

# 一种基于泛函网络求数值积分方法研究

韦修喜<sup>1</sup> 周永权<sup>2</sup> 蓝晓玲<sup>1</sup>

(广西大学计算机与电子信息学院 南宁 530004)<sup>1</sup> (广西民族大学数学与计算机科学学院 南宁 530006)<sup>2</sup>

**摘要** 提出了一种基于泛函网络求数值积分新方法,给出了一种泛函网络模型及学习算法,并将该模型用于求任意函数的数值积分,理论上证明了泛函网络用于逼近数值积分定理。最后通过5个数值积分算例,并与传统计算方法作了比较分析,仿真结果表明,提出的数值积分方法精度高,适应性强,且不需要确定被积函数的原函数,因此该方法在工程技术中有较大的应用价值。

**关键词** 泛函网络,基函数,学习算法,函数逼近,数值积分

**中图分类号** TP183

## Numerical Integration Method Study Based on Function Network

WEI Xiu-xi<sup>1</sup> ZHOU Yong-quan<sup>2</sup> LAN Xiao-ling<sup>1</sup>

(College of Computer and Electron Information, Guangxi University, Nanning 530004, China)<sup>1</sup>

(College of Computer and Information Science, Guangxi University for Nationalities, Nanning 530006, China)<sup>2</sup>

**Abstract** A new method based on functional network was proposed to solve numerical integration methods. A functional network model for the numerical approximation and learning algorithm were given, and its model was used to solve numerical integration of arbitrary functions. The theorem of numerical integration solution based on functional network theorem was presented and proved. Finally, five experimental results show that the new measure has higher precision and strong adaptability, and primitive functions of integrand need not to be confirmed. Therefore, this new measure has application values in many engineering areas.

**Keywords** Functional network, Basis functions, Learning algorithm, Function approximation, Numerical integration

## 1 引言

在自然科学和生产实践中,有许多实际问题最后都归结为求数值积分问题,比如PID调节器就涉及到积分的计算。

在一定条件下,虽有Newton-Leibniz公式: $\int_a^b f(x)dx = F(b) - F(a)$ , ( $F'(x) = f(x)$ )可以计算定积分的值,但是,众所周知,除了有限的几种类型外,对于一般的被积函数 $f(x)$ 往往无法求得原函数,或者求得的原函数非常复杂。此外,在工程实际中,函数 $f(x)$ 是用函数表形式给出而没有解析表达式,这就更无法使用Newton-Leibniz公式了。因此有必要研究定积分的数值计算方法,以便解决定积分的近似计算。目前,有关数值积分的计算方法有很多,如Newton-Cotes方法,Romberg方法,Gauss方法等<sup>[1,2]</sup>,其中Newton-Cotes方法是一种利用插值多项式来构造数值积分的常用方法,但是高阶的Newton-Cotes方法的收敛性没有保证,因此,在实际计算中很少使用高阶的Newton-Cotes公式;Romberg方法收敛速度快、计算精度较高,但是计算量较大;Gauss方法积分精度高、数值稳定、收敛速度较快,但是节点与系数的计算较麻烦,而

且要求已知积分函数 $f(x)$ 。基于此,本文提出一种基于泛函网络的数值积分计算方法,其基本思想是训练泛函网络来逼近被积函数以实现定积分的数值计算。仿真结果表明,本文提出基于泛函网络的数值积分方法积分,具有精度高、收敛速度快、数值稳定等特点。甚至可以处理一些奇异积分的问题,而且不需要给定被积函数,因此能有效解决建模困难的系统或未知系统的求积分问题,在工程实际中有较大的应用价值。

## 2 泛函网络简介

泛函网络<sup>[3]</sup>是最近提出的一种对神经网络的有效推广。与神经网络不同,它处理的是一般的泛函模型,而不仅仅是Sigmoidal函数,并且在各个处理单元之间没有权值,神经元函数不是固定的,而是可学习的,是一个给定的函数簇的线性组合。我们可以根据特定的问题来选择不同的函数簇(例如:多项式、三角函数、Fourier展开级数等)。在文献[4]中,作者通过具体实例把泛函网络与神经网络进行比较,从仿真结果可以看出,泛函网络的性能优于神经网络。泛函网络已经成功地应用于很多方面,在文献[5]中作者用泛函网络来进行混沌时间序列的预测,提取被混沌信号屏蔽的信息,在文献

到稿日期:2008-05-13 本课题得到国家自然科学基金(60461001),广西自然科学基金(0542048;0832082),国家民委科研项目(08GX01),广西研究生教育创新计划资助项目(T32084)资助。

韦修喜(1982-),男,硕士研究生,主要从事神经网络及应用方面研究,E-mail:weixiuxi@163.com;周永权(1962-),男,博士,教授,主要研究方向为神经网络、计算智能及应用;蓝晓玲(1981-),女,硕士研究生,主要从事计算机应用方面研究。

[4]中作者用泛函网络来逼近微分、差分 and 泛函方程的解。泛函网络在上述的实验中都表现出了较好的性能。

图 1 给出了一个典型的泛函网络模型图,其主要组成部分如下:

- (1) 一个输入单元层。其功能是输入信息,在图 1 中输入层包含  $\{x_1, x_2, x_3\}$ 。
- (2) 若干中间存储单元层。它存储由神经元产生的信息,在图 1 中只有一个中间存储单元层,包含  $\{x_4, x_5\}$ 。
- (3) 一层或多层泛函神经元。每个泛函神经元是一个计算单元,对一个或多个输入的信息进行处理产生输出,在图 1 中有两层泛函神经元,分别为  $\{f_1, f_2\}$  和  $\{f_3\}$ 。
- (4) 一层输出层。包含输出信息,图 1 中为  $\{x_6\}$ 。
- (5) 一些直接连接。它们连接输入层、中间层神经元和层,在图 1 中用箭头表示,箭头的方向表示信息流的方向。

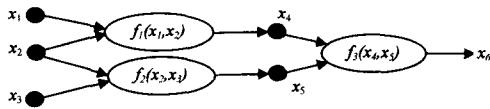


图 1 泛函网络模型图

泛函网络和神经网络的主要区别在于:首先,泛函网络单元之间的连接是直接连接而没有权值,而且泛函网络的神经元函数不像神经网络是固定的 Sigmoidal 函数,而是不固定的、可学习的。泛函网络和神经网络还有其它的区别,在此不再赘述,详细请参见文献[4]。

### 3 基于数值积分的泛函网络模型

泛函网络与神经网络一样,也有各种各样的结构,不可能用一个统一的通用的结构来描述所有的泛函网络,也不可能用一个统一的函数来表示所有的泛函网络。本文用于求解数值积分的泛函网络模型如图 2 所示。

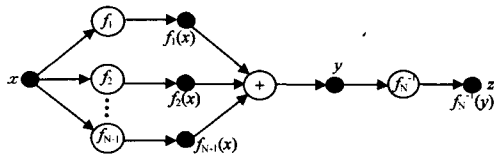


图 2 一种泛函网络用于求数值积分模型

图 2 给出的是一个单输入  $x$ , 单输出  $z$  的泛函网络模型,  $f_i, i=1, 2, \dots, N-1$  表示泛函神经元。该网络的输出表达式如下:

$$z = F(x) = \sum_{i=1}^N f_i(x) \quad (1)$$

设  $a_j$  为泛函网络参数,  $c_j(x)$  为基函数簇, 则泛函网络的参数矩阵  $a = [a_1, a_2, \dots, a_N]^T$ , 基函数矩阵为  $C(x) = (c_0(x), c_1(x), \dots, c_N(x))^T$ , 泛函网络输出为

$$y(x) = \sum_{j=0}^N a_j c_j(x) = a^T C(x) \quad (2)$$

误差函数为:

$$e(t) = f(x_t) - g(x_t), t=0, 1, \dots, m-1 \quad (3)$$

其中  $m$  为样本点数,  $f(x)$  为被积函数,  $g(x)$  为理想函数, 用来逼近被积函数  $f(x)$ 。

### 4 泛函网络学习算法

为了验证学习算法的正确性和有效性,我们定义平方根

误差(RMSE)来度量基于泛函网络求数值积分的误差:

$$RMSE = \sqrt{\frac{1}{p} \sum_{p=1}^r \|b_p - \hat{b}_p\|^2 / r} \quad (4)$$

其中,  $\hat{b}_p$  是泛函网络的输出,  $\|\cdot\|$  指函数的范数。

对于有教师训练学习算法来讲,给定一组学习样本数据  $\{(x_i^{(k)}, y_i^{(k)}) | k=1, 2, \dots, n\}$

误差代价函数可定义为

$$E_j = x_{2j} - \sum_{j=1}^n a_j \phi_j(x_{1j}) \quad (5)$$

为了找到最优的网络参数,需要最小化误差平方和

$$E = \sum_{k=1}^n E_k^2 = \sum_{k=1}^n [x_{2j} - \sum_{j=1}^n a_j \phi_j(x_{1j})]^2 \quad (6)$$

为了保证泛函网络表达式的唯一性,需要给出一些网络的初始值。在这里设网络的初始值为

$$f(x_{i0}) = u_{i0}, i=1, 2, \dots, n \quad (7)$$

加惩罚项式(8)到式(7)中,得到目标函数:

$$E = E_0 + \lambda_1 \sum_{i=1}^n (f(x_{i0}) - u_{i0})^2 \\ = \sum_{j=1}^n (x_{2j} - \sum_{i=1}^n a_i^T \phi_i(x_{1j}))^2 + \lambda_1 \sum_{i=1}^n (f(x_{i0}) - u_{i0})^2 \quad (8)$$

式(9)分别对  $a_i^T, \lambda_1$  求偏导,得

$$\begin{cases} \frac{\partial E}{\partial a_i^T} = -2 \sum_{j=1}^n E_j \phi_i(x_{1j}) + 2\lambda_1 (f_i(x_{i0}) - u_{i0}) \phi_i(x_{i0}) = 0 \\ \frac{\partial E}{\partial \lambda_1} = \sum_{i=1}^n a_i^T \phi_i(x_{1j}) - u_{i0} = 0 \end{cases} \quad (9)$$

在给定基函数后,通过解方程组(10),可求得最优的泛函网络参数  $a_i^T$ 。

### 5 基于泛函网络数值积分逼近定理

泛函网络具有很强的数值逼近能力,但是并不是选什么样的函数簇都可以逼近任意函数,用于逼近的函数簇的选取将直接关系到收敛速度,选取不当甚至导致不收敛。三角函数是初等函数,其积分值简单、易计算,而且在  $[-\pi, \pi]$  上是正交函数,其基函数的线性组合可实现对被积函数的逼近<sup>[6]</sup>。本节以三角基函数为例,给出数值积分定理,并给与证明。其它基函数泛函网络的积分定理及证明类似。设  $c_j(x)$  为三角基函数,即

$$c_j(x) = \begin{cases} \cos(jx), & j=0, 1, 2, \dots, N/2 \\ \sin[(j-N/2)x], & j=N/2+1, N/2+2, \dots, N \end{cases} \quad (10)$$

为泛函网络基函数簇( $N$  为偶数),且  $x \in [0, \pi]$ 。

定理 1 设  $\alpha, \beta$  为积分上下限,且  $0 \leq \alpha, \beta \leq \pi, a_j$  为泛函参数,则有

$$I = \int_{\alpha}^{\beta} f(x) dx \approx (\beta - \alpha) a_0 + \sum_{j=1}^{N/2} \frac{1}{j} a_j [\sin(j\beta) - \sin(j\alpha)] + \sum_{j=1}^{N/2} \frac{1}{j} a_{N/2+j} [\cos(j\alpha) - \cos(j\beta)]$$

证明:

$$I = \int_{\alpha}^{\beta} f(x) dx \approx \int_{\alpha}^{\beta} g(x) dx = \int_{\alpha}^{\beta} [a_0 + \sum_{j=1}^{N/2} a_j \cos(jx) + \sum_{j=1}^{N/2} a_{N/2+j} \sin(jx)] dx \\ = \int_{\alpha}^{\beta} [a_0 + \sum_{j=1}^{N/2} a_j \cos(jx) + \sum_{j=1}^{N/2} a_{N/2+j} \sin(jx)] dx$$

$$\begin{aligned}
 &= (\beta - \alpha)a_0 + \sum_{j=1}^{N/2} a_j \int_{\alpha}^{\beta} \cos(jx) dx + \\
 &\quad \sum_{j=1}^{N/2} a_{j+N/2} \int_{\alpha}^{\beta} \sin(jx) dx \\
 &= (\beta - \alpha)a_0 + \sum_{j=1}^{N/2} \frac{1}{j} a_j [\sin(j\beta) - \sin(j\alpha)] + \sum_{j=1}^{N/2} \\
 &\quad \frac{1}{j} a_{j+N/2} [\cos(j\alpha) - \cos(j\beta)]
 \end{aligned}$$

## 6 数值积分算例

为了验证本文提出的数值积分算法的优越性,本文选取了参考文献[7-9]中给出的一些实例进行计算,与传统的梯形法、Simpson方法、CompositeSimpson方法和Romberg方法比较,显然本文算法精度更高,适应性更强。

例1 文献[6]用梯形法和simpson方法在[0,2]积分区间分别计算被积函数 $x^2, x^4, \sqrt{1+x^2}, 1/(1+x), \sin x, e^x$ 等6个函数的积分,结果如文献[7]中的表4.7所示。

在本文算法中,我们选择图2所示的泛函网络结构,对所有的泛函神经元选取三角函数为基函数,输入个数 $N=10$ ,参数的初始值随机选取,泛函网络训练样本集为 $\{x_k = k\pi/100 | f(x_k), k=0, 1, \dots, 100\}$ ,其中 $f(x)$ 表示被积函数。表1列出了文献[7]、文献[11,12]的结果和本文算法的结果。

表1 几种数值积分方法对应函数的积分值

f(x)	$x^2$	$x^4$	$(1+x^2)^{1/2}$	$1/(1+x)$	$\sin x$	$e^x$
精确值	2.667	6.400	2.958	1.099	1.416	6.389
梯形法	4.000	16.000	3.326	1.333	0.909	8.389
Simpson方法	2.667	6.667	2.964	1.111	1.425	6.421
神经网络算法	2.665	6.393	2.959	1.101	1.415	6.388
不等距点分割ES法	2.666	6.398	2.9577	1.098	1.416	6.388
本文结果	2.667	6.3995	2.95789	1.0986	1.416	6.389

例2 计算积分 $\int_0^{48} \sqrt{1+\cos^2 x} dx$ 。

用Romberg方法计算该积分时遇到了困难<sup>[9]</sup>,用Composite Simpson's rule计算时,将积分区间[0,48]等分成100个子区间,计算结果为58.4708215。文献[10]中采用神经网络计算方法如下:

$$\begin{aligned}
 \int_0^{48} \sqrt{1+\cos^2 x} dx &= 15 \int_0^{\pi} \sqrt{1+\cos^2 x} dx + \\
 \int_0^{0.8761} \sqrt{1+\cos^2 x} dx &\approx 58.5209
 \end{aligned}$$

本文算法中,我们使用图2所示的泛函网络结构,对所有泛函神经元选取基函数 $\phi(p)=[1, p, p^2, \sin p, \cos p, \sin 2p, \cos 2p]$ ,参数的初始值随机选取,泛函网络训练样本集为 $\{x_k = k/2 | f(x_k), k=0, 1, \dots, 100\}$ ,其中 $f(x)$ 表示被积函数,计算结果为58.4705。表2列出了文献[9-11]中的结果和本文算法的计算结果。积分误差变化曲线如图3所示。

表2 几种数值积分方法对应函数的积分值

f(x)	Simpson	神经网络算法	ES方法	本文算法	精确值
$\sqrt{1+\cos^2 x}$	58.47082	58.5205	58.47065	58.4705	58.4704691

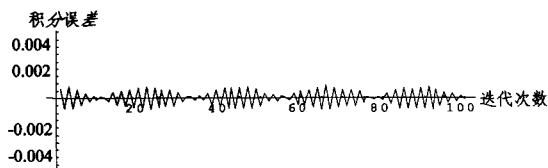


图3 积分误差变化曲线

例3 为了检验本文算法具有奇异积的处理能力,考虑如下奇异函数

$$f(x) = \begin{cases} e^{-x}, & 0 \leq x < 1 \\ e^{-x/2}, & 1 \leq x < 2 \\ e^{-x/3}, & 2 \leq x \leq 3 \end{cases}$$

该函数的精确积分值为1.546036。文献[10]中采用神经网络计算得到的结果为1.5467。本文算法中,我们使用图2所示的泛函网络结构,对所有的泛函神经元选取三角函数为基函数,输入个数 $N=10$ ,参数的初始值随机选取,泛函网络训练样本集为 $\{x_k = k\pi/100 | f(x_k), k=0, 1, \dots, 100\}$ ,其中 $f(x)$ 表示被积函数,计算结果为1.546036。表3列出了文献[10,11]中的结果和本文算法的计算结果。积分误差变化曲线如图4所示。

表3 几种数值积分方法对应函数的积分值

f(x)	神经网络算法	ES方法	本文算法	精确值
$e^{-x^i}, i=1, 2, 3$	1.5467	1.5459805	1.54604	1.546036

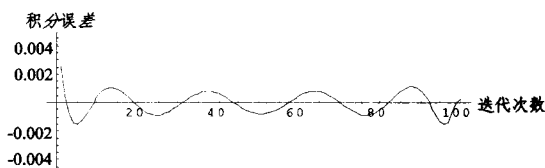


图4 积分误差变化曲线

例4 计算积分 $\int_0^1 e^{-x^2} dx$ 。

此积分的被积函数的原函数不是初等函数,所以无法用牛顿-莱布尼茨公式来计算这个积分。文献[10]中分别用矩形法、梯形法和Simpson方法计算该积分。

本文算法中,我们使用图2所示的泛函网络结构,对所有的泛函神经元选取多项式 $\phi(p)=[1, p, p^2, p^3]$ ,参数的初始值随机选取,泛函网络训练样本集为 $\{x_k = k/100 | f(x_k), k=0, 1, \dots, 100\}$ ,其中 $f(x)$ 表示被积函数,计算结果为0.746823。表4列出了文献[9-11]中的结果和本文算法的计算结果。积分误差变化曲线如图5所示。

表4 几种数值积分方法对应函数的积分值

f(x)	矩形法	梯形法	Simpson	ES方法	本文算法	精确值
$e^{-x^2}$	0.77782	0.74621	0.74683	0.74683	0.746823	0.746824

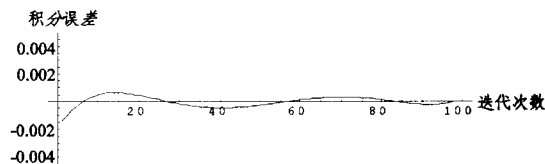


图5 积分误差变化曲线

例5 计算积分 $I = \int_{-1}^1 \frac{x+1}{\sqrt{3x+1}} dx$ 。

该函数的精确积分值是1.8294,文献[12]中用Gauss-Legendre公式计算得到1.436,用Gauss-Per Kai公式计算得到1.7533。

考虑到被积函数的积分上下限 $[-1, 1] \notin [0, \pi]$ ,因此本

(下转第238页)

1.45675898761, - 1.5678456772, 0.66765756893, -0.75678765688), 与最优解对应的最优值  $g(x) = 0.055894567$ , 对两个问题测试后所得到的解的均值及最差解如表 1 所示。

表 1 MOLEA 算法测试解

Value problems	Best	Mean	Worst
Test1	5198.5367	5199.3367	5199.9689
Test2	0.055894567	0.055999745	0.055999987

**结束语** 本文提出了处理约束优化问题的一种新方法——多目标线性进化算法(MOLEA),该方法将约束优化问题在高维搜索空间的求解转换到二维搜索空间中。利用给出的线性适应度函数,方便了对个体适应度值的评价。设计的基于密度的交叉算子可以较好地扩大解的搜索空间。数值实验显示出新算法是一种便于实现、通用性较强的方法。下一步我们的工作将集中在二维搜索空间中利用分类的方法改进 MOLEA,以提高算法的性能,并试图推广到其它的组合优化问题。

### 参考文献

[1] Deb K. Multi-objective genetic algorithms: problem difficulties and construction of test problem[J]. Evolutionary Computa-

tion,1999,7(3):205-230

[2] 崔逊学,林闯,方廷健.多目标进化算法的研究与进展[J].模式识别与人工智能,2003,16(3):306-313  
 [3] Deb K, Agrawal S, Pratap A, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197  
 [4] Beyer H-G, Deb K. On self-adaptive features in real-parameter evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation, 2001, 5(3): 250-270  
 [5] 边肇祺,张学工.模式识别.第二版[M].北京:清华大学出版社,2001:83-117  
 [6] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithm: Empirical results[J]. Evolutionary Computation Journal, 2000, 8(2): 125-148  
 [7] Michalewicz Z. Genetic algorithms, Numerical optimization and constraints// Eshelman L J. ed. Proceedings of the 6th International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann Publishers, 1995: 151-158  
 [8] 邹秀芬,刘敏忠,吴志健,等.解约束多目标优化问题的一种鲁棒的进化算法[J].计算机研究与发展,2004,41(6):986-990  
 [9] Deb K. Multi-objective Optimization Using Evolutionary Algorithms. Chichester, UK: John Wiley & Sons, Ltd, 2001  
 [10] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary computation, 2000, 4(3): 284-294

(上接第 226 页)

文计算方法如下:

$$\int_{-1}^1 \frac{1+x}{\sqrt[3]{3x+1}} dx = \int_{-1}^0 \frac{1+x}{\sqrt[3]{3x+1}} dx + \int_0^1 \frac{1+x}{\sqrt[3]{3x+1}} dx = \int_0^1 \frac{x}{\sqrt[3]{3x-2}} dx + \int_0^1 \frac{1+x}{\sqrt[3]{3x+1}} dx$$

本文算法中,使用图 2 所示的泛函网络结构,对所有的泛函神经元选取三角函数为基函数,参数的初始值随机选取,输入个数  $N=10$ ,泛函网络训练样本集为  $\{x_k = k\pi/100 | f(x_k), k=0,1,\dots,100\}$ ,计算结果为  $I=1.11190+0.71748i=1.82938$ 。表 5 列出了文献[12]中的结果和本文算法的计算结果。积分误差变化曲线如图 6 所示。

表 5 几种数值积分方法对应函数的积分值

f(x)	Gauss-Legendre 公式	Gauss-Per Kai 公式	本文算法	精确值
$(1+x)/\sqrt[3]{3x+1}$	1.436	1.7533	1.82938	1.8294

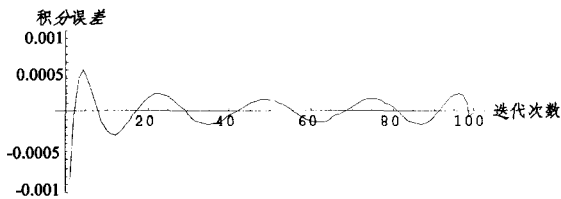


图 6 积分误差变化曲线

**结束语** 泛函网络是一个崭新的研究方向,其在某些方面的性能优于神经网络。本文给出了基于泛函网络求解数值积分的模型、算法,该算法适合于求任意函数的数值积分,因此,本文提出的数值积分计算方法适应性强。通过仿真实例可以看出,与传统计算方法相比,本文的计算精度更高,而且本文给出的积分算法不限制积分区间,是一种有效的数值积

分方法,在工程应用领域中具有较大的应用前景。

### 参考文献

[1] 林成森.数值计算方法[M].北京:科学出版社,1998:173-215  
 [2] 朱方生,刘大美,李素贞.计算方法[M].武汉:武汉大学出版社,2003  
 [3] Castillo E. Functional networks. Neural Processing Letters, 1998, 7: 151-159  
 [4] Castello E, Cobo A, Gutierrez J M. Working with Differential Functional and Difference Equations Using Functional Networks [J]. Appl. Math. Model, 1999, 23: 89-107  
 [5] Castello E, Gutierrez J M. Nonlinear Time Series Modeling and Prediction Using Functional Networks, Extracting Information Masked by Chaos[J]. Phys. Lett. A, 1998, 244: 71-84  
 [6] Yang Shiow - Shung, Tseng Ching - Shiuw. An orthogonal neural network for function approximation. IEEE Transaction on Systems, Man, and Cybernetics, Part B; Cybernetics, 1996, 26(5): 779-785  
 [7] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 190  
 [8] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 212  
 [9] Burden R L, Faires J D. Numerical Analysis (Seventh Edition) [M]. Brooks/Cole, Thomson Learning, Inc., 2001: 206, 772  
 [10] Wang X H, He Y G, Zeng Z Z. Numerical integration study based on triangle basis neural network algorithm[J]. Journal of Electronics & Information Technology, 2004, 26(3): 394-399  
 [11] 周永权,张明,赵斌.基于进化策略方法求任意函数的数值积分[J].计算机学报,2008,31(2):196-206  
 [12] 朱磊.关于 Gauss\_PerKai 型数值积分方法[J].合肥工业大学学报:自然科学版,2007,30(5):655-656