



引用格式:裴永臣, 关景晗, 王佳伟. 一种任意重积分自适应递归式快速计算方法[J]. 科学技术与工程, 2021, 21(7): 2595-2601.

Pei Yongchen, Guan Jinghan, Wang Jiawei. An adaptive recursive fast method for arbitrary multiple integral[J]. Science Technology and Engineering, 2021, 21(7): 2595-2601.

数学

一种任意重积分自适应递归式快速计算方法

裴永臣, 关景晗, 王佳伟

(吉林大学机械与航空航天工程学院, 长春 130025)

摘要 当前一些科学计算软件虽然能够利用数值积分得到复杂积分的近似解, 但最多只能求解三重积分。为解决积分计算重数受限问题, 基于累次积分的解析计算思路, 结合递归算法, 提出了一种任意重积分自适应递归式快速计算方法。从原理上介绍了算法思路和递归过程, 给出了在 MATLAB 中运行的算法源程序代码。在算例和工程应用上, 将该方法与现有方法进行了对比。结果表明了该方法的有效性, 可使计算时间大大缩短, 只需提供积分区间和被积函数即可求解, 省去了将程序代码与积分重数进行匹配的编程步骤。方法操作简单, 计算精度高, 速度快, 可满足实际工程需要。

关键词 数值积分; 多重积分; 递归算法; 工程应用

中图法分类号 O241.83; 文献标志码 A

An Adaptive Recursive Fast Method for Arbitrary Multiple Integral

PEI Yong-chen, GUAN Jing-han, WANG Jia-wei

(School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130025, China)

[Abstract] At present, some scientific calculation software can get approximate solution of complex integration through numerical integration. However, scientific calculation software can only solve triple integration at most. To solve the problem of limited multiplicity in integral calculation, an adaptive recursive fast calculation method for arbitrary multiple integrals was proposed. It was based on the analytic calculation idea of multiple integrals and recursive algorithm. The algorithm and recursion process of the method were introduced in principle, and the source code of the method was given in MATLAB. The method was compared with the existing methods by examples. The results indicate the effectiveness of the proposed method. This method can reduce the calculation time greatly. It can be solved only by providing integral interval and integrand function, eliminating the programming steps of matching program code with integral multiplicity. The method is simple in operation, and high in calculation accuracy and speed, which can meet the actual engineering needs.

[Keywords] numerical integration; multiple integral; recursive algorithm; engineering application

在各学科领域科学研究及工程应用中, 如电力系统动态仿真^[1]、准光学电磁场传播^[2]、磁力驱动器磁力求解^[3-4]等实际问题, 积分计算是必不可少的工具和环节。传统解析积分 Newton-Leibniz 公式只适用于初等原函数存在情况下的定积分求解, 而在实际应用中, 往往会出现无初等原函数、原函数极为复杂以及无解析表达式这三种情况, 如 e^{-x^2} 无初等原函数、 $x^2 / \sqrt{2x^2 + 3}$ 的原函数极为复杂、实验测得数据(无解析表达式)。数值积分通过收集积分区间内离散点及其对应函数值来逼近未知函数区间积分值, 可以用来求解复杂定积分问题的近

似值^[5]。

关于数值积分, 中外学者还进行了各方面研究^[6-8]。张建斌等^[9]设计出了基于 MATLAB 的数值积分求解器, 实现了数值积分的可视化功能。陈付龙^[10]讨论了二元数值积分的计算方法, 并给出了部分 C 语言代码。唐珺等^[11]将递归算法与自适应积分结合用于定积分计算, 与辛普森积分相比减少了计算量。还有学者利用遗传算法^[12]、神经网络算法^[13-15]、粒子群算法^[16]、混沌布谷鸟算法^[17]计算数值积分。Angulo 等^[18]将数值积分应用于红细胞生长模型研究。尽管涉及数值积分

收稿日期: 2020-03-16; 修订日期: 2020-09-14

基金项目: 国家自然科学基金(51675215); 吉林省自然科学基金(20170101060JC)

第一作者: 裴永臣(1980—), 男, 汉族, 吉林柳河人, 博士, 教授, 博士研究生导师。研究方向: 工程计算与分析方法。E-mail: peiyc@jlu.edu.cn。

的研究较多,但是对于三重以上积分数值计算的讨论相对较少。

此外,当前广泛使用的各类科学计算软件具备一定的数值积分计算功能,如著名的 MATLAB、Maple、Mathematica 等,其数值逼近方法大大加快了复杂积分计算速度。然而,这些科学计算软件自带函数所能计算的积分重数普遍也不高于三重。例如,MATLAB 软件虽然具有强大的矩阵运算能力,能够很好地胜任数据计算这一繁琐工作,但能够进行数值积分的自带函数只有 quad、dblquad、triplequad 三种,分别为一重、二重、三重积分函数,也就是说 MATLAB 自带函数最多只能计算三重积分,远不能满足科学的研究和工程应用需要。

为了提高科学计算软件的积分计算重数,根据多重积分的解析计算方式将累次积分的思想应用于数值积分中^[19],现提出一种任意重积分自适应递归式快速计算方法(简称“自适应递归法”)。首先,详细介绍了任意重积分的算法思路,并将递归过程分为递推(划分积分区间)和回归(求解函数值)两部分进行阐述。其次,以 MATLAB 为例,该方法由积分区间、多重积分函数和被积函数表达式三部分组成。其中,多重积分函数采用递归算法和自适应辛普森公式,具有独立性和自适应性,即当积分重数改变时多重积分函数能够自动适应变化,而无需重新进行编程,避免了重复性操作。最后,结合算例,讨论了该方法与近似三重积分法^[3]和辛普森法^[19]三者之间的差别。

1 算法思路

递归算法在程序设计中被经常用到。程序编写过程中如果程序运行时又调用了其本身,称这种编写方式为递归^[20]。采用递归算法可以将一个复杂问题一层一层转化为多个较原问题更为简单的小问题求解,有效地缩短了程序编写的代码长度。同时递归需要有递归出口,当满足递归出口的约束条件时,递归结束。

该方法可用于求解已知被积函数表达式的任意重积分,得到该积分数值解。具体计算步骤如下。

设任意重积分为

$$I = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} f(X_1, X_2, \dots, X_n, P_1, P_2, P_3, \dots) dX_1 dX_2 \cdots dX_n \quad (1)$$

式(1)中: a_1, a_2, \dots, a_n 为积分下限; b_1, b_2, \dots, b_n 为积分上限; X_1, X_2, \dots, X_n 为积分变量; P_1, P_2, P_3, \dots 为可变参数。

(1) 根据实际生产或研究需求,输入被积函

数为

$$f(X_1, X_2, \dots, X_n, P_1, P_2, P_3, \dots) \quad (2)$$

(2) 输入问题求解每个积分变量的积分参数(积分下限、积分上限和误差容限),并以积分参数矩阵的形式给出。积分参数矩阵 A 为 n 行 3 列矩阵,即

$$A = \begin{bmatrix} a_1 & b_1 & t_1 \\ a_2 & b_2 & t_2 \\ \vdots & \vdots & \vdots \\ a_n & b_n & t_n \end{bmatrix} \quad (3)$$

式(3)中:第 i 行元素 a_i, b_i, t_i 为第 i 个积分变量的积分下限、积分上限、误差容限。

(3) 设置递归出口,若积分参数矩阵 A 行数为 0,则递归终止。

(4) 读取积分参数矩阵 A 中第 1 个积分变量的积分参数。

(5) 删除积分参数矩阵 A 中第 1 个积分变量的积分参数,即删去第一行。

(6) 根据步骤(4)中读取的第 1 个积分变量的积分下限、积分上限,将积分区间划分成 m 份,即积分节点为 $m + 1$ 个。由步骤(4)、步骤(5)知,如若重复执行步骤(4)时,相当于第 i 次执行时读取的是积分参数矩阵 A 中第 i 个积分变量的积分参数。第 i 个积分变量的积分区间划分为

$$X_i = [a_i \ a_i + h_i \ a_i + 2h_i \cdots \frac{a_i + b_i}{2} \cdots b_i - 2h_i \ b_i - h_i \ b_i] \quad (4)$$

式(4)中: h_i 为第 i 个积分变量的积分步长, $h_i = \frac{1}{m}(b_i - a_i)$ 。

第 i 个积分变量的积分节点对应函数为

$$[f_{a_i} \ f_{a_i+h_i} \ f_{a_i+2h_i} \cdots f_{\frac{a_i+b_i}{2}} \cdots f_{b_i-2h_i} \ f_{b_i-h_i} \ f_{b_i}] \quad (5)$$

(7) 再次执行步骤(4)~步骤(6),直到满足步骤(3)中约束条件,递归终止,此时利用数值积分公式回归求解。

以上步骤的执行流程如图 1 所示。

2 递归过程

整个递归过程分为两步:第一步,递推,依次划分各积分变量的积分区间,从积分变量 X_1 到积分变量 X_n ;第二步,回归,反向求解各积分节点处函数值,从函数 f_n 到函数 f_1 。

2.1 递推:划分积分区间

求解任意重积分与定积分数值计算方法的具体实施过程相似。定积分数值计算方法先确定被

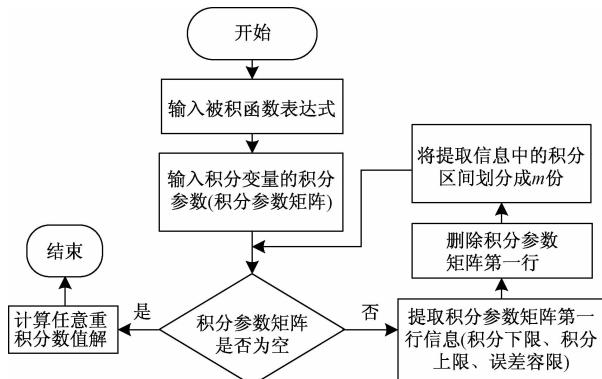


图 1 积分流程示意图

Fig. 1 Schematic diagram of integration process

划分后的积分节点及其所对应的函数值, 然后利用数值积分公式得到数值解。此外, 该方法与解析方法虽然在求解形式上有所不同, 但两者的求解思路存在相同之处, 即递归方法, 求解 n 重积分时, 先求解 $n-1$ 重积分; 而求解 $n-1$ 重积分时, 可先求 $n-2$ 重积分; 依次递推直到最后求解的是定积分。于是任意重积分问题可转化为定积分问题来求解。将任意重积分看作求 X_1 的积分, 将其他未赋值积分变量暂时看作已知量, 用 Y_{n-1} 表示, 此时的被积函数用 $f_1(X_1, Y_{n-1})$ 表示。于是式(1)又可写成下面的形式, 即

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} f(X_1, X_2, \dots, X_n, P_1, P_2, P_3, \dots) \times dX_1 dX_2 \cdots dX_n = \int_{a_1}^{b_1} f_1(X_1, Y_{n-1}) dX_1 \quad (6)$$

此时, 任意重积分转化成定积分问题, 利用定积分数值计算方法即可得到该问题数值解。然而, 被划分后的积分节点对应的函数值中含有未赋值积分变量 X_2, X_3, \dots, X_n , 无法确定 $f_1(X_1, Y_{n-1})$ 的值。因此, 只有当 $f_1(X_1, Y_{n-1})$ 的值确定后, 才能通过数值积分公式得到该问题数值解。由于 $f_1(X_1, Y_{n-1})$ 中含有未赋值积分变量, 故再次执行步骤(4)~步骤(6)。 $f_1(X_1, Y_{n-1})$ 中 X_1 已知, 所以求解 $f_1(X_1, Y_{n-1})$ 相当于求解 $n-1$ 重积分, 可以看作是 X_1 被赋值后求 X_2 的积分, 其他未赋值积分变量 X_3, \dots, X_n

X_4, \dots, X_n 暂时看作已知量, 用 Y_{n-2} 表示。此时的被积函数表达式用 $f_2(X_2, Y_{n-2})$ 表示。 $f_1(X_1, Y_{n-1})$ 与 $f_2(X_2, Y_{n-2})$ 关系为

$$f_1(X_1, Y_{n-1}) = \int_{a_2}^{b_2} f_2(X_2, Y_{n-2}) dX_2 \quad (7)$$

此时, $n-1$ 重积分转化成定积分问题, 利用定积分数值计算方法即得到该问题数值解。然而, 被划分后的积分节点对应的函数值中含有未赋值积分变量 X_3, X_4, \dots, X_n , 无法确定 $f_2(X_2, Y_{n-2})$ 的值。因此, 只有当 $f_2(X_2, Y_{n-2})$ 的值确定后, 才能通过数值积分公式得到该问题数值解。由于 $f_2(X_2, Y_{n-2})$ 中含有未赋值的积分变量, 故再次执行步骤(4)~步骤(6)。

以此类推, $f_i(X_i, Y_{n-i})$ 的求解相当于求解 $n-i$ 重积分, $\dots, f_{n-1}(X_{n-1}, Y_1)$ 的求解相当于求解定积分, 即

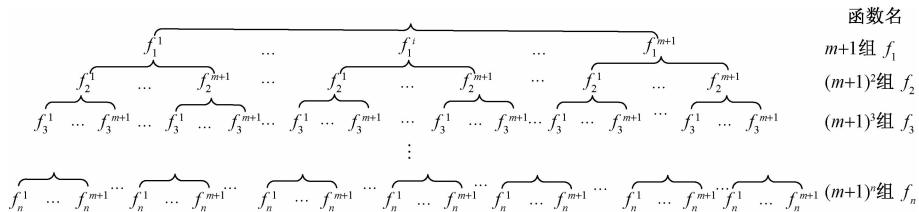
$$\begin{cases} f_i(X_i, Y_{n-i}) = \int_{a_i}^{b_i} f_{i+1}(X_{i+1}, Y_{n-i-1}) dX_{i+1}, \\ \quad i = 1, 2, \dots, n-1 \\ \vdots \\ f_{n-1}(X_{n-1}, Y_1) = \int_{a_n}^{b_n} f_n(X_n) dX_n \end{cases} \quad (8)$$

式中: Y_{n-i} 为 $X_{i+1}, X_{i+2}, \dots, X_n$ 的集合。

X_1 的各积分节点对应函数中 X_1 的赋值均为该积分节点, 故可将 $f_1(X_1, Y_{n-1})$ 分成 $m+1$ 组, 然后再根据 X_2 的积分节点将之前每组函数再次分成了 $m+1$ 组。以此类推, 直到 X_n 的积分节点将 X_{n-1} 每组函数再次分成了 $m+1$ 组。根据积分变量数和各积分变量的积分节点数, 最终求解的函数共有 $(m+1)^n$ 个。前后两次调用的函数求解关系如图 2 所示, 即前一次调用的积分变量的每个积分节点对应函数值是通过后一次调用的积分变量的所有积分节点对应函数值通过数值积分公式求解而得。

2.2 回归: 求解函数值

整个递归过程中函数求解关系可以通过图 2 得知, 函数中各积分变量的详细赋值如图 3 所示, 图 2 中倒数第一行函数值均可通过图 3 中相应位置以及其祖先节点所在位置的积分节点求得。



函数的上角标代表赋值的不同积分节点, 下角标与所求解积分变量的下角标相对应

图 2 各积分变量的积分节点对应函数树状图

Fig. 2 Tree diagram of the corresponding functions of the integration nodes of each integration variable

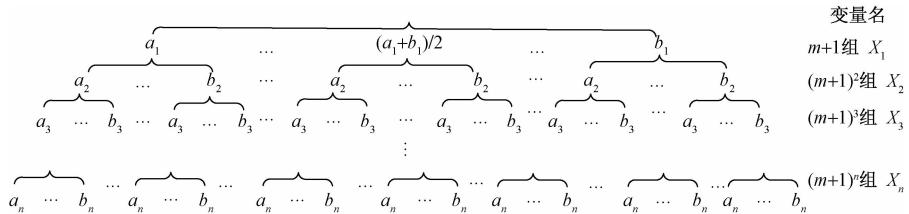


图3 积分节点对应函数的积分变量赋值树状图

Fig. 3 Tree diagram of integral variable assignment of function corresponding to integral node

结合图2、图3,图2中只有函数 f_n 中无未赋值积分变量。也就是说,仅当函数 f_n 的值确定后,才能通过数值积分公式进一步求解函数 f_{n-1} 的值。函数 f_{n-1} 的值确定后,可根据积分变量 X_{n-1} 的积分节点,求得函数 f_{n-2} 的值。以此类推,最终求得该任意重积分数值解。

3 MATLAB 算法实现

在 MATLAB 中,本文方法所用的数值积分基层算法是 quad 函数自带的自适应辛普森公式。程序共有三个脚本,文件名分别为 mm. m、multiquad. m、myfun. m,其中 mm. m 为主程序文件,multiquad. m 为多重积分函数 multiquad 定义文件(在计算中无需改动),myfun. m 为多变量被积函数 myfun 定义文件,运行时只需运行主程序即可。

3.1 主程序说明

该方法主程序如下:

```
clear;clc
```

```
P1 = * ; P2 = * ; ... % P1, P2,...是多变量被积函数 myfun 必要的可变参数
```

```
A = [a1,b1,t1;a2,b2,t2;...;ai,bi,ti;...;an,bn,tn] % 积分参数矩阵
```

```
Ip = multiquad([ ],[ ],A,@ myfun,P1,P2,...) % 调用多重积分函数 multiquad
```

3.2 多重积分函数 multiquad 定义

主程序中调用的多重积分函数 multiquad,定义为 $I = \text{multiquad}([],[],A,@ \text{myfun},P1,P2,...)$ 。前两个参数用于程序运行过程中的数据传递,主程序中初始设定为空矩阵,myfun 表示为多变量被积函数。多重积分函数 multiquad 以 MATLAB 自带积分函数 quad 为基层算法求解核心,因此具有自适应性,通过多次调用达到求解任意重积分目的。多重积分函数 multiquad 定义如下:

```
function [ I ] = multiquad( x,X0,A,int_fun,varargin )
a = A(1,1); % 积分下限
b = A(1,2); % 积分上限
t = A(1,3); % 误差容限
A(1,:) = [ ]; % 删除积分参数矩阵 A 第一行
if size(A,1) == 0
    I = zeros( size(x) );
    for i = 1:length(x)
```

```
I(i) = quad( @ f_multiquad,a,b,t,[ ],[ X0;x(i)],int_
fun,varargin{:}); % 使用函数 quad 进行计算
end
elseif size(A,1) >0
    if length(x) == 0
        I = quad( @ multiquad, a,b,t,[ ],X0,A,int_fun,varargin
{:}); % 使用函数 quad 进行计算
    else
        I = zeros( size(x) );
        for i = 1:length(x)
            I(i) = quad( @ multiquad,a,b,t,[ ],[ X0;x(i)],A,int_
fun,varargin{:}); % 使用函数 quad 进行计算
        end
    end
end
函数 f_multiquad 为多重积分函数 multiquad 的内部调用函数,用于计算积分区间划分后的各积分节点对应函数值。
function [ y ] = f_multiquad( x,X0,int_fun,varargin )
if size(X0,1) >0
    X = [ X0 * ones( size(x) );x ];
else
    X = [ x ];
end
for i = 1:size(X,2)
    y(i) = feval( int_fun,X(:,i),varargin{:}); % 计算指定
函数在某点的函数值
end
```

3.3 多变量被积函数 myfun 定义

多变量被积函数 myfun 括号中 X 为积分变量 X_1, X_2, \dots, X_n 的集合。多变量被积函数 myfun 接受 $X, P1, P2, \dots$ 并返回被积函数值 y。主函数中的多变量被积函数 myfun 定义如下:

```
function [ y ] = myfun( X,P1,P2,... )
X1 = X(1,:); % 积分变量 X1
X2 = X(2,:); % 积分变量 X2
:
Xn = X( n,:); % 积分变量 Xn
y = f( X1,X2,...,Xn,P1,P2,P3,...); % 被积函数 f( X1,X2,...,
Xn,P1,P2,P3,...)
```

4 算例分析

选取三种具有代表性的函数进行程序验证,并将自适应递归法与解析法,近似三重积分法^[3]和辛普森法^[19]在计算时间和精度上进行了对比。

例 1

$$\int_0^{0.5} \int_1^3 \int_0^0 \int_{-\frac{\pi}{3}}^{\pi} \int_1^1 [e^{2x_1} x_2 \sin(3x_3) + x_5^3/x_4] dx_1 dx_2 dx_3 dx_4 dx_5 = -\frac{4}{3} (e-1)^2 + \frac{\pi}{12} \approx -3.6748572.$$

例 2

$$\int_0^{\frac{\pi}{2}} \int_0^2 \int_0^1 \int_0^4 (3y^2 z \cos x + m) dx dy dz dm = 16 + 8\pi \approx 41.1327412.$$

例 3 工程应用

磁力驱动器凭借极高的密封性和无接触特性, 被广泛应用于制药、化工、石油、食品等行业^[3]。磁力驱动器的内外磁套之间的轴向磁力^[3]被写为

$$F_a = A \int_{R_1}^{R_2} \int_{R_3}^{R_4} \int_0^{2\pi} \int_0^{2\pi} A_1 r_1 r_2 dr_1 dr_2 d\alpha d\beta \quad (9)$$

式(9)中: r_1 为内磁套表面上任意一点的位置; r_2 为外磁套表面上任意一点的位置; α 为内磁套表面上任意一点的角度; β 为外磁套表面上任意一点的角度, $A = \frac{B_{r1} B_{r2}}{4\pi\mu_0}$, 其中 β 为外磁套表面上任意一点的角度, B_{r1} 为内磁套的剩余磁通密度, B_{r2} 为外磁套的剩余磁通密度, μ_0 为空间磁导率; A_1 表示为

$$A_1 = \frac{2\delta_0}{(A_2 + 2\delta_0^2)^{3/2}} - \frac{\delta_0 + \delta_2}{[A_2 + (\delta_0 + \delta_2)^2]^{3/2}} - \frac{\delta_0 - \delta_1}{[A_2 + (\delta_0 - \delta_1)^2]^{3/2}} \quad (10)$$

式(10)中: δ_0 为内磁套和外磁套之间的相对位移, $\delta_0 = 0.02$ m; δ_1 为内磁套厚度; δ_2 为外磁套厚度; A_2 表示为

$$A_2 = (r_2 \sin \beta - r_1 \sin \alpha)^2 + (r_2 \cos \beta - r_1 \cos \alpha)^2 \quad (11)$$

磁性套筒的材料参数和尺寸参考文献[3], 如表1所示。

表 1 磁性套筒的材料参数和尺寸**Table 1 Material parameters and dimensions of magnetic sleeve**

材料参数	材料尺寸
$B_{r1} = B_{r2} = 1.298$ T, $\mu_0 = 4\pi \times 10^{-7}$ T·m/A	$R_1 = 0.0150$ m, $R_2 = 0.0394$ m, $R_4 = 0.0810$ m, $R_3 = 0.0454$ m, $\delta_1 = 0.0340$ m, $\delta_2 = 0.0340$ m

以上三个算例的计算结果如表2所示。

通过表2得知, 例1和例2中, 辛普森法($h=0.05$)与自适应递归法($t_i=10^{-6}$)精度相同, 辛普森法($h=0.01$)与自适应递归法($t_i=10^{-7}$)精度相同, 但计算时间差距明显。例1中, 自适应递归法的计算时间与辛普森法相比, 分别缩短了69.6%和99.8%。例2中, 自适应递归法的计算时间与辛普森法相比, 分别缩短了87.8%和99.9%。例3中, 近似三重积分法计算磁力驱动器的轴向磁力耗时严重, 而辛普森法和自适应递归法能够将计算时间减低到20 s以内。值得注意的是, 例3中辛普森法的 h 只在0.01时有数值解。综合以上算例, 本文提出的自适应递归法的计算效率高于近似三重积分法和辛普森法, 主要是由于以下两点:

(1) 辛普森法中的步长 h 受积分区间的影响。 h 取值要小于各变量积分区间中的最小值, 如例3中 $h < 0.5$ 时则无法求解。这种情况导致了积分区间划分的不合理, 尤其是积分上下限差值较大时, 过小的 h 增加了程序的运行时间。此外, 辛普森法的积分区间划分方式固定, 步长 h 恒定, 增加了不必要的计算量, 而自适应递归法巧妙地将各变量积分区间进行了独立划分, 并单独设定误差限, 实现了计算量的合理分配。自适应递归法可以在保证精度要求的同时有效地降低计算量, 避免了辛普森法中因提升精度而导致计算时间呈指数形增长现象的发生。

表 2 例1、例2和例3的计算结果
Table 2 Calculation results of case 1, 2 and 3

方法	例1		例2		例3	
	时间/s	结果	时间/s	结果	时间/s	结果
解析法	—	-3.6748572	—	41.1327412	—	—
近似三重积分法	$n=100$	—	—	—	81.8	1 439.800 000 0
	$n=1000$	—	—	—	451	1 423.636 320 6
	$n=10000$	—	—	—	1 954	1 422.017 649 4
辛普森法	$h=0.1$	0.9	3.675 088 9	0.24	41.132 749 5	—
	$h=0.05$	27	-3.674 872 7	4.1	41.132 741 7	—
	$h=0.01$	41 700 s	-3.674 857 2	2 256 s	41.132 741 2	15.6
自适应递归法	$t_i=10^{-6}$	8.2	-3.674 844 0	0.5	41.132 740 6	0.6
	$t_i=10^{-7}$	100.0	-3.674 857 2	1.9	41.132 741 2	19.9

注: n 为近似三重积分法^[3]中沿 dr 方向的等分数; h 为辛普森法^[16]中的步长; t_i 为自适应递归法中各积分变量的误差容限; 例3中 $\delta_0=20$ mm; Inf表示经MATLAB计算后的结果为无穷大。

(2) 自适应递归法的优势在于输入各变量积分上下限和被积函数表达式后, 即可求解, 适应性强, 而近似三重积分和辛普森法缺少对积分重数的适应能力。近似三重积分仅能计算四重积分。采用辛普森法计算多重积分, 积分重数与程序中的函数 TNR_ITG^[16]为一一对应关系, 而不是自适应递归法中的多对一关系, 导致函数 TNR_ITG 的内容需根据积分重数的变化而不断调整。积分重数与函数 TNR_ITG 的改动幅度为正相关。

5 结论

针对当前科学计算软件求解积分重数的有限性及传统解析积分的局限性, 提出了一种任意重积分自适应递归式快速计算方法。该方法参考定积分计算方式, 依次将各积分变量的积分区间进行自适应划分, 当完全确定划分后各积分变量的积分节点时, 再反向逐层求积, 从而实现递归过程。

(1) 基于该方法给出了 MATLAB 中的算法源程序代码, 并进行了算例分析, 验证了该方法在多重积分求解时的高效率。

(2) 与现有方法的比较结果表明, 提出的自适应递归法不仅计算时间短、结果精度高, 效率高, 而且凭借对积分重数的适应能力, 省去了不断编写程序的繁琐过程, 操作更为方便, 具备通用性和普适性。

(3) 该方法是直线往复运动磁力驱动器磁力求解问题的一种更简洁、快速且准确的计算方法, 可应用于涉及积分运算的实际应用, 具有一定的工程应用价值。

参 考 文 献

- [1] 戴汉扬, 汤涌, 宋新立, 等. 电力系统动态仿真数值积分算法研究综述[J]. 电网技术, 2018, 42(12): 3977-3984.
Dai Hanyang, Tang Yong, Song Xinli, et al. Review on numerical integration algorithms for dynamic simulation of power system [J]. Power System Technology, 2018, 42(12): 3977-3984.
- [2] 梁彬, 白明, 金铭, 等. 瑞利索莫菲衍射与物理光学积分的应用比较[J]. 北京航空航天大学学报, 2012(9): 1214-1218.
Liang Bin, Bai Ming, Jin Ming, et al. Correlation between Rayleigh-Somerfield diffraction and physical optics integration [J]. Journal of Beijing University of Aeronautics and Astronautics, 2012(9): 1214-1218.
- [3] Li J G, Tan Q C, Zhang Y Q, et al. Study on the calculation of magnetic force based on the equivalent magnetic charge method[J]. Physics Procedia, 2012, 24(Part A): 190-197.
- [4] Li J G, Tan Q C, Pei Y C, et al. Influence of non-coaxial alignment on interaction forces for the contactless magnetic driver of a reciprocating motion[J]. International Journal of Applied Electromagnetic and Mechanics, 2015, 49(2): 169-177.
- [5] 张雨浓, 何良宇, 金龙, 等. 一组一点外推的数值积分公式
- 的提出与验证[J]. 数学的实践与认识, 2013, 43(8): 204-209.
Zhang Yunong, He Liangyu, Jin Long, et al. Proposing and verification of a group of 1-node-extrapolated numerical intergration formulas[J]. Mathematics in Practice and Theory, 2013, 43(8): 204-209.
- [6] 夏炎, 田波. Cowell 数值积分器的变步长与自起步方法[J]. 科学技术与工程, 2020, 20(17): 6742-6747.
Xia Yan, Tian Bo. The method of step size change and self-startup of Cowell integrator [J]. Science Technology and Engineering, 2020, 20(17): 6742-6747.
- [7] 张芮丹, 王辉, 吴昊, 等. 多参数积分化暂态高频电力变压器模型及其参数在线辨识方法[J]. 科学技术与工程, 2019, 19(21): 181-188.
Zhang Ruidan, Wang Hui, Wu Hao, et al. Multi-parameter integral transient high-frequency power transformer model and its parameter online identification method[J]. Science Technology and Engineering, 2019, 19(21): 181-188.
- [8] 李亚东, 张子军, 杨凤田, 等. 某型电动飞机起飞爬升性能分析及飞行试验[J]. 科学技术与工程, 2019, 19(35): 364-369.
Li Yadong, Zhang Zijun, Yang Fengtian, et al. Analysis of take-off and climb performance of a certain type of electric aircraft and flighting test[J]. Science Technology and Engineering, 2019, 19(35): 364-369.
- [9] 张建斌, 赵静, 许晓晴. 基于 MATLAB-GUI 的数值积分界面设计[J]. 实验室研究与探索, 2017, 36(1): 13-16.
Zhang Jianbin, Zhao Jing, Xu Xiaqing. The numerical integration design based on MATLAB-GUI[J]. Research and Exploration in Laboratory, 2017, 36(1): 13-16.
- [10] 陈付龙. 二元数值积分的计算方法[J]. 计算机工程与应用, 2007, 43(19): 32-34.
Chen Fulong. Two-dimensional numerical value integral's computing method[J]. Computer Engineering and Applications, 2007, 43(19): 32-34.
- [11] 唐珺, 杨道杰. 自适应积分的递归实现研究[J]. 中国高新技术企业, 2016(27): 13-16.
Tang Jun, Yang Daojie. Research on recursive implementation of adaptive integral (in Chinese)[J]. China High-Tech Enterprises, 2016(27): 13-16.
- [12] 赖志柱, 张云艳. 基于遗传算法求任意函数的数值积分[J]. 计算机工程与应用, 2014, 50(2): 54-57.
Lai Zhizhu, Zhang Yunyan. Solving numerical integration based on genetic algorithms [J]. Computer Engineering and Applications, 2014, 50(2): 54-57.
- [13] 肖红, 许少华, 李盼池. 基于数值积分的多聚合过程神经网络算法[J]. 信息与控制, 2013, 42(5): 608-621, 624.
Xiao Hong, Xu Shaohua, Li Panchi. Multi-aggregation process neural networks algorithm based on numerical integration[J]. Information and Control, 2013, 42(5): 608-621, 624.
- [14] 罗玉雄, 文卉. 一种基于神经网络算法的数值积分方法[J]. 传感技术学报, 2006, 19(4): 1187-1189, 1194.
Luo Yuxiong, Wen Hui. A numerical integration approach using neural network algorithm[J]. Chinese Journal of Sensors and Actuators, 2006, 19(4): 1187-1189, 1194.
- [15] 李盼池, 施光尧. 基于数值积分的离散过程神经网络算法及应用[J]. 系统工程理论与实践, 2013, 33(12): 3216-3222.

- Li Panchi, Shi Guangyao. Numerical integration-based discrete process neural networks algorithm and applications [J]. System Engineering-Theory & Practice, 2013, 33(12) : 3216-3222.
- [16] 韦杏琼, 周永权. 基于粒子群算法的数值积分方法研究 [J]. 微电子学与计算机, 2009, 26(7) : 117-119.
Wei Xingqiong, Zhou Yongquan. Numerical integral method research based on PSO [J]. Microelectronics & Computer, 2009, 26 (7) : 117-119.
- [17] 凌巍高, 欧 旭, 罗德相. 求解二重数值积分的混沌布谷鸟优化算法 [J]. 微电子学与计算机, 2014, 31(8) : 148-150, 154.
Ling Weigao, Ou Xu, Luo Dexiang. Two dimensional numerical integration based on chaotic cuckoo search optimization algorithm [J]. Microelectronics & Computer, 2014, 31 (8) : 148-150, 154.
- [18] Angulo O, Crauste F, López-Marcos J C. Numerical integration of an erythropoiesis model with explicit growth factor dynamics [J]. Journal of Computational and Applied Mathematics, 2018, 330 : 770-782.
- [19] 朱振广. 多重积分数值计算的一种递归方法 [J]. 辽宁工学院学报, 2004, 24(1) : 24-26, 39.
Zhu Zhenguang. A recurrence method of estimating multivariable integral [J]. Journal of Liaoning Institute of Technology, 2004, 24 (1) : 24-26, 39.
- [20] 魏 斌, 马继辉, 牛 虎. 基于递归算法的树型结构图的设计与实现 [J]. 计算机应用与软件, 2011, 28(1) : 96-98.
Wei Bin, Ma Jihui, Niu Hu. Design and implementation of recursion algorithm-based tree structure diagram [J]. Computer Applications and Software, 2011, 28(1) : 96-98.