepsilon} + f''(x) \boldsymbol{\omega} \tag{2.4}$$

Описания элементарных (базовых) функций усечённого гипер-дуального аргумента приведены в [4]. Например, $\ln(X) = \ln(x) + x^{-1} \boldsymbol{\varepsilon} - x^{-2} \boldsymbol{\omega}$, откуда $Re(X) = f(x) = \ln(x)$, $Im_1(X) = f'(x) = x^{-1}$ и $Im_2(X) = f''(x) = -x^{-2}$.

Вычисление сложной усеченной гипер-дуальной функции (function composition) вида $F = f_1(f_2(\dots f_n(X) \dots), X), X)$ (где значение f_n используется в качестве аргумента для f_{n-1}) необходимо начать с вычисления $F_n = f_n(X)$, продолжить вычислениями $F_{n-1} = f_{n-1}(F_n, X)$, $F_{n-2} = f_{n-2}(F_{n-1}, X)$, ..., $F = f_1(F_2, X)$.

Для применения АД (на базе усеченных гипер-дуальных чисел и усеченных гипер-дуальных функций) к соотношениям (1.8) и (1.9) необходимо выполнить следующие отображения

$$x \rightarrow X = x + 1 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega}, \quad f(x) \rightarrow F(X) = f(x) + f'(x) \cdot \boldsymbol{\varepsilon} + f''(x) \cdot \boldsymbol{\omega}$$

Тогда для (1.8)

$$\begin{aligned}
 \bar{X}_i &= \bar{x}_i + 0 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega}, & X_{1i} &= x_{1i} + 1 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega}, & X_{2i} &= x_{2i} + 1 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega}, \\
 f_{0i} &= F(\bar{X}_i) \cdot Re, & \Delta f_i &= F(X_{1i}) \cdot Re + F(X_{2i}) \cdot Re - 2 \cdot F(\bar{X}_i) \cdot Re, \\
 f'_{1i} &= F(X_{1i}) \cdot Im1, & f'_{2i} &= F(X_{2i}) \cdot Im1, & f''_{1i} &= F(X_{1i}) \cdot Im2, & f''_{2i} &= F(X_{2i}) \cdot Im2
 \end{aligned}$$

и для (1.9) имеем

$$\begin{aligned}
 X_i &= x_i + 0 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega}, & X_{i+1} &= x_{i+1} + 1 \cdot \boldsymbol{\varepsilon} + 0 \cdot \boldsymbol{\omega} \\
 f_{0i} &= f(x_i), & \Delta f_i &= f(x_{i+1}) \cdot Re - f(x_i) \cdot Re, \\
 f'_{1i} &= F(X_{i+1}) \cdot Im1, & f''_{1i} &= F(X_{i+1}) \cdot Im2
 \end{aligned}$$

Примеры задания исходных функций см. Приложение 1.

3. Программная реализация

Используя программный код на языке SWIFT, описывающий тип данных Thdn (truncated hyper-dual number) [4], была выполнена компьютерная реализация описанного выше подхода на языке программирования SWIFT 5 для macOS 11.2.1. Для методов (1.8) и (1.9) были

написаны компьютерные процедуры `integral_102()` и `integral_01()` соответственно (см. Приложение 1), которые использовались в численных экспериментах.

4. Численные эксперименты

Для проведения численного анализа предлагаемого подхода были рассмотрены самые разнообразные исходные функции с разными пределами интегрирования и количеством расчетных точек. Некоторые результаты вычислений приведены в таблице 4.1.

#	n	$f(x)$	$[a, b]$	J	I	Точное решение
1	1	$\sin(x)$	$[0, \frac{\pi}{2}]$	1.000754996...	1.016606220...	1.0
	2			1.000048988...	1.000903933...	
2	5	$\frac{\ln(x+1)}{x^2+1}$	$[0, 1]$	0.272199339...	0.272214831...	0.272198445...
	10			0.272198329...	0.272199324...	
3	50	$\frac{e^{-x^2}}{x^2+1}$	$[0, 50]$	0.673042143...	0.636904646...	0.671646710...
	100			0.671649354...	0.670775543...	
4	100	$e^{-x^2} \cdot \cos(5x)$	$[0, 50]$	0.001710868...	-0.012174837...	0.001710820...
	500			0.001710820...	0.001707534...	
5	10	$\frac{1}{\sqrt{x}}$	$[1, 9]$	4.000111232...	4.000888939...	4.0
	20			4.000006489...	4.000075801...	
6	10	$\frac{\sin(x)}{x}$	$[0, \pi]$	1.8519910403...	1.851939204...	1.851937052...
	20			1.851943790...	1.851937198...	
7	40	$\cos(10 \cdot \cos(x))$	$[0, 1]$	-0.301927778...	-0.301927472...	-0.301927797...
	80			-0.301927796...	-0.301927777...	
8	20	$\frac{x}{tg(x)}$	$[0, \frac{\pi}{2}]$	1.088799758...	1.088833161...	1.088793045...
	80			1.088793150...	1.088793674...	
9	40	$\frac{x \cdot \sin(x)}{1 + \cos^2(x)}$	$[0, \pi]$	2.467401084...	2.467400842...	2.467401100...
	80			2.467401099...	2.467401084...	
10	5	$\frac{1}{x^2+1}$	$[0, 1]$	0.785398165...	0.785395063...	0.785398163...
	10			0.785398163...	0.78539806...	

Таблица 4.1. Результаты численного интегрирования

Из анализа полученных результатов следует, что формула (1.8) дает аналогичный результат что и формула (1.9) при удвоенном количестве расчетных точек в последней (см. желтые клетки таблицы 1). В тоже время по формуле (1.8) получается результат на порядок лучше, чем по методу Симпсона [5] при одинаковом количестве расчетных точек.

5. Заключение

Полученные формулы численного интегрирования (1.8) и (1.9), основанные на использовании типа данных Thdn и автоматического дифференцирования дают заметно лучшие результаты чем известные методы численного интегрирования 4-го порядка погрешности.

Как видно из компьютерных процедур `integral_102()` и `integral_01()` (см. Приложение 1), соотношения (1.8) и (1.9) относительно просто программируются.

Описанные выше процедуры численного интегрирования обладают универсальностью, позволяющей рассматривать функцию $F(X)$ как некий «черной ящик», поставляющий значение функции $f(x)$ и точные величины ее первой и второй производных по данному значению $X = x + 1 \cdot \varepsilon + 0 \cdot \omega$.

Для интегрирования по неограниченным интервалам можно использовать метод замены переменных на конечный интервал, например,

$$\int_{-\infty}^{\infty} f(x) dx = \int_{-1}^1 f\left(\frac{t}{1-t^2}\right) \frac{1+t^2}{(1-t^2)^2} dt, \quad \int_a^{\infty} f(x) dx = \int_0^1 f\left(a + \frac{t}{1-t}\right) \frac{dt}{(1-t)^2},$$

$$\int_{-\infty}^b f(x) dx = \int_0^1 f\left(b - \frac{1-t}{t}\right) \frac{dt}{t^2}$$

и тем самым вычислять значения интегральных функций.

Приложение 1.

Код для численного эксперимента на языке Swift 5 (macOS 11.2). Тип данных Thdn (truncated-dual number) приведен в [4].

// Для реализации приведенного ниже кода необходимо добавить файл, описывающий тип данных Thdn (см. [4])

```
let pi = Double.pi;
// INPUT DATA:
// xo – start point, end point, xn – end point, n – number of intervals, F – function pointer
// OUTPUT DATA:
// J – integral value
func integral_102(xo: Double, xn: Double, n: Int, F:(Thdn -> Thdn) -> Double{
    let Δ = (xn - xo)/Double(n), h = Δ/2.0, h2 = h*h, h3 = h*h2,
        r1 = h*3.0/4.0, r2 = h2/4.0, r3 = h3/24.0;
    var J = 0.0, S = 0.0, Δf = 0.0, f0 = 0.0, F2:Thdn,
```

```

    F1:Thdn = F(Thdn(re:x xo, im1:1, im2:0));
    for x in stride(from: xo + Δ, through: xn, by: Δ){
        f0 = F(Thdn(re:x -h, im1:0, im2:0)).re;
        F2 = F(Thdn(re:x, im1:1, im2:0));
        Δf = F1.re + F2.re - 2.0*f0;
        S = f0*Δ + Δf*r1 + (F1.im1 - F2.im1)*r2 + (F1.im2 + F2.im2)*r3;
        J += S; F1 = F2;
    }
    return J;
}

// INPUT DATA:
// xo – start point, end point, xn – end point, n – number of intervals, F– function pointer
// OUTPUT DATA:
// J – integral value

func integral_01(xo: Double, xn: Double, n: Int, F:(Thdn) -> Thdn) -> Double{
    let Δ = (xn - xo)/Double(n), Δ2 = Δ*Δ, Δ3 = Δ*Δ2,
        r1 = Δ*3.0/4.0, r2 = Δ2/4.0, r3 = Δ3/24.0;
    var J = 0.0, S = 0.0, Δf = 0.0,
        f0 = F(Thdn(re:xo, im1:0, im2:0)).re, F1:Thdn;
    for x in stride(from: xo + Δ, through: xn, by: Δ){
        F1 = F(Thdn(re: x, im1:1, im2:0));
        Δf = F1.re - f0;
        S = f0*Δ + Δf*r1 - F1.im1*r2 + F1.im2*r3;
        J += S; f0 = F1.re;
    }
    return J;
}

//-----
// example of the function under study: 1/(1 + x2)
func X1_2(X: Thdn) -> Thdn {return Thdn(re: 1.0)/(Thdn(re: 1.0) + X*X)}

// example of the function under study: sin(x)/x
func SINX(X: Thdn) -> Thdn {
    if X.re == 0.0 {return Thdn(re: 1.0, im1: 0.0, im2: -1/3)}
    return Thdn.sin(X: X)/X}

// function calls integral_102() and integral_01()
let J1 = integral_102(xo:0.0, xn:π, n:20, F:SINX);
let J2 = integral_01(xo:0.0, xn:π, n:20, F:SINX);

```

ЛИТЕРАТУРА

1. Кеч В., Теодореску П. Введение в теорию обобщённых функций с приложениями в технике. - М.: Мир, 1978, - 520 с.
2. Naumann U. The art of differentiating computer programs. Society for industrial and applied mathematics, Philadelphia, USA, 2012.
3. Corliss G., Faure C., Griewank A., Hascoft L., Naumann U. Automatic Differentiation Bibliography // Automatic Differentiation of Algorithms: From Simulation Optimization. Springer, 2002. p. 383—425.

4. Олифер В. И. Усеченные гипер-дуальные числа в автоматическом дифференцировании. –URL:
http://viosolutions.amerihomesrealty.com/pdf/Усеченные_гипер-дуальные_числа_в_автоматическом_дифференцировании.pdf
5. Двайт Г.Б. Таблицы интегралов и другие математические формулы. - М.: "Наука", 1973. - 228 с.

Абстракт

В данной публикации рассматривается метод численного интегрирования на основе разложения в ряд Тейлора и автоматического дифференцирования с использованием специальных дуальных чисел (усеченных гипер-дуальных чисел). Представлена компьютерная реализация этого метода для языка SWIFT операционной системы macOS. Проведены численные эксперименты.

Ключевые слова: *численное интегрирование, усеченные гипер-дуальные числа, автоматическое дифференцирование, numerical integration, truncated hyper-dual numbers, automatic differentiation*

20 февраля 2021 г.