DOI：10.19789/j.1004-9398.2025.04.005
文献引用：刘海峰，贾旭康，樊定成，等．双辛普森递推算法的研究与实现[J]．首都师范大学学报（自然科学版），2024，46（4）：37-44. LIU H F，JIA X K，FAN D C，et al. Research and implementation of double Simpson's recursive algorithm[J]. Journal of Capital Normal University（Natural Science Edition），2025，46（4）：37-44.

双辛普森递推算法的研究与实现*

刘海峰**，贾旭康，樊定成，梁星亮

（陕西科技大学数学与数据科学学院，陕西 西安　710021）

摘要：针对双梯形递推算法收敛速度慢、节点利用不充分的问题，提出一种精度更高、算法收敛更快的方法——双辛普森递推算法，由此推导出二重积分在矩形区域下区间逐次减半的双辛普森递推公式，并给出具体的算法实现以及应用该算法分别计算矩形区域和不规则区域下连续函数的二重积分问题。通过计算上半椭球曲面的面积，对比得出双辛普森递推算法比双梯形递推算法的收敛速度更快、计算精度更高、程序计算时间更短的结论。最后通过求解弦振动方程的初值问题，给出双辛普森递推算法在一般积分区域下二重积分计算的应用，从而说明双辛普森递推算法的适用性更强，进一步说明该算法在偏微分方程领域中的应用。

关键词：二重数值积分；双梯形递推算法；双辛普森递推方法；弦振动方程
中图分类号：TP301.6　　**文献标志码**：A

Research and implementation of double Simpson's recursive algorithm*

LIU Haifeng**，JIA Xukang，FAN Dingcheng，LIANG Xingliang

（School of Mathematics and Data Science，Shaanxi University of Science & Technology，Xi'an Shaanxi　710021）

Abstract：Aiming at the problems of slow convergence speed and insufficient node utilization of the double trapezoidal recursion algorithm，a method with higher accuracy and faster convergence of the algorithm—double Simpson recursion algorithm is proposed，from which the double Simpson recursion formula of double integration is halved in the interval under a rectangular region，and the specific algorithm implementation is given. The algorithm is applied to calculate the double integration problem of continuous functions under rectangular region and irregular region，respectively. By calculating the area of the upper semi-ellipsoidal surface，it is concluded that the double Simpson recursion algorithm has faster convergence speed，higher calculation accuracy and shorter program calculation time than the double trapezoidal recursion algorithm. Finally，by solving the initial value problem of the chord vibration equation，the application of the double Simpson recursion algorithm in the double integral calculation under the general integration region is given，which shows that the applicability of the double Simpson recursion algorithm is stronger，and the application of the algorithm in the field of partial differential equations is further explained.

收稿日期：2023-03-22
*国家自然科学基金项目（12001345）；陕西科技大学大学生创新创业训练计划项目（2023005）
**通信作者：liuhf@sust.edu.cn

Keywords: double numerical integration; double trapezoidal recursion method; double Simpsons recursive method; string vibration equation
CLC: TP301.6 **DC**: A

0 引 言

近年来,二重积分的数值解法问题备受国内外学者的关注。刘海峰和陈敏歌[1]首次根据定积分的复化梯形公式,推导出区间逐次减半的双梯形递推公式来数值求解二重积分,并给出双梯形递推算法的实现过程,此外,存在该算法计算二重积分的精度较差、算法的收敛性较慢等问题;宁桂英和周永权[2]利用类似于遗传算法的思想,提出差分进化算法来求解二重积分的近似值,这种算法虽然在二重积分被积表达式形式比较简单的情况下收敛速度较快、精度较高,但当二重积分的形式很复杂时,会出现程序的运行时间较慢、计算精度不是很好控制等问题,尤其表现在偏微分方程的求解方面;于丽妮[3]、李满枝等[4]、刘辉玲和叶锋[5]、黎锁平[6]采用概率论中几何概型的思想,将二重积分的计算转换成概率的计算,由此运用蒙特卡罗法来求二重积分的近似值。利用统计方法求二重积分的近似值主要是根据二重积分的几何意义:曲顶柱体的体积将体积的计算转化成概率的计算,但是由于概率本身是通过随机数体现的,而随机数中分布的选取可能会对二重积分的结果造成一定的误差,最常见的就是正态分布生成随机数的方法,但是如果一个二重积分本身就是带有统计色彩,实际问题中概率分布一般都是非参数统计类的分布,这样贸然选取就会出现众多问题;许小勇和周凤英[7]利用Laguerre小波及其积分算子矩阵,结合矩阵论的知识对二重积分进行数值求解,此算法主要是将二重积分的计算转换成矩阵的计算,这样如果矩阵中的元素的量级较大的话,可能会产生较大的误差;Ranjan和Dama-yanti[8]以及Siraj等[9]分别利用混合求积规则和Haar小波来数值求解二重积分的近似值;Abdulhameed和Memon[10]利用改进的梯形法则,来求解二重积分。另一方面,在实际应用中[11-15],二重积分的被积表达式或者其积分区域过于复杂,会出现无法利用牛顿-莱布尼茨公式直接求解或者很难求得其精确解的情况。由此可见,研究二重积分的数值解法对于求解二重积分至关重要。

为了解决区间逐次减半的双梯形递推算法计算精度较低、链表节点分裂次数较多、算法运行时间较长等问题,本文基于二重积分区间逐次减半的双梯形递推算法[1]来研究二重积分区间逐次减半的双辛普森递推算法,并与文献[1]中的双梯形递推算法进行对比。为体现双辛普森递推算法的优势所在,针对文献[17]将双辛普森递推算法应用到偏微分方程领域并给出弦振动方程的数值求解方法。通过对偏微分方程的求解,体现了双辛普森递推算法本身优势以及其研究意义所在。

1 双辛普森递推公式的理论推导

1.1 矩形区域上二重积分的双辛普森公式

对矩形区域$D=[a,b]\times[c,d]$,求下列连续函数$f(x,y)$的二重积分

$$I=\iint_D f(x,y)\mathrm{d}\sigma。\tag{1}$$

首先将积分区域D剖分:将x轴上$[a,b]$进行m等分,将y轴上$[c,d]$进行n等分,形成$m\times n$个小矩形区域$D_{ij}=[x_{i-1},x_i]\times[y_{j-1},y_j]$。其中:$x_i$、$y_j$分别为$x,y$方向上的分点且$x_i=a+ih_m$,$y_j=c+jt_n$。其中:$i=0,1,\cdots,m$;$j=0,1,\cdots,n$且$h_m=\dfrac{b-a}{m}$;$t_n=\dfrac{d-c}{n}$。于是在小矩形区域$D_{ij}$上对连续函数$f(x,y)$的二重积分为

$$I_{ij}=\iint_{D_{ij}} f(x,y)\mathrm{d}\sigma=\int_{y_{j-1}}^{y_j}\mathrm{d}y\int_{x_{i-1}}^{x_i} f(x,y)\mathrm{d}x。\tag{2}$$

分别在x、y方向上针对一元函数定积分均使用一维辛普森公式得到关于式(2)的二重积分在小矩形区域D_{ij}上的双辛普森求积公式为

$$\begin{aligned}I_{ij}\approx\frac{h_m}{6}\int_{y_{j-1}}^{y_j}[f(x_{i-1},y)+4f(x_{i-\frac{1}{2}},y)+f(x_i,y)]\mathrm{d}y\approx\\ \frac{h_m t_n}{36}[f(x_{i-1},y_{j-1})+4f(x_{i-1},y_{j-\frac{1}{2}})+f(x_{i-1},y_j)+\\ 4f(x_{i-\frac{1}{2}},y_{j-1})+16f(x_{i-\frac{1}{2}},y_{j-\frac{1}{2}})+4f(x_{i-\frac{1}{2}},y_j)+\\ f(x_i,y_{j-1})+4f(x_i,y_{j-\frac{1}{2}})+f(x_i,y_j)]。\end{aligned}\tag{3}$$

让i、j变动,反复使用式(3),再由重积分对积分

区域具有可加性,做进一步推演,即可得出对矩形区域D上的连续函数$f(x,y)$的二重积分的双辛普森求积公式为

$$I=\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}。\tag{4}$$

1.2　矩形区域上二重积分区间逐次减半双辛普森递推算法

在1.1节的基础上,对矩形区域D划分规则予以改变,进一步做第l次划分:将x方向上的区间$[a,b]$做2^l等分,则每个小区间的长度记为h_{2^l},分点为$a=x_0^{(l)}<x_1^{(l)}<\cdots<x_{2^l}^{(l)}=b$且满足$x_i^{(l)}=a+ih_{2^l}$ $(i=0,1,\cdots,2^l)$;同样对y方向上的区间$[c,d]$做2^l等分,则每个小区间的长度记为t_{2^l},分点为$c=y_0^{(l)}<y_1^{(l)}<\cdots<y_{2^l}^{(l)}=d$且满足$y_j^{(l)}=c+jt_{2^l}$ $(j=0,1,\cdots,2^l)$。于是在小矩形区域$[x_{i-1}^{(l)},x_i^{(l)}]\times[y_{j-1}^{(l)},y_j^{(l)}]$上对连续函数$f(x,y)$的二重积分为

$$\begin{aligned}I_{ij}^{(l)}\approx\frac{h_{2^l}t_{2^l}}{36}\big[&f(x_{i-1}^{(l)},y_{j-1}^{(l)})+4f(x_{i-1}^{(l)},y_{j-\frac12}^{(l)})+\\&f(x_{i-1}^{(l)},y_j^{(l)})+4f(x_{i-\frac12}^{(l)},y_{j-1}^{(l)})+\\&16f(x_{i-\frac12}^{(l)},y_{j-\frac12}^{(l)})+4f(x_{i-\frac12}^{(l)},y_j^{(l)})+\\&f(x_i^{(l)},y_{j-1}^{(l)})+4f(x_i^{(l)},y_{j-\frac12}^{(l)})+f(x_i^{(l)},y_j^{(l)})\big]。\end{aligned}\tag{5}$$

由于重积分对积分区域具有可加性,进一步推演,从而就得到在第l次区域划分时,对整个矩形区域D上的连续函数$f(x,y)$的二重积分双辛普森求积公式为

$$I^{(l)}=\sum_{i=1}^{2^l}\sum_{j=1}^{2^l}I_{ij}^{(l)}。\tag{6}$$

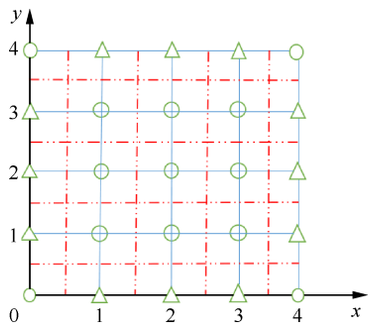
为了更方便、更高效地使用式(6)对二重积分进行数值求解,以下特将式(6)转化成递推公式进行求解。

引入集合来对式(6)进行分类合并。令$\Omega_x^l=\{x_0^{(l)},x_1^{(l)},\cdots,x_{2^l}^{(l)}\}$与$\Omega_y^l=\{y_0^{(l)},y_1^{(l)},\cdots,y_{2^l}^{(l)}\}$的笛卡尔积表示第$l$次区域划分时所有网格节点构成的集合$\Omega_{xy}^l=\Omega_x^l\times\Omega_y^l$。将这个网格节点集$\Omega_{xy}^l$分为6个子集,见图1。

(1)大矩形区域D的4个顶点构成的点集$\Omega_o^l=\{(x_0^{(l)},y_0^{(l)}),(x_0^{(l)},y_{2^l}^{(l)}),(x_{2^l}^{(l)},y_0^{(l)}),(x_{2^l}^{(l)},y_{2^l}^{(l)})\}$。

(2)所有剖分区域的小矩形在大矩形区域D上的边界点构成的点集$\Omega_\triangle^l=\{(x_0^{(l)},y_j^{(l)})|0<j<2^l\}\cup\{(x_i^{(l)},y_0^{(l)})|0<i<2^l\}\cup\{(x_{2^l}^{(l)},y_j^{(l)})|0<j<2^l\}\cup\{(x_i^{(l)},y_{2^l}^{(l)})|0<i<2^l\}$(即为图1中三角形的位置)。

(3)所有剖分区域的小矩形在大矩形区域D上的内部点(不包含内部的中间点)构成的点集$\Omega_\oplus^l=\{(x_i^{(l)},y_j^{(l)})|0<i<2^l,0<j<2^l\}$(即为图1中大矩形内部的小圆圈所在位置)。



注:○、△、⊕代表不同点集。

图1　区间分割l次积分区域的情形

(4)所有剖分区域的小矩形在大矩形区域D的边界同一个边界轴上任意2个端点的中点构成的点集

$$\Omega_\otimes^l=\left\{(x_0^{(l)},y_{j-\frac12}^{(l)})|1\leqslant j\leqslant 2^l\right\}\cup\left\{(x_{i-\frac12}^{(l)},y_0^{(l)})|1\leqslant i\leqslant 2\right\}^l\cup\left\{(x_{2^l}^{(l)},y_{j-\frac12}^{(l)})|1\leqslant j\leqslant 2^l\right\}\cup\left\{(x_{i-\frac12}^{(l)},y_{2^l}^{(l)})|1\leqslant i\right\}\leqslant 2^l,$$

即为图1中大矩形的4个边界上的实线与虚线的交点组成的集合。

(5)所有剖分区域的小矩形在大矩形区域D内部的同一个内部轴上任意2个端点的中点构成的点集

$$\Omega_\Psi^l=\left\{(x_i^{(l)},y_{j-\frac12}^{(l)})|0<i<2^l,1\leqslant j\leqslant 2^l\right\}\cup\left\{(x_{i-\frac12}^{(l)},y_j^{(l)})|1\leqslant i\leqslant 2^l,0<j<2^l\right\},$$

即为图1中大矩形内部实线与虚线以及虚线与实线的交点构成的集合)。

(6)所有剖分区域的小矩形内部中点构成的点集

$$\Omega_\square^l=\Omega_{xy}^l-\Omega_o^l-\Omega_\triangle^l-\Omega_\oplus^l-\Omega_\otimes^l-\Omega_\Psi^l,$$

即为图1中大矩形内部虚线与虚线的交点构成的集合。

根据前一次的区域划分与后一次的区域划分并利用上面所记符号将式(5)代入式(6)进行合并整理得到,在第l次区域划分时,对整个矩形区域D上的连续函数$f(x,y)$的二重积分的双辛普森求积公式为

$$\begin{aligned}I^{(l)}\approx\frac{h_{2^l}t_{2^l}}{36}\Bigg[&\sum_{P_o^{(l)}\in\Omega_o^l}f(P_o^{(l)})+2\sum_{P_\triangle^{(l)}\in\Omega_\triangle^l}f(P_\triangle^{(l)})+\\&4\sum_{P_\oplus^{(l)}\in\Omega_\oplus^l}f(P_\oplus^{(l)})+4\sum_{P_\otimes^{(l)}\in\Omega_\otimes^l}f(P_\otimes^{(l)})+\\&8\sum_{P_\Psi^{(l)}\in\Omega_\Psi^l}f(P_\Psi^{(l)})+16\sum_{P_\square^{(l)}\in\Omega_\square^l}f(P_\square^{(l)})\Bigg]。\end{aligned}\tag{7}$$

为了以下书写的方便,现将式(7)右端记为$Q^{(l)}$,即

$$Q^{(l)}=\frac{h_{2^l}t_{2^l}}{36}\Bigg[\sum_{P_o^{(l)}\in\Omega_o^l}f(P_o^{(l)})+2\sum_{P_\triangle^{(l)}\in\Omega_\triangle^l}f(P_\triangle^{(l)})+4\sum_{P_\oplus^{(l)}\in\Omega_\oplus^l}f(P_\oplus^{(l)})+4\sum_{P_\otimes^{(l)}\in\Omega_\otimes^l}f(P_\otimes^{(l)})+8\sum_{P_\Psi^{(l)}\in\Omega_\Psi^l}f(P_\Psi^{(l)})+16\sum_{P_\square^{(l)}\in\Omega_\square^l}f(P_\square^{(l)})\Bigg]。$$

同理可得,在第$l-1$次区域划分时,对整个矩形区域D上的连续函数$f(x,y)$的二重积分的双辛普森求积公式为

$$I^{(l-1)}\approx Q^{(l-1)}=\frac{h_{2^{l-1}}t_{2^{l-1}}}{36}\Bigg[\sum_{P_o^{(l-1)}\in\Omega_o^{l-1}}f(P_o^{(l-1)})+2\sum_{P_\triangle^{(l-1)}\in\Omega_\triangle^{l-1}}f(P_\triangle^{(l-1)})+4\sum_{P_\oplus^{(l-1)}\in\Omega_\oplus^{l-1}}f(P_\oplus^{(l-1)})+4\sum_{P_\otimes^{(l-1)}\in\Omega_\otimes^{l-1}}f(P_\otimes^{(l-1)})+8\sum_{P_\Psi^{(l-1)}\in\Omega_\Psi^{l-1}}f(P_\Psi^{(l-1)})+16\sum_{P_\square^{(l-1)}\in\Omega_\square^{l-1}}f(P_\square^{(l-1)})\Bigg]。$$

考虑到

$$h_{2^{l-1}}=2h_{2^l},\ t_{2^{l-1}}=2t_{2^l},\ \Omega_o^{l-1}=\Omega_o^l,$$

$$\Omega_\triangle^l=\Omega_\otimes^{l-1}\cup\Omega_\triangle^{l-1},\ \Omega_\oplus^l=\Omega_\oplus^{l-1}\cup\Omega_\Psi^{l-1}\cup\Omega_\square^{l-1},$$

及$\Omega_\otimes^l,\Omega_\Psi^l,\Omega_\square^l$是区域分裂时的新生点集,于是就有在第$l$次区域划分时,对整个矩形区域$D$上连续函数$f(x,y)$上的二重积分的双辛普森递推算法的求积公式为

$$Q^{(l)}=\frac{h_{2^l}t_{2^l}}{36}\Bigg\{\sum_{P_o^{(l-1)}\in\Omega_o^{l-1}}f(P_o^{(l-1)})+2\Bigg[\sum_{P_\triangle^{(l-1)}\in\Omega_\triangle^{l-1}}f(P_\triangle^{(l-1)})+\sum_{P_\otimes^{(l-1)}\in\Omega_\otimes^{l-1}}f(P_\otimes^{(l-1)})\Bigg]+4\Bigg[\sum_{P_\oplus^{(l-1)}\in\Omega_\oplus^{l-1}}f(P_\oplus^{(l-1)})+\sum_{P_\Psi^{(l-1)}\in\Omega_\Psi^{l-1}}f(P_\Psi^{(l-1)})+\sum_{P_\square^{(l-1)}\in\Omega_\square^{l-1}}f(P_\square^{(l-1)})\Bigg]+4\sum_{P_\otimes^{(l)}\in\Omega_\otimes^l}f(P_\otimes^{(l)})+8\sum_{P_\Psi^{(l)}\in\Omega_\Psi^l}f(P_\Psi^{(l)})+16\sum_{P_\square^{(l)}\in\Omega_\square^l}f(P_\square^{(l)})\Bigg\}。$$

即

$$Q^{(l)}=\frac{1}{4}Q^{(l-1)}+\frac{h_{2^l}t_{2^l}}{36}\Bigg[-2\sum_{P_\otimes^{(l-1)}\in\Omega_\otimes^{l-1}}f(P_\otimes^{(l-1)})+4\sum_{P_\otimes^{(l)}\in\Omega_\otimes^l}f(P_\otimes^{(l)})-4\sum_{P_\Psi^{(l-1)}\in\Omega_\Psi^{l-1}}f(P_\Psi^{(l-1)})+8\sum_{P_\Psi^{(l)}\in\Omega_\Psi^l}f(P_\Psi^{(l)})-12\sum_{P_\square^{(l-1)}\in\Omega_\square^{l-1}}f(P_\square^{(l-1)})+16\sum_{P_\square^{(l)}\in\Omega_\square^l}f(P_\square^{(l)})\Bigg]。\tag{8}$$

式中$l=1,2,\cdots,Q^{(l)}$表示对$I^{(l)}$的近似。

为了进行逐步迭代求解,此时还需要设置初值,即当$l=0$时,直接对矩形区域$D=[a,b]\times[c,d]$使用双辛普森公式得到

$$Q^{(0)}=\frac{(b-a)(d-c)}{36}\Bigg[f(a,c)+4f\left(\frac{a+b}{2},c\right)+f(b,c)+4f\left(a,\frac{c+d}{2}\right)+16f\left(\frac{a+b}{2},\frac{c+d}{2}\right)+4f\left(b,\frac{c+d}{2}\right)+f(a,d)+4f\left(\frac{a+b}{2},d\right)+f(b,d)\Bigg]。\tag{9}$$

1.3 不规则积分区域的计算方法

对于一般积分区域,求连续函数的二重积分问题[16],可以利用微元法的思想将一般积分区域分割成比较小的积分区域,此时这些小的积分区域就可以近似地看成矩形区域,然后在每个小矩形区域上应用双辛普森递推算法计算此连续函数的二重积分,再利用二重积分区域的可加性将这些二重积分的值相加,即可得到连续函数在一般区域下的二重积分计算的近似值。

2 二重积分双辛普森递推算法流程

该算法在Visual Studio 2022编译器中使用C语言进行算法的编写与实现。为了确保二重积分计算的准确性,使用的数据类型均为double类型,该算法的具体流程有如下步骤。

(1)定义链表节点的数据结构typedef struct node {double data;struct node* next;}LinkList。

(2)申请x、y方向上的链表头节点空间,分别用LinkList* XHead和LinkList* YHead指向这2个链表头节点。

(3)定义初始化x、y这2个方向上链表的函数double init_list(LinkList* head, double *f, double *1),其中*f指向x、y区间的上界值,*1指向x、y区间的下界值。初始化过程包括申请该方向的2个节点fnode和lnode,其中fnode节点存储x、y区间输入方向的上界值,lnode节点存储x、y区间输入方向的下界值,然后将lnode节点先插入到头节点后,fnode再插入到lnode节点后,最后分别返回x、y方向积分区间的长度。

(4)定义链表裂变函数void split(LinkList* head),申请一个新的节点空间,新节点存储相邻两个节点数据域的平均值,然后将该节点插入到这

2个相邻节点之间。

(5)定义被积函数double ff(double x,double y)。

(6)定义链表求长函数int Length(LinkList* head)。

(7)定义链表节点求和函数void newNode_add(LinkList* head1, LinkList* head2, double* lineMidPiont_m, double* InnerLinePiont_m, double* InnerMidPiont_m, double* lineMidPiont_m1, double* InnerLinePiont_m1, double* InnerMidPiont_m1),该函数lineMidPiont_m是链表第m次分裂积分区域的$\Omega_{\otimes}^{m}$点集(边界中点)的函数值之和;InnerLinePiont_m是链表第m次分裂积分区域的Ω_{ψ}^{m}(内界点)点集的函数值之和;InnerMidPiont_m是链表第m次分裂积分区域的$\Omega_{\square}^{m}$点集(内中点)的函数值之和;lineMidPiont_m1是链表第$m-1$次分裂积分区域的$\Omega_{\otimes}^{m-1}$点集的函数值之和;InnerLinePiont_m1是链表第$m-1$次分裂积分区域的Ω_{ψ}^{m-1}点集的函数值之和;InnerMidPiont_m1是链表第$m-1$次分裂积分区域的$\Omega_{\square}^{m-1}$点集的函数值之和。根据链表裂变的特征,除去矩形区域的4个顶点,m次分裂边界中点的值可以由2个相邻节点的平均值求出,若规定下标从0开始,$m-1$次分裂的边界中点就是m次分裂的下标为奇数的节点;m次分裂的内中点为划分后的小矩形区域的中间点,$m-1$次分裂的内中点为2个方向上的链表节点下标同时为奇数对应的点;m次分裂的内界点为划分后小矩形区域边界的中间点,具体特征为一个方向的链表节点下标为奇数,另一个方向的2个相邻节点的中间点所对应的点,而$m-1$次分裂的内中点具体特征为一个方向上的链表节点下标为奇数,另一个方向上的链表节点下标为偶数。根据上述节点的位置特征,可以实现该函数的具体功能。

(8)计算积分区域上的初始积分值。

(9)使用QueryPerformanceFrequency(LARGE_INTEGER *a)函数获得计算机高精度计时器的频率,接着定义LARGE_INTEGER start变量,调用QueryPerformanceCounter(&start)函数记录每次计算二重积分的开始时刻。

(10)利用for循环并结合二重积分的双辛普森递推算法的求积公式,计算每次分裂后的二重积分近似值,并将此次分裂的二重积分近似值与前一次分裂的二重积分近似值作差取绝对值作为此次区间分裂的二重积分计算精度,并用double类型的变量ee存储起来并输出。

(11)当每次迭代的精度达到所要求的精度或者最大节点分裂次数时,迭代终止计算。同时,定义变量LARGE_INTEGER over,使用QueryPerformanceCounter(&over)函数记录二重积分计算的结束时刻,输出最终计算的二重积分近似值以及程序运行所用时间(结束时间减去开始时间,以ms为单位)。

为了更直观地展示双辛普森递推算法的实现过程,将该算法通过流程图来描述,如图2所示。

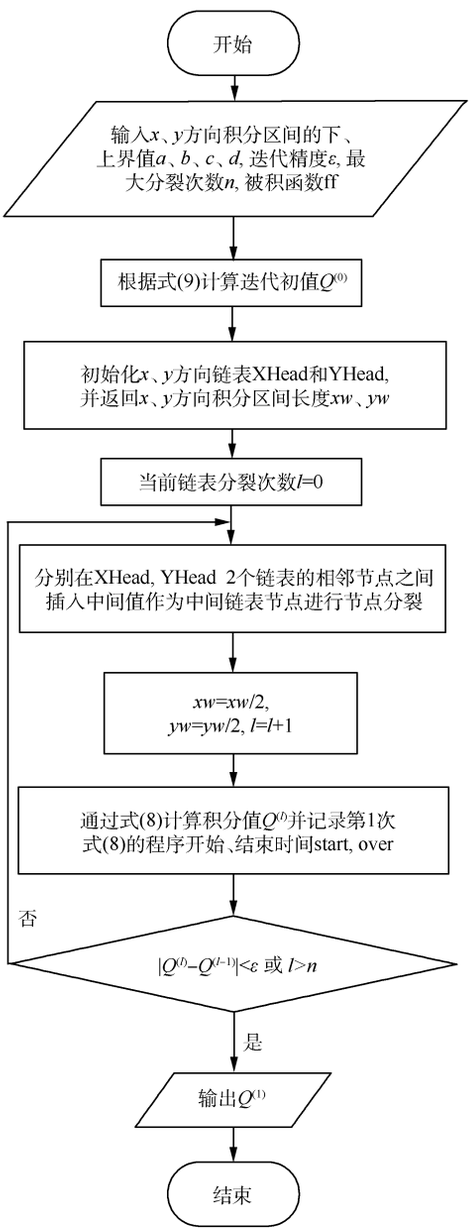


图2 二重积分双辛普森递推算法流程

3 计算案例以及算法复杂度分析

下面将对同一问题、在同一计算精度要求下、用同样Windows系统及其Visual Studio 2022编译

环境下即相同软硬件环境下进行双梯形递推算法和双辛普森递推算法的复杂度对比分析。

例题1　求矩形闭区域$[0,1]\times[0,2\pi]$上的二重积分[1]

$$\iint_{[0,1]\times[0,2\pi]} f(r,\theta)\,\mathrm{d}r\mathrm{d}\theta, \tag{10}$$

其中

$$f(r,\theta)=abr\sqrt{1+\frac{R^2r^2}{1-r^2}\left(\frac{\cos^2\theta}{a^2}+\frac{\sin^2\theta}{b^2}\right)},$$

且$\theta\in[0,2\pi]$, $r\in[0,1)$。

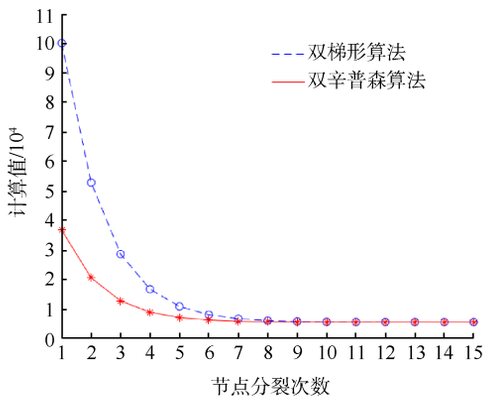
积分式(10)是著名的教堂顶部曲面(半椭球曲面)面积问题,由于式(10)是关于r的瑕点为1的二重瑕积分,因此取积分上限的极限值0.999 9来代替1。对于参数$a=30.6$, $b=29.6$, $R=30$,在相同的精度要求$|Q^{(l)}-Q^{(l-1)}|<10^{-2}$下,基于相同的软硬件计算环境,分别使用文献[1]中双梯形递推算法(算法1)和双辛普森递推算法(算法2)得到的积分值对比结果如表1所示。可知,在同一台计算机上,计算二重积分式(10)时,在节点分裂15次时双辛普森递推算法的计算结果已经满足精度要求,而双梯形递推算法在节点分裂17次时前后2次计算结果还有较大变化,不满足精度要求,这说明了双辛普森递推算法的收敛速度比双梯形递推算法的收敛速度更快。

表1　2种算法结果对比例题1

分裂次数	算法1		算法2	
	近似值	计算时间/ms	近似值	计算时间/ms
0	197 230.276	0	67 922.750	0
1	100 223.811	0.16	36 792.864	0.59
2	52 917.585	0.20	20 610.312	1.02
3	28 686.133	0.24	12 770.284	1.17
⋮	⋮	⋮	⋮	⋮
12	5 605.039	4.21×10^3	5 597.105	2.51×10^4
13	5 599.088	1.71×10^4	5 596.802	9.64×10^4
14	5 597.373	7.40×10^4	5 596.767	3.85×10^5
15	5 596.919	3.04×10^5	5 596.764	1.53×10^6
16	5 596.803	1.23×10^6	—	—
17	5 596.774	4.56×10^6	—	—

注:算法1代表双梯形递推算法;算法2代表双辛普森递推算法;—指无数据。

以下绘制出这2种算法的积分近似值与链表节点分裂次数之间的关系如图3所示。可知,在同一台计算机上,计算二重积分式(10)时,在初始节点分裂时,双辛普森递推算法的计算结果比双梯形递推算法更加快速接近精确值且双辛普森递推算法的积分近似值随节点分裂次数的变化幅度要比双梯形递推算法小得多,这说明双辛普森递推算法在节点分裂次数较少的情况下就能达到比较精确的计算结果,故双辛普森递推算法有更高的计算精度。



注:○、*为2种算法的节点。

图3　2种算法计算值随节点分裂次数的变化

由表1可知,算法2计算出$Q^{(15)}$时机器用时1.53×10^6 ms,同时$|Q^{(15)}-Q^{(14)}|\approx0.003<10^{-2}$;算法1计算出$Q^{(17)}$时机器用时$4.56\times10^6$ ms,这已经明显超过算法2计算出$Q^{(15)}$的机器用时,而相应的$|Q^{(17)}-Q^{(16)}|\approx0.029>10^{-2}$,即还未达到计算精度的停机要求,这种情况表明在相同的计算精度要求下,算法2(即双辛普森递推算法)需要的计算时间明显少于算法1(即双梯形递推算法)的计算时间,从而说明了双辛普森递推算法的时间优势。

在相同的区间分裂次数的情况下,计算下一个$Q^{(l)}$时,双辛普森递推算法需要计算新增节点函数值的节点个数明显多于双梯形递推算法需要计算新增节点函数值的节点个数。由此可知,并根据表1的结果也验证了双辛普森递推算法比双梯形递推算法具有更高的收敛速度。

例题2　为了说明双辛普森递推算法对二重积分的一般区域也成立,以下通过计算偏微分方程中的一维波动方程解析解的例子来体现该算法的应用。

一维波动方程是针对弦振动现象而提出的。假设弦是无限长,满足弦振动方程的模型假设条件[17],对如下的弦振动方程及其柯西问题

$$\begin{cases}\dfrac{\partial^2u}{\partial t^2}-\dfrac{\partial^2u}{\partial x^2}=t\sin x,\\ u|_{t=0}=0,\\ \dfrac{\partial u}{\partial t}|_{t=0}=\sin x。\end{cases} \tag{11}$$

根据二阶线性偏微分方程的有关知识,可以推导出柯西问题式(11)的解析解为

$$u(x,t)=\frac{1}{2}\left\{[\cos(x-t)-\cos(x+t)]+\iint\limits_{\substack{0\leqslant\tau\leqslant t\\ x-(t-\tau)\leqslant\xi\leqslant x+(t-\tau)}}\tau\sin\xi\mathrm{d}\xi\mathrm{d}\tau\right\}。\tag{12}$$

对式(12)中后面的三角区域的二重积分问题采用如 1.3 所述的一般区域的二重积分的双辛普森递推算法进行数值求解。

取 $x,t\in[0,10]$,2 个方向的步长均取 0.5,利用 1.3 节所述的微元法的思想,将每个三角区域分别按照不同的划分方式划分成很多小矩形区域,分别在每个小矩形区域上使用双辛普森递推公式,最终将这些值相加即可得出式(12)的一种划分规则下的数值解。同时,对式(12)直接积分得,其解析解为 $u(x,t)=t\sin x$。

为了量化划分区域的程度,采用划分的细度(即每个小矩形对角线长度的最大值)来描述划分区域的程度。在相同的精度 $|Q^{(l)}-Q^{(l-1)}|<10^{-6}$ 和软硬件环境下,绘制出不同划分方式的误差分布情况(图 4)。可知,随着划分细度的减小,式(12)的数值解与解析解的平均误差也随之减小,在划分细度为 0.007 1 m 处,式(12)的数值解与解析解的平均误差为 0.001 6,因此认为该划分方式为合理的区域划分方式。

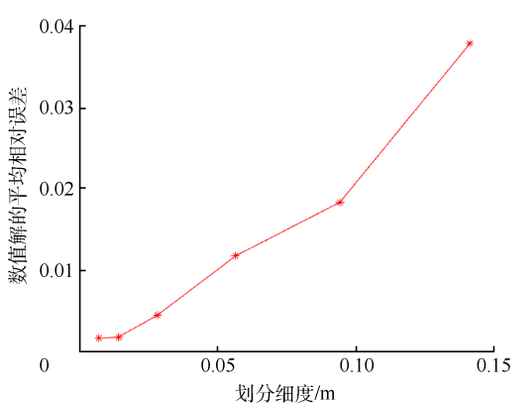


图 4　不同划分方式的误差分布

最终根据上述区域划分方式计算出利用双辛普森递推公式求解得到的式(12)数值解与解析解对比数据(表 2)。可知,利用双辛普森递推算法计算式(12)时得到的数值解与其解析解之间的平均相对误差大多都集中在 0.001 的量级,说明利用双辛普森递推算法计算一般积分区域下的二重积分式(12)的值具有更高的精确度。

表 2　双辛普森递推算法部分数值解与解析解比较

振动弦长的纵向位移/m	时间/s	解析解	数值解	平均相对误差
0.5	2.5	1.198 564	1.202 156	0.003 0
	5.0	2.397 128	2.416 973	0.008 3
	7.5	3.595 692	3.623 609	0.007 8
	10.0	4.794 255	4.865 215	0.014 8
5.0	2.5	−2.397 311	−2.397 794	0.000 2
	5.0	−4.794 621	−4.787 104	0.001 6
	7.5	−7.191 932	−7.186 910	0.000 7
	10.0	−9.589 243	−9.538 457	0.005 3
10.0	2.5	−1.360 053	−1.363 383	0.002 4
	5.0	−2.720 106	−2.730 059	0.003 7
	7.5	−4.080 158	−4.089 282	0.002 2
	10.0	−5.440 211	−5.500 707	0.011 1

为了更直观地看出表 2 数值解与解析解之间的差别,以下绘制出使用双辛普森递推算法计算出的数值解与解析解之间的对比图,以及相对误差随积分区域位置之间的关系如图 5 和图 6 所示。可知双辛普森递推算法只在个别点处相对误差较大,在大部分点相对误差较小;其数值解与解析解的吻合效果非常好,数值解基本上在解析解所处的曲面上,从而检验了一般区域下双辛普森递推算法计算出的结果的准确性和合理性。

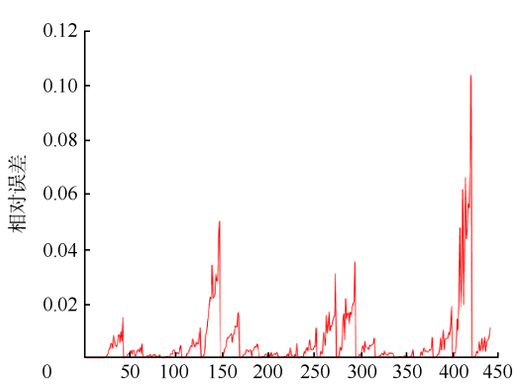


图 5　数值解与解析解相对误差对比

经计算最终得出通过最优的划分方式,说明了在相同的精度和相同软硬件计算环境下,对于一般积分区域使用 1.3 所述的方法计算二重积分近似值,双辛普森递推算法的计算精度较高,更加有利于计算更为复杂的问题,从而更能体现双辛普森递推算法在偏微分方程领域中的应用。

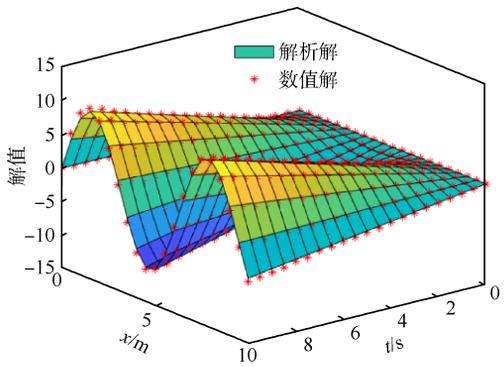


图6　波动方程初值问题对比

4　结束语

本文基于定积分的复化辛普森公式推导出二重积分的双辛普森递推公式,并给出了双辛普森递推算法实现的技术,得出了双辛普森递推算法的收敛速度比双梯形递推算法更快且其内存消耗更小的结论,同时在相同节点分裂次数下,双辛普森递推算法的舍入误差继承了上次分裂结果的1/4,因此大大提高了双辛普森递推算法的计算精度。由于其收敛速度快的原因,双辛普森递推算法计算时程序运行时间更短,这更能体现出双辛普森递推算法的优势所在。

参 考 文 献

[1] 刘海峰,陈敏歌.二重数值积分双梯形递推算法的研究与实现[J].计算机工程与应用,2006,42(15):94-96.
[2] 宁桂英,周永权.一种求解二重积分的差分进化算法[J].哈尔滨理工大学学报,2013,18(2):121-125.
[3] 于丽妮.基于MATLAB的蒙特卡罗方法求解二重积分[J].白城师范学院学报,2017,31(6):33-36.
[4] 李满枝,王洪涛,苗俊红.二重积分的Monte-Carlo数值仿真[J].计算机仿真,2011,28(5):114-117.
[5] 刘辉玲,叶锋.计算多重积分的均匀随机数蒙特卡罗法的实现[J].电脑知识与技术,2008,4(35):2289-2291.
[6] 黎锁平.运用蒙特卡罗方法求解随机性问题[J].甘肃工业大学学报,2001,27(2):95-97.
[7] 许小勇,周凤英.应用Laguerre小波算子矩阵求二重数值积分[J].计算机工程与应用,2017,53(16):45-49.
[8] RANJAN J S, DAMAYANTI N. A comparative study of numerical integration based on mixed quadrature rule and Haar wavelets [J]. Bulletin of Pure & Applied Sciences-Mathematics and Statistics, 2019, 38e (2): 532-539.
[9] LSLAM S U, AZIZ I, KHAN W. Numerical integration of multi-dimensional highly oscillatory, gentle oscillatory and non-oscillatory integrands based on wavelets and radial basis functions [J]. Engineering Analysis with Boundary Elements, 2012, 36 (8): 1284-1295.
[10] ABDULHAMEED A F, MEMON Q A. An improved Trapezoidal rule for numerical integration[J]. Journal of Physics(Conference Series),2021,2090(1):012104.
[11] 吴小庆,张合.2010年数学建模竞赛A题的计算方法研究[J].周口师范学院学报,2012,29(2):41-44.
[12] 赵小伟,徐刚,刘冀.用定积分方法求解倾斜偏转卧式储油罐容积[J].油气储运,2012,31(4):310-313.
[13] 闫明,张义民,李鹤,等.机械零件相关失效可靠度计算的二重积分模型[J].东北大学学报(自然科学版),2011,32(10):1460-1463.
[14] 黄春妙,汪继秀.二重积分在储油罐发生变位时的储油模型中的应用[J].科教导刊(上旬刊),2014(5):212-213.
[15] 裴礼文.数学分析中的典型问题与方法[M].3版.北京:高等教育出版社,2015.
[16] 徐萃薇,孙绳武.计算方法引论[M].4版.北京:高等教育出版社,2002.
[17] 谷超豪,李大潜,陈恕行,等.数学物理方程[M].3版.北京:高等教育出版社,2012.
[18] 陈纪修,於崇华,金路.数学分析(下册)[M].3版.北京:高等教育出版社,2019.

(责任编辑:马田田)

(上接第36页)

[18] MAY J P. The cohomology of restricted Lie algebras and of hopf algebras[J]. Journal of Algebra,1966,3(2):123-146.
[19] RAVENEL D C. Complex cobordism and stable homotopy groups of spheres[M]. New York: Academic Press,1986.
[20] LIU X G, WANG X J. A four-filtrated May spectral sequence and its applications [J]. Acta Mathematica Sinica(English Series),2008,24(9):1507-1524.
[21] ZHONG L N,WANG Y Y. Detection of a nontrivial product in the stable homotopy groups of spheres[J]. Algebraic and Geometric Topology,2013,13(5):3009-3029.

(责任编辑:马田田)