上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：快速高斯和算法

学生姓名：高梓轩

学生学号：517021910472

专　　业：数学与应用数学 (致远荣誉计划)

指导教师：徐振礼

学 院 (系)：数学科学学院、致远学院

上海交通大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文《快速高斯和算法》，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在___年解密后适用本授权书。

本学位论文属于

不保密□。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

快速高斯和算法

摘　要

在科学计算和工程计算的许多应用中，为了构造有效的核求和或卷积算法，经常需要用高斯和或指数和来逼近相互作用的核．在本文中，通过引入 de la Vallée-Poussin 和与 Chebyshev 多项式，我们提出了一种与核函数无关的高斯和 (Sum-Of-Gaussian, SOG) 逼近方法．SOG 适用于一般的相互作用核函数并且高斯带宽的下界是可控制的，因此高斯带宽可以很容易用快速高斯算法求和．高斯项数还可以通过基于平方根法的平衡截断的模型缩减进行减少．数值结果表明了模型的精度和模型降阶的效率，算法具有良好的性能．我们称之为 VPMR 算法．其中 VP 表示 de la Vallée-Poussin 和, MR 则表示模型降阶 (Model Reduction,MR)．这种算法也可以通过变量代换完成指数和 (Sum-Of-Exponential, SOE) 估计．本文针对 VPMR 算法对于 SOG 估计给出了详细的误差分析，结合理论分析和实际实验两方面证明算法高精度的性质．

在已有结果的基础上，我们又开发了一种基于 SOE 的快速的时间卷积积分数值算法，允许对 N 个时间步长进行 N 阶计算复杂度来逼近一个连续的时间卷积积分．如果核函数奇异，我们则利用截断操作将核函数分为奇异与非奇异两部分．对于分裂卷积核的非奇异部分，卷积积分可以作为一个常微分方程系统利用 Runge-Kutta 方法解决．其余的奇异部分则利用广义泰勒展开进行显式逼近．这一算法的优点来源于 VPMR 算法的 SOE 估计是有效的，准确的，并且其带宽可控．我们对基于 SOE 的卷积积分与卷积积分方程进行了数值分析．在不同核函数上的数值结果表明，时间卷积积分和时间卷积积分方程在精度和效率上都表现出了该方法的良好的性能．此外，我们还研究了 SOG 在快速高斯变换算法中的应用．通过理论分析和实际实验，证明了这种耦合方式的优越性．

关键词：高斯和估计，相互作用核函数，de la Vallée-Poussin 和，模型降阶，时间卷积积分，快速高斯变换

A FAST SUM-OF-GAUSSIAN METHOD

ABSTRACT

Approximation of interacting kernels by sum of Gaussians (SOG) or sum of exponentials (SOE) is frequently required in many applications of scientific and engineering computing in order to construct efficient algorithms for kernel summation or convolution problems. In this paper, we propose a kernel-independent SOG(SOE) method by introducing the de la Vallée-Poussin sum and Chebyshev polynomials. The SOG works for general interacting kernels and the lower bound of Gaussian bandwidths is tunable and thus the Gaussians can be easily summed by fast Gaussian algorithms. The number of Gaussians can be further reduced via the model reduction based on the balanced truncation based on the square root method. Numerical results on the accuracy and model reduction efficiency show attractive performance of the proposed method. We call the algorithm VPMR. VP represents the de la Vallée-Poussin sum and MR represents the model reduction. This algorithm can perform sum-of-exponential (SOE) estimation by variable substitution. VPMR can perform sum-of-exponential (SOE) estimation by variable substitution. In this paper, the error analysis of VPMR algorithm for SOG estimation is given in detail, and the high precision of the algorithm is proved by combining with practical experiments.

On the basis of existing results, we develop a fast algorithm for convolution quadrature based on the SOE, which allows an order N calculation for N time steps of approximating a continuous temporal convolution integral.We employ the SOE expansion for the finite part of the splitting convolution kernel such that the convolution integral can be solved as a system of ordinary differential equations due to the exponential kernels. The remaining part is explicitly approximated by employing the generalized Taylor expansion. The significant features of our algorithm are that the SOE method is efficient and accurate, and works for general kernels with controllable upperbound of positive exponents. We provide numerical analysis for the SOE-based convolution quadrature. Numerical results on different kernels, the convolution integral and integral equations demonstrate attractive performance of both accuracy and efficiency of the proposed method. In addition, we also examined the application of SOG to FGT. Through theoretical analysis and practical experiments, we prove the advantages of this coupling.

Key words: Sum-of-Gaussians, interaction kernels, de la Vallée-Poussin sums, model reduction, convolution integral, fast Gauss transform

目 录

插图索引

表格索引

算法索引

第一章　概述

对于在有限区间 D 内给定的函数 $f(x)$，与一个估计的误差阈值 ϵ，本文旨在考虑函数 $f(x)$ 的高斯和 (Sum-Of-Gaussian,SOG) 展开

$$\max_{x\in D}\left|f(x)-\sum_j w_j e^{-t_j x^2}\right| < \varepsilon \max_{x\in D}|f(x)|, \tag{1–1}$$

其中 w_j 和 $1/\sqrt{t_j}$ 被分别称为第 j 项的权重和带宽. 在过去几十年中, SOG 技术引起了广泛的关注. 因为它可以在许多科学计算的应用中发挥作用，诸如物理空间中的卷积积分[1-3]，求和问题[4-5]，以及非反射边界条件的波动方程问题[6-9]. 上述问题的许多核函数实际上都具备径向基的形式，即 $f(x) = f(\|\boldsymbol{x}\|), \boldsymbol{x} \in \mathbb{R}^d$，例如幂核函数 $\|\boldsymbol{x}\|^{-s}, s > 0$, Hardy 核函数 $\sqrt{\|\boldsymbol{x}\|^2+s^2}$, Matérn 核函数[10] 等. 对这些核函数的 SOG 估计可以极大降低在对应问题中的运算成本，因为高斯核函数可以利用展开而使得不同维度的变量分离，从而对核函数 $f(x)$ 的卷积可以处理为若干个一维卷积问题的加和.

公式(1–1)的求解在文献[11-21] 中有较为深刻的研究. 目前解决这个问题的方法主要有两种. 其一为利用函数 $f(x)$ 的 Laplace 变换的最佳有理近似. Laplace 变换的线性性质与指数函数的拉普拉斯变换为有理函数为这种方法提供了理论可能. 径向基函数的换元 $y=\sqrt{x}$ 使得得到的指数和展开 (Sum-Of-Exponential,SOE) 可以转化为 SOG. 如果最后的结果对于带宽没有要求. 这种方法是一种高效的算法. 例如. 幂核函数 x^{-s} 可以有如下的逆 Laplace 变换式[12],

$$x^{-s} = \frac{1}{\Gamma(s)}\int_{-\infty}^{\infty} e^{-e^t x+st}dt, \tag{1–2}$$

其中 $\Gamma(\cdot)$ 为 Gamma 函数. 对这个积分进行数值积分求解, 从而可以得到一组 SOE. 这种利用数值积分得到 SOG 或者 SOE 的方法往往具有高精度的特点. 一般核函数的基于逆 Laplace 变换和数值积分的 SOG 逼近方法在 Dietrich 和 Hackbusch 的文献中有详细介绍[14].

另一种实现 SOG 逼近的主流方法是最小二乘法. 然而直接应用最小二乘法往往会出现病态矩阵而导致误差较大的情况. 利用分部差分分解和改进的 Gram-Schmidt 方法可以解决最小二乘法出现的误差问题，从而显著提高了精度[21]. Greengard 等人[3] 为径向对称核函数的 SOG 开发了一种黑箱方法. 该方法分配了位于正实轴上的一组对数等距点 t_j，然后通过自适应二分法构造了一组采样点 x_i. 从而拟合矩阵 A 由 $A_{ij} = e^{-t_j x_i}$ 给出，而右端的向量 b 由 $b_i = f(x_i)$ 给出. 在通过求解最小二乘问题得到权值后，引入了模型约简 (Model Reduction,MR)[22] 技术中的平方根方法来减少指数项数量，并实现一个接近最优的 SOG 近似.

使用高斯函数逼近目标函数实际上是一个高度非线性的问题. 设计 t_j 的近似方案，使它有一个正数下界（即带宽的下界与高斯项数量无关）是非平凡且极其有意义的. 因为在快速高斯变换 (Fast Gauss Transform,FGT)[23-24] 计算由 SOG 近似的核求和问题的应用中，一个小的带宽下界会严重降低算法的性能. 由于这一缺陷, FGT 在某些特定带宽的高斯函数核求和问题中无法得到广泛应用. 如果能够设计出可以解决带宽控制的 SOG 算法，将可以大大提升 FGT 的性能. 在这项工作中，提出了一种新的核无关的 SOG 方法，它保持了高精度和可调的带宽下界. 通过一个变量替换，用 de la Vall´ee-Poussin(VP) 和[25-26] 来构造核的高斯近

似. 变量替换引入了一个参数 n_c, 它允许调整高斯函数的最小带宽. 此外, MR 技术可以进一步用于减少在指定的精度水平下的高斯项数, 从而实现一个优化过后的 SOG 近似.

SOE 近似与 SOG 近似类似. 设计一些附加约束的 SOE 近似是强非线性和高度非平凡的问题, 在文献中是一个广泛研究的课题[11-16, 18, 21, 27]. 在得到高精度 SOG 展开的基础上, 可以通过变量代换来得到目标函数的 SOE 展开. 同样地, 这样得到的 SOE 展开也具有高精度, 带宽可控等优良性质. 高精度的近似在科学计算中有极其广泛的应用, 诸如时间卷积积分的计算与时间卷积积分方程的求解. 卷积积分的计算与卷积积分方程的数值求解原本广泛使用基于 Laplace 变换的 Runge-Kutta 方法. 该种方法的核心思想即是利用 Laplace 变换把卷积积分的核函数转化为 SOE 的形式. 因而利用高精度的 SOE 来代替 Laplace 变换去求解时间卷积积分问题是理论可行的. 本文则对这种理论可能进行了理论和实验两个方面的尝试. 从误差分析与数值算例说明这种原创算法的优良性质.

本文所使用的利用 VP 和进行 SOG(SOE) 估计, 之后使用 MR 技术进行约简的方法系首创. 并且研究的结果相较于历史结果从某种程度上更好解决了历史上 SOG(SOE) 估计问题中所遇到的带宽可控, 系数解析的难题. 这种难题的突破使得 VPMR 算法所得到的结果可以与许多其他的算法相耦合, 从而拓宽了该算法的应用领域.

本文的其余部分组织如下. 第二章详细介绍 SOG 估计和 SOE 估计的算法, 并将其与其它历史方法进行比较. 第三章介绍 SOE 估计耦合下的时间卷积积分计算, 并对其进行详细的误差分析. 第四章介绍 SOE 估计耦合下的卷积积分方程的求解, 通过一个简单的算例说明耦合的优点. 第五章介绍 SOG 估计耦合下的多重快速高斯变换, 并对耦合之后的时间复杂度与误差进行理论分析. 第六章则展示所有耦合算法的相关算例.

第二章　高斯和估计与指数和估计

在这一节中，我们首先考虑用高斯函数的线性组合逼近有限实区间上的函数，即使用 p 个高斯函数的和

$$f_p(x) = \sum_{j=1}^{p} w_j e^{-x^2/s_j^2}, \tag{2–1}$$

对目标函数 $f(x)$ 进行逼近[28]. 其中 w_j 和 s_j 分别为线性系数和带宽. 定义 $s_p = \min_j |s_j|$ 为最小带宽.

2.1　de la Vallée Poussion 和

首先考虑 VP 和. 设目标函数 $f(x)$ 为定义在实正半轴上的光滑连续函数，且在无穷远处存在有限极限. 不失一般性，假设目标函数的极限为 0，即 $\lim_{x\to\infty} f(x) = 0$. 考虑如下的变量代换：

$$x = \sqrt{-n_c \log\left(\frac{1+\cos t}{2}\right)},\ t = [0, \pi], \tag{2–2}$$

其中 $t \leftrightarrow x$ 为一个一一映射. 参数 n_c 为一个正常数，它决定了带宽的下界. 设 $\varphi(t) = f(x)$，则 $\varphi(t)$ 在 $[0, \pi]$ 上光滑连续. 而且有 $\varphi(0) = f(0)$ 和 $\varphi(\pi) = f(\infty) = 0$. 对 $\varphi(t)$ 进行偶延拓，以至于其可以看做 $(-\infty, \infty)$ 上周期为 2π 的偶函数.

函数 $\varphi(t)$ 的 VP 和定义为[29]

$$V_n[\varphi(t)] = \frac{1}{n}\sum_{\ell=n}^{2n-1} S_\ell[\varphi(t)], \tag{2–3}$$

其中

$$S_\ell[\varphi(t)] = \sum_{k=0}^{\ell} a_k \cos(kt), \tag{2–4}$$

为 $\varphi(t)$ 的 Fourier 级数的部分和, Fourier 系数 a_k 则定义为

$$a_k = \begin{cases} \dfrac{1}{\pi}\displaystyle\int_0^\pi \varphi(t)dt, & \text{for } k = 0, \\[2ex] \dfrac{2}{\pi}\displaystyle\int_0^\pi \varphi(t)\cos(kt)dt, & \text{for } k \geq 1. \end{cases} \tag{2–5}$$

VP 和，即公式(2–3)可以分解为两部分，

$$V_n[\varphi(t)] = S_n[\varphi(t)] + \sum_{\ell=1}^{n-1}\left(1 - \frac{\ell}{n}\right) a_{n+\ell} \cos\left[(n+\ell)t\right]. \tag{2–6}$$

在公式(2–6)代入逆变换 $t = \arccos(2e^{-x^2/n_c} - 1)$ 可得到

$$f_p(x) = \sum_{\ell=0}^{n} a_\ell T_\ell\left(2e^{-x^2/n_c} - 1\right) + \sum_{\ell=1}^{n-1}\left(1 - \frac{\ell}{n}\right) a_{n+\ell} T_{n+\ell}\left(2e^{-x^2/n_c} - 1\right) \tag{2–7}$$

其中 $f_p(x) = V_n[\varphi(t)]$ 为目标函数 $f(x)$ 的一个逼近. 由定理2.1可知它是一个项数为 $p = 2n$ 的 SOG 展开. 这里 $T_m(x)$ 为 m 阶 Chebyshev 多项式, 定义为

$$T_m(x) = \cos(m \arccos(x)) = \sum_{\ell=0}^{\lfloor m/2 \rfloor} (-1)^\ell \binom{m-\ell}{2\ell} x^{m-2\ell}(1-x^2)^\ell. \tag{2–8}$$

将公式(2–8)代入到公式(2–7)中并且对系数进行整理, 可以得到如下的定理2.1.

定理 2.1 定义在公式 (2–7)中的函数 $f_p(x)$ 可以写做如下 SOG 的形式,

$$f_p(x) = \sum_{j=0}^{2n-1} w_j e^{-jx^2/n_c}, \tag{2–9}$$

其中 $p = 2n$. 这里系数 w_j 为,

$$w_j = \begin{cases} a_0 + \sum\limits_{\ell=1}^{n} (-1)^\ell a_\ell + \sum\limits_{\ell=1}^{n-1} (-1)^{n+\ell} \left(1 - \dfrac{\ell}{n}\right) a_{n+\ell}, & j = 0, \\ 2^{2j} \sum\limits_{\ell=j}^{n} (-1)^{\ell-j} \dfrac{\ell}{\ell+j} \dbinom{\ell+j}{\ell-j} a_\ell + \sum\limits_{\ell=1}^{n-1} c_n^{j\ell} a_{n+\ell}, & 1 \le j \le n, \\ \sum\limits_{\ell=j-n}^{n-1} c_n^{j\ell} a_{n+l}, & j > n, \end{cases} \tag{2–10}$$

有

$$c_n^{j\ell} = (-1)^{n+\ell-j} \left(1 - \frac{\ell}{n}\right) \frac{(n+\ell)}{n+\ell+j} \binom{n+\ell+j}{n+\ell-j} 2^{2j}.$$

公式(2–9)给出了第 j 个 $(j > 0)$ 高斯项带宽为 $s_j = \sqrt{n_c/j}$ 的 SOG 展开的显式表达. 注意到整个表达式中引入的唯一近似值是在公式(2–5)中的 Fourier 系数的数值计算. 并且可以引入快速余弦变换等方法来加速系数的计算过程.

注 最小带宽为 $s_p = \sqrt{n_c/(2n-1)}$, 从而 n_c 决定了所有带宽的下界. 如果保证 $n_c \sim n$, 带宽会渐近地趋于常数.

SOE 的展开同理, 对应公式2–2的变量代换变为

$$x = -n_c \log\left(\frac{1+\cos t}{2}\right), \ t \in [0, \pi], \tag{2–11}$$

VP 和表达式变为

$$V_n[\varphi(r)] = \frac{2}{n\pi} \sum_{\ell=n}^{2n-1} \sum_{j=0}^{\ell} \alpha_j \cos(jt) \int_0^\pi \varphi(\tau) \cos(j\tau) d\tau \tag{2–12}$$

其中 $\alpha_j = 1, j \ge 1$ 且 $\alpha_0 = 1/2$. 代入后有

$$f(x) \approx \frac{1}{\pi} \int_0^\pi \phi(\tau) d\tau + \sum_{j=1}^{2n-1} a_j T_j(2e^{-x/n_c} - 1) \tag{2–13}$$

其中系数为

$$a_j = \max\left\{\frac{2}{\pi}, \frac{4n-2j}{n\pi}\right\}\int_0^{\pi} K(\tau)\cos(j\tau)d\tau \tag{2–14}$$

最后得到 SOE

$$f(x) \approx \sum_{j=0}^{2n-1} w_j e^{-jx/n_c}, \tag{2–15}$$

其中线性系数为

$$w_j = \begin{cases} 2a_0 + \sum\limits_{\ell=1}^{n}(-1)^{\ell}\dfrac{n}{2n-\ell}a_{\ell} + \sum\limits_{\ell=1}^{n-1}(-1)^{n+\ell}\dfrac{n-\ell}{2n-\ell}a_{n+\ell}, & \text{for } j=0 \\ 2^{2j}\sum\limits_{\ell=j}^{n}(-1)^{\ell-j}\dfrac{n\ell}{(\ell+j)(2n-\ell)}\binom{\ell+j}{\ell-j}a_{\ell} + \sum\limits_{\ell=1}^{n-1}c_n^{j\ell}\dfrac{n}{2n-\ell}a_{n+\ell}, & \text{for } 1\le j\le n \\ \sum\limits_{\ell=j-n}^{n-1}\dfrac{nc_n^{j\ell}}{2n-\ell}a_{n+\ell}, & \text{for } j>n \end{cases} \tag{2–16}$$

其中

$$c_n^{j\ell} = (-1)^{n+\ell-j}\left(1-\frac{\ell}{n}\right)\frac{(n+\ell)}{n+\ell+j}\binom{n+\ell+j}{n+\ell-j}2^{2j}. \tag{2–17}$$

不难证明公式(2–16)与公式(2–10)等价. 后续的误差分析与模型降阶也均以 SOG 作为基准.

2.2 误差分析

这里我们将讨论使用 VP 和 $V_n[\varphi(t)]$ 估计目标函数 $\varphi(t)$ 所产生的误差. 假设 $\varphi(t)$ 在 $[-\pi,0)\cup(0,\pi]$ 内至少二阶可微, 将会分别讨论 $t=0$ 处不可微和至少一阶可微两种情况下的误差.

当 $\varphi(t)$ 在 $t=0$ 处不可微时, 已有文献证明 VP 和 $V_n[\varphi(t)]$ 在 $\mathbb{R}$ 上一致收敛于 $\varphi(t)$, 但是在 $t=0$ 一点处的收敛速率明显更低. 对于具体的收敛阶, 根据 Boyer 和 Goh 的文献[30], 有如下结果:

$$V_n[\varphi(0)] - \varphi(0) = -\frac{\ln 2}{n\pi}\sqrt{n_c}f'(0) + O\left(n^{-3/2}\right). \tag{2–18}$$

当 $\varphi(t)$ 在 $t=0$ 处一阶可微时, 不妨设 $f'(0)=0$ 因为许多径向函数都满足 $f'(0)=0$ 这一性质, 例如逆二次核和 Matérn 核 ($\nu\ge 1$). 在这种情况下在公式(2–18)中的右侧第一项为 0, 误差的阶数会更高. 实际上, 定理2.2表明, 误差将会达到 $O\left(n^{-2}\right)$, 而不是 $O(n^{-3/2})$.

定理 2.2 假设 $V_n[\varphi(t)]$ 为 $\varphi(t)$ 的 n 阶 VP 和估计, 且 $\varphi(t)$ 在 $[0,\pi]$ 上二阶可微, 周期为 2π. 若 $f'(0)=0$, 则有

$$V_n[\varphi(0)] - \varphi(0) = O\left(n^{-2}\right), \tag{2–19}$$

$$V_n[\varphi(t)] - \varphi(t) = o\left(n^{-2}\right), \quad t\ne 0. \tag{2–20}$$

证明 $S_n[\varphi(t)]$ 和 $V_n[\varphi(t)]$ 分别表示 n 阶 Fourier 部分和和 n 阶 $\varphi(t)$ 和估计. 引入 Fejér 部分和为

$$\sigma_n[\varphi(t)] = \frac{1}{n+1}\sum_{\ell=0}^{n} S_{\ell}[\varphi(t)] \tag{2–21}$$

从而 VP 和估计的误差可以表示为

$$V_n[\varphi(t)] - \varphi(t) = 2\sigma_{2n}[\varphi(t)] - \sigma_n[\varphi(t)] - \varphi(t). \tag{2–22}$$

由 Fejér 部分和的性质知[26]

$$\sigma_n[\varphi(t)] - \varphi(t) = \frac{1}{n\pi}\int_{-\pi}^{\pi}\left[\varphi(t+\xi) - \varphi(t)\right]\frac{\sin^2\frac{n\xi}{2}}{2\sin^2\frac{\xi}{2}}d\xi. \tag{2–23}$$

把公式(2–23)代入(2–22)中有

$$V_n[\varphi(t)] - \varphi(t) = \frac{1}{n\pi}\int_{-\pi}^{\pi}\left[\varphi(t+\xi) - \varphi(t)\right]\frac{\cos n\xi - \cos 2n\xi}{4\sin^2\frac{\xi}{2}}d\xi. \tag{2–24}$$

考虑 $t = 0$ 的情形，公式(2–24)可以展开为两部分，I_1 和 I_2，即

$$I_1 = \frac{1}{n\pi}\int_0^{\pi}(\varphi(\xi) - \varphi(0))\frac{\cos n\xi - \cos 2n\xi}{4\sin^2\frac{\xi}{2}}d\xi \tag{2–25}$$

与

$$I_2 = \frac{1}{n\pi}\int_{-\pi}^{0}(\varphi(\xi) - \varphi(0))\frac{\cos n\xi - \cos 2n\xi}{4\sin^2\frac{\xi}{2}}d\xi. \tag{2–26}$$

首先考虑 I_1，将其分解为 $I_1 = I_{11} + I_{12}$，即

$$I_{11} = \frac{1}{n\pi}\int_0^{\pi}(\varphi(\xi) - \varphi(0))(\cos n\xi - \cos 2n\xi)\left(\frac{1}{4\sin^2\frac{\xi}{2}} - \frac{1}{\xi^2}\right)d\xi \tag{2–27}$$

与

$$I_{12} = \frac{1}{n\pi}\int_0^{\pi}(\varphi(\xi) - \varphi(0))(\cos n\xi - \cos 2n\xi)\frac{1}{\xi^2}d\xi. \tag{2–28}$$

在公式(2–27)中，$1/4\sin^2(\xi/2) - 1/\xi^2$ 在 $[0,\pi]$ 内取值范围为 $(0.08, 0.15)$，而且被积函数的每一部分都在 $[0,\pi]$ 内连续，故可由第一积分中值定理，存在一个正实数 M_1 使得

$$I_{11} = \frac{M_1}{n\pi}\int_0^{\pi}(\varphi(\xi) - \varphi(0))(\cos n\xi - \cos 2n\xi)d\xi. \tag{2–29}$$

由分部积分法，可得

$$\begin{aligned}
I_{11} =& \frac{M_1}{n\pi}\left[\left(\frac{1}{n}\sin n\xi - \frac{1}{2n}\sin 2n\xi\right)\frac{\varphi(\xi)-\varphi(0)}{\xi}\Bigg|_{\xi=0}^{\pi}\right] - \\
& \int_0^{\pi}\frac{d\frac{\varphi(\xi)-\varphi(0)}{\xi}}{d\xi}\left(\frac{1}{n}\sin n\xi - \frac{1}{2n}\sin 2n\xi\right)d\xi \\
=& \frac{M_1}{n^2\pi}\int_0^{\pi}\frac{d\frac{\varphi(\xi)-\varphi(0)}{\xi}}{d\xi}\left(\sin n\xi - \frac{1}{2}\sin 2n\xi\right)d\xi \\
=& O\left(\frac{1}{n^2}\right),
\end{aligned} \tag{2–30}$$

其中最后两步使用了一个事实，$\varphi'(0) = 0$ 且 $\varphi''(0)$ 存在．对于 I_{12}，也可以将其分为两部分，$I_{12} = I_{121} + I_{122}$，即

$$I_{121} = \frac{1}{n\pi}\int_0^{\frac{1}{n}}(\varphi(\xi) - \varphi(0))(\cos n\xi - \cos 2n\xi)\frac{1}{\xi^2}d\xi, \tag{2–31}$$

与

$$I_{122} = \frac{1}{n\pi}\int_{\frac{1}{n}}^{\pi}(\varphi(\xi)-\varphi(0))(\cos n\xi - \cos 2n\xi)\frac{1}{\xi^2}d\xi. \tag{2–32}$$

注意到 $\cos n\xi - \cos 2n\xi$ 在 $[0, 1/n]$ 上非负而且单调递增. 由于 $f''(t)$ 存在, 剩余部分 I_{121} 可积且有界, 对其应用第二积分中值定理, 即则存在一个正实数 $M_2 \le 1/n$ 使得

$$I_{121} = \frac{1}{n\pi}(\cos 1 - \cos 2)\int_{M_2}^{\frac{1}{n}}\frac{\varphi(\xi)-\varphi(0)}{\xi^2}d\xi = O\left(\frac{1}{n^2}\right). \tag{2–33}$$

注意到 $[\varphi(\xi)-\varphi(0)]/\xi^2$ 在 $[1/n, \pi]$ 上有界, 存在一个正实数 M_3 使得

$$|I_{122}| \le \frac{M_3}{n\pi}\left|\int_{\frac{1}{n}}^{\pi}(\cos n\xi - \cos 2n\xi)d\xi\right| = \frac{M_3(2\sin 1 - \sin 2)}{2n^2\pi} = O\left(\frac{1}{n^2}\right). \tag{2–34}$$

最后把这些估计式加和, 便有

$$I_1 = I_{11} + I_{12} = I_{11} + I_{121} + I_{122} = O\left(\frac{1}{n^2}\right). \tag{2–35}$$

同理有

$$I_2 = O\left(\frac{1}{n^2}\right). \tag{2–36}$$

将 I_1 和 I_2 的估计式代入公式(2–24)中便得到公式(2–19)所描述的估计式. 对于 $t \ne 0$ 的情形采用相同的方法, 这里予以省略. □

2.3 模型降阶方法

在本节中, 考虑我们考虑减少 SOG 估计中的高斯项数的方法. 我们将平方根方法 (Square Method) 应用于模型降阶[22, 31] 中, 以实现一个接近最优的近似. 目的为找到一个 q 项的 SOG, 使得

$$\sum_{j=1}^{2n-1} w_j e^{-jx^2/n_c} \approx \sum_{\ell=1}^{q}\widetilde{w}_\ell e^{-x^2/s_\ell^2}, \tag{2–37}$$

其中 $q < 2n-1$ 且有 $s_q = \min_\ell |s_\ell| \approx \sqrt{n_c/(2n-1)}$. 这里要注意 $j = 0$ 项为常数项已忽略.

设 $y = x^2$, 代入到公式2–38的左端, 并对其应用 Laplace 变换, 即

$$L\left[\sum_{j=1}^{2n-1} w_j e^{-jy/n_c}\right] = \sum_{j=1}^{2n-1}\frac{w_j}{z + j/n_c}. \tag{2–38}$$

公式2–38右端可以用一个线性系统来表示,

$$\boldsymbol{c}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b} = \sum_{j=1}^{2n-1}\frac{w_j}{z + j/n_c}, \tag{2–39}$$

其中 $\boldsymbol{A}$ 为一个对角矩阵, $\boldsymbol{b}$ 和 $\boldsymbol{c}$ 分别为一个列向量和一个行向量, 称为系数矩阵, 定义为

$$\begin{aligned}
\boldsymbol{A} &= -\mathrm{diag}\left\{\frac{1}{n_c}, \frac{2}{n_c}, \cdots, \frac{(2n-1)}{n_c}\right\},\\
\boldsymbol{b} &= \left(\sqrt{|w_1|},\ \sqrt{|w_2|},\ \cdots,\ \sqrt{|w_{2n-1}|}\right)^T,\\
\boldsymbol{c} &= \left(\mathrm{sign}(w_1)\sqrt{|w_1|},\ \mathrm{sign}(w_2)\sqrt{|w_2|},\ \cdots,\ \mathrm{sign}(w_{2n-1})\sqrt{|w_{2n-1}|}\right).
\end{aligned} \tag{2–40}$$

利用这种表示形式，可以使用平方根法[31] 来减少其项数和系数. 这种技术可以减少公式2–38中有理函数的数量，从而减少 SOG 中的高斯项数. 这种方法的第一步是求解两个 Lyapunov 方程，求解出 $\boldsymbol{P}$ 与 $\boldsymbol{Q}$,

$$\boldsymbol{AP}+\boldsymbol{PA}^*+\boldsymbol{bb}^*=0,\quad \boldsymbol{A}^*\boldsymbol{Q}+\boldsymbol{QA}+\boldsymbol{c}^*\boldsymbol{c}=0, \tag{2–41}$$

其中 $*$ 表示共轭转置. 第二步是利用平方根法找到一个平衡变换矩阵 $\boldsymbol{X}$，从而可以计算出乘积 $\boldsymbol{PQ}$ 的奇异值. 这样约简后的系统的系数矩阵为 $\widetilde{\boldsymbol{A}}^{q\times q}$, $\tilde{\boldsymbol{b}}^{q\times 1}$ 和 $\tilde{\boldsymbol{c}}^{1\times q}$. 它们定义为 $\boldsymbol{XAX}^{-1}$, $\boldsymbol{Xb}$, 和 $\boldsymbol{cX}^{-1}$ 的分别前 $q\times q$, $q\times 1$, $1\times q$ 子块. 这些子块满足

$$\sup_{z=i\mathbb{R}}\left|\tilde{\boldsymbol{c}}(z\widetilde{\boldsymbol{I}}-\widetilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{b}}-\boldsymbol{c}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{b}\right|\le\delta, \tag{2–42}$$

其中 δ 为一个特定常数，与 $\boldsymbol{PQ}$ 的奇异值有关. 这样再利用特征分解和逆拉普拉斯变换，实现了 MR 过程，得到了优化后的 SOG 近似，即公式(2–37).

下面给出本文 MR 技术的详细算法. 如算法2–1所示，这是一种最基础的基于平方根法的模型降阶技术. 实际实验表明，其它 MR 技术的方法也可以应用，因为模型降阶方法的实质是将每一项看做信号进行约简冗余. 这里举出的是给定误差进行的模型降阶，也可以给定降阶项数进行模型降阶，这需要对第 7 步进行些许改动. 即直接选择约简之后的项数，在最后计算约简之后的误差. 两种不同约简方式的选择取决于实际需要.

算法 2–1 基于平方根法的模型降阶技术

输入: 线性系数 $\{w_r\}_{r=0}^{2n-1}$, 带宽参数 n_c 与误差 ϵ.

输出: 降阶后的线性系数 $\{m_l\}_{l=0}^{q}$ 与带宽系数 $\{s_l\}_{l=0}^{q}$.

1 生成一个对角矩阵 $A=-diag(1/n_c,2/n_c,...,(2n-1)/n_c)$. 生成一个列向量 $B=(\sqrt{|w_1|},\sqrt{|w_2|},...\sqrt{|w_{2n-1}|})^T$. 生成一个行向量 $C=(sgn(w_1)\sqrt{|w_1|},sgn(w_2)\sqrt{|w_2|},...,sgn(w_{2n-1})\sqrt{|w_{2n-1}|})$;

2 求解 Lyapunov 方程 $AP+PA^T=-BB^T$ 与 $AQ+QA^T=-CC^T$;

3 计算 P 的 Cholesky 因子 S, Q 的 Cholesky 因子 L，即求解方程 $P=SS^T, Q+LL^T$;

4 计算 S^TL 的奇异值分解 $S^TL=U\Sigma V^T$，其中 $\Sigma=diag(\sigma_1,\sigma_2,...\sigma_{2n-1})$;

5 计算 $\widetilde{T}=SU\Sigma^{-\frac{1}{2}}$;

6 生成矩阵 $\widetilde{A}=\widetilde{T}^{-1}A\widetilde{T}$. 生成列向量;

7 $\widetilde{B}=\widetilde{T}^{-1}B$. 生成行向量 $\widetilde{C}=C\widetilde{T}$;

8 求满足不等式 $2\sum_{i=q+1}^{2n-1}\sigma_i\le\epsilon$ 的 q(一般取最小值);

9 取矩阵 $\widetilde{A}$ 的前 $q\times q$ 子块作为矩阵 $\widehat{A}$ 取列向量 $\widetilde{B}$ 的前 q 行作为列向量 $\widehat{B}$ 取行向量 $\widetilde{C}$ 前 q 列作为行向量 $\widehat{C}$;

10 计算特征值分解 $\widehat{A}=X\Lambda X^{-1}$. 设 $s_l=\Lambda_{ll}, l=1,2,...,q$ 并且 $s_0=0$;

11 计算 $\dot{B}=X^{-1}\widehat{B},\dot{C}=\widehat{C}X$. 设 $m+l=\dot{B}_l\dot{C}_l(l=1,2,...,q)$ 并且 $m_0=w_0$;

我们注意到, MR 技术最初是为极点和近似而设计的，其近似的最优性由控制理论[22, 32] 中已知的结果保证. 然而，由于所有的节点都位于复平面的左半部分，允许将它直接应用于简化 SOG 近似[3]. 具体的证明论述过于复杂，这里予以省略.

整个算法由于由两部分构成，即 VP 和估计和 MR 技术，因而将其称之为 VPMR 算法.

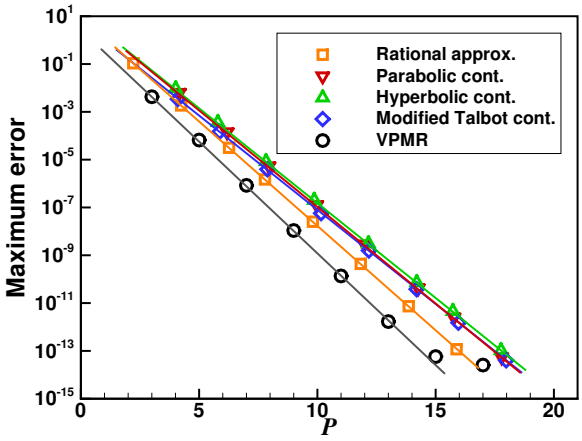

图 2–1 高斯核的 SOE 近似的最大误差

它不仅可以求解目标函数的 SOG 估计问题，也可以通过变量代换，解决目标函数的 SOE 估计问题.

注 由于 VPMR 算法需要高精度的矩阵操作，因此我们采用了多精度工具箱 (Multiple Precision Toolbox)[33] 来实现该算法. 这些工具箱在 VP 和和 MR 技术的过程中使用. VPMR 算法的计算机代码以开源的形式发布，网站为 https://github.com/ZXGao97. 使用平台为 MATLAB. 目前该代码已经完成 VPMR 算法的可视化，并且可以实时输出保存相应参数，给与用户较高的自由度. 代码经过不断的维护调试，现已可以稳定计算高频振荡核函数，分段函数等不易计算的结果. 本文中所有数值算例均使用了该代码.

2.4 一个实例：高斯函数的展开

本节中我们研究了 VPMR 算法在高斯核函数的 SOE 近似中的性能，这是在许多应用中经常需要的[27, 34]. 高斯函数核 $e^{-x^2/4\delta}$ 的逆拉普拉斯变换表示为

$$e^{-x^2/4\delta} = \frac{1}{2\pi i}\int_{\Gamma} e^{z}\sqrt{\frac{\pi}{z}}e^{-\sqrt{z}|x|/\sqrt{\delta}}dz, \tag{2–43}$$

其中 Γ 为复平面中从第三象限的 $-\infty$ 出发，绕过原点返回第二象限的 $-\infty$ 的任何曲线. 它主要有三种路径，包括抛物线，双曲线和修改后的 Talbot 曲线[35-36]. 所有这些路径都有一定的参数，并且需要进行优化，以达到最优的收敛速度[37-39]. 从而利用留数定理和柯西积分定理，用最佳有理逼近得到高斯核的 SOE 近似[36]. 文献中明确指出不同路径的积分会对数值求解的精度产生影响. 这里选用文献中使用的三种效果最好的路径进行实验.

我们使用前文中给出的 VPMR 算法和上面讨论的其他现有工作，对高斯核的 SOE 近似进行了比较. 取 $\delta = 1$，最大误差定义为

$$E_{\infty} = \max_{x\in(0,100]}\left|e^{-\frac{x^2}{4\delta}} - \sum_j m_j e^{-s_j x}\right| \tag{2–44}$$

它用来刻画不同算法的精确程度. 这里在 $[10^{-5}, 10^{2}]$ 取 100000 个随机节点监测误差. 我们从文献[27] 中得到了基于 Laplace 变换与最佳有理逼近的误差结果. 至于 VPMR 算法，我们

取 $n_c = \lceil n/4 \rceil$，结果如图2–1所示. 图中显示了 5 种不同的的 SOE 方法：最佳有理逼近、抛物线路径、双曲路径、修正的 Talbot 路径和 VPMR 算法，分别用橙色、红色、绿色、蓝色和黑色曲线表示. 结果表明, VPMR 算法无论是在收敛速度方面还是精度方面均更优. 所有五种方法均可以在项数超过 20 之后达到 10^{-13} 的精度. 精确计算结果[27] 发现三种路径方法的收敛阶数为 $O(6.3^{-n})$，最佳有理逼近的收敛阶数为 $O(7.5^{-n})$，而 VPMR 算法的收敛阶数为 $O(9.0^{-n})$. 由此可以证明 VPMR 算法相较于历史算法的优越性.

第三章　指数和下的卷积积分

高精度 SOE 估计的一个重要应用便是处理时间卷积积分的快速计算[40]. 考虑核函数 $f(x)$ 与光滑函数 $g(x)$ 的时间卷积积分

$$y(t) = f * g = \int_0^t f(t-\tau)g(\tau)d\tau. \tag{3–1}$$

这一类卷积积分的近似由于其在偏微分方程[41-42]，分数阶微分方程[43-45] 以及非线性 Volterra 方程[46-49] 中的较多应用而引起了广泛的关注. 历史上对于这种卷积积分的估计通常都是基于 Laplace 变换后使用 Runge-Kutta 方法完成的，即先对核函数 $f(x)$ 应用 Laplace 变换得 $F(s)$，则卷积积分可以转化为

$$y(t) = \frac{1}{2\pi i}\int_\Gamma F(\lambda)\int_0^t e^{\lambda(t-\tau)}g(\tau)d\tau d\lambda. \tag{3–2}$$

设 $u(t) = \int_0^t e^{\lambda(t-\tau)}g(\tau)d\tau$，则 u 满足一个常微分方程 $u' = \lambda u + g(t), u(0) = 0$，从而可以应用 Runge-Kutta 等求解常微分方程的数值方法. 而曲线积分部分可以采用数值积分解决. 该方法本质上是一种插值方法，利用函数 $g(t)$ 的插值点的线性组合来近似卷积. 这种方法可以处理奇异、多时间尺度和高振荡的核函数，因而受到广泛的关注[50]. Schädle 等人[51] 开发了一种改进的算法，将 N 时间步内的运算复杂度和运算存储空间均降到了 $O(N\log N)$. López-Fernández 和 Sauter[52] 引入了一个允许可变时间步长的广义卷积积分. 这些改进的算法使 Lubich 的方法适应性更强，并显著减少了存储空间. 但也有文献指出，这类算法仅适用于分段卷积核，因此不适用于波动方程[52]. 此外，它还需要核函数解析形式的 Laplace 变换，这对于机器学习和统计中经常使用的 Matérn 等函数来说是很困难的[53]. 而且这种方法的误差估计也会很困难.

Lubich 的方法相当于对核函数进行拉普拉斯变换，从而积分曲线上的每一个离散点代表一个指数项. 因而可以考虑利用 VPMR 算法生成的 SOE 近似代替离散曲线积分而生成的 SOE，然后对每个指数项执行 Runge-Kutta 方法. 这个想法的优点有如下五个方面. 第一，我们可以引入比 Laplace 变换之外的更有效的 SOE 方法，以更好地去逼近一些核函数. 其次，可以利用 MR 技术来减少指数项的数量，从而大幅节省计算成本. 第三，采用 VPMR 算法的耦合可以使得求解过程在保证精度的基础上更加简便，运算复杂度更低. 第四，这种方法相较于 Laplace 变换更容易做出严格精确的误差估计. 最后，部分复杂核函数难以求出解析的 Laplace 变换，但可以应用 VPMR 算法. 面对奇异核函数的情况时，根据 Lubich[54-55] 的思想，对奇异部分进行分割，利用多项式插值对奇异部分局部逼近. 本章中我们还对产生的误差进行了详细的分析. 以证明耦合算法的高精度.

3.1　卷积积分的快速计算

考虑公式(3–1)中的 $y(t)$ 的估计. 首先假设 $f(\tau)$ 是非奇异的，利用 VPMR 算法有如下 SOE 逼近，

$$f(\tau) \approx f_{\mathrm{es}}(\tau) = \sum_{\ell=1}^{P} m_\ell e^{-s_\ell \tau}, \quad \tau \in [0,t] \tag{3–3}$$

其中 $m_\ell, s_\ell \in \mathbb{C}$ 且 $\mathrm{Re}(s_\ell) \geq 0$，有误差 $\|f(\tau) - f_{\mathrm{es}}(\tau)\|_\infty < \varepsilon$，其中 $0 < \varepsilon \ll 1$.

把得到的 SOE 逼近代入卷积积分中则有

$$y(t) \approx \int_0^t f_{\mathrm{es}}(t-\tau)g(\tau)d\tau = \sum_{\ell=1}^{P} m_\ell Y_\ell(t), \tag{3–4}$$

其中

$$Y_\ell(t) = \int_0^t e^{-s_\ell(t-\tau)}g(\tau)d\tau. \tag{3–5}$$

公式3–4的和式中的每一项都可以看做如下常微分方程在 $\tau = t$ 的解

$$Y_\ell'(\tau) = -s_\ell Y_\ell(\tau) + g(\tau) \ \text{ with } Y(0) = 0. \tag{3–6}$$

公式(3–6)可以使用时间步长 h 的 Runge-Kutta 方法有效求解. 若时间尺度为 $t = Nh$，则运算复杂度可以记为 $O(N)$.

具体来说，隐式 Runge-Kutta 方法可以写为

$$Y_\ell^{n+1} = Y_\ell^n + h\sum_{i=1}^{q} b_i K_i, \tag{3–7}$$

其中

$$K_i = -s_\ell\left(Y_\ell^n + h\sum_{j=1}^{q} a_{ij}K_j\right) + g(t_n + c_i h), \ \ i = 1, 2, \cdots, q, \tag{3–8}$$

这里 a_{ij}, b_i 和 c_i 为参数, Y_ℓ^n 为 $Y_\ell(nh)$ 的近似解. 本文参照文献[56] 的格式对 Runge-Kutta 方法进行刻画. 一种 Runge-Kutta 方法称为 p 阶 (Order), q 级 (Stage), S 阶级 (Order Stage) 是指其局部截断误差为 $O(h^{p+1})$，共有 q 个子步，每一子步误差为 $O(h^{S+1})$. 按照 Butcher 表格，可以设参数矩阵为 $\boldsymbol{A} = (a_{ij})_{q\times q}$，$\boldsymbol{\beta}^T = (b_1, \cdots, b_q)$ 与 $\boldsymbol{\zeta} = (c_1, \cdots, c_q)$. 由公式(3–7)规定的 Runge-Kutta 方法的稳定性函数定义为

$$r(z) = 1 + z\boldsymbol{\beta}^T(\boldsymbol{I} - z\boldsymbol{A})^{-1}\boldsymbol{E} = \frac{\det(\boldsymbol{I} - z\boldsymbol{A} + z\boldsymbol{E}\boldsymbol{\beta}^T)}{\det(\boldsymbol{I} - z\boldsymbol{A})}. \tag{3–9}$$

其中 $\boldsymbol{E}$ 为 q 维单位列向量. 本文这里选择满足 $b_j = a_{qj}, j = 1, \cdots, q, c_q = 1$, $\boldsymbol{A}$ 的所有特征值有正实部的隐式 Runge-Kutta 方法. 服从上述条件的 Runge-Kutta 方法有 Lobatto IIIC 方法等. 下一节将对这种方法进行详细介绍. 这一性质说明该 Runge-Kutta 方法具有 L-稳定性，即

$$|r(z)| \leq 1 \text{ for } \mathrm{Re}(z) \leq 0 \text{ and } r(\infty) = 0. \tag{3–10}$$

考虑到常微分方程(3–6)存在刚性较强的情况, L-稳定性是至关重要的一个条件. 否则会导致求解过程中明显的误差积累. 并且这种格式有一个非常好的性质，即可以把相邻时间步的数值解写成递推格式，从而减少运算复杂度与运算空间. 将公式(3–7)代入公式(3–6)中，可以写出由 Runge-Kutta 方法得到数值解的递推表达式

$$Y_\ell^{n+1} = h\sum_{j=0}^{n} \boldsymbol{v}_{n-j}(z_\ell)\boldsymbol{g}_j = r(z_\ell)Y_\ell^n + h\boldsymbol{\psi}_\ell\boldsymbol{g}_n, \tag{3–11}$$

其中 $\boldsymbol{\psi}_\ell = \boldsymbol{\beta}^T(\boldsymbol{I} - z_\ell\boldsymbol{A})^{-1}$，$z_\ell = -s_\ell h$. 这里 $\boldsymbol{v}_n(z)$ 和 $\boldsymbol{g}_j$ 定义为

$$\boldsymbol{v}_n(z) = r(z)^n\boldsymbol{\beta}^T(\boldsymbol{I} - z\boldsymbol{A})^{-1}, \quad \boldsymbol{g}_j = (g(t_j + c_1 h), \cdots, g(t_j + c_q h))^T. \tag{3–12}$$

从而卷积积分公式(3–1)可以改写为

$$y(t) \approx \sum_{\ell=1}^{P} m_\ell \left[r(z_\ell) Y_\ell^{N-1} + h\boldsymbol{\psi}_\ell \boldsymbol{g}_{N-1} \right], \tag{3–13}$$

其中 Y_ℓ^{N-1} 由公式(3–11)求得. 由于 $\boldsymbol{\psi}_\ell$ 和 $\boldsymbol{g}_{N-1}$ 均为长度为 q 的向量, 每一步的时间复杂度均为 $O(P)$. 而由于 P 认为是一个预设参量, 可以认为每一步时间复杂度为 $O(1)$. 并且递推的形式使得历史数据可以保存, 求解全域时间点上的值的时间消耗会显著更低.

3.2 Lobatto IIIC 方法

本节着重介绍一种满足 $b_j = a_{qj}, j = 1, \cdots, q, c_q = 1$, 而且 $\boldsymbol{A}$ 的所有特征值有正实部的隐式 Runge-Kutta 方法, 称为 Lobatto IIIC 方法. 本文后续所有 Runge-Kutta 方法均选用不同阶数的 Lobatto IIIC 方法.

考虑一个黎曼积分

$$\int_{t_n}^{t_n+h_n} f(t)dt \tag{3–14}$$

其中 f 为一个连续函数, 则该积分等价于求解在 $t = t_n + h_n$ 处的初值问题

$$\frac{d}{dt}y = f(t),\ y(t_n) = 0 \tag{3–15}$$

由于 $y(t_n + h_n) = \int_{t_n}^{t_n+h_n} f(t)dt$, 公式(3–14)的积分可以用标准数值积分公式近似

$$\int_{t_n}^{t_n+h_n} f(t)dt \approx h_n \sum_{i=1}^{s} b_i f(t_n + c_i h_n) \tag{3–16}$$

其中 s 个节点参数为 $c_1, \cdots, c_s$ 与 s 个权值参数为 $b_1, \cdots, b_s$. Lobatto 积分公式, 有时在文献中也称为 Gauss-Lobatto 积分公式由一组满足以下条件的节点参数和权值参数给出. s 个节点参数为 s 阶多项式 $\frac{d^{s-2}}{dt^{s-2}}(t^{s-1}(1-t)^{s-1})$ 的根. 这些节点参数满足 $c_1 = 0 < c_1 < \cdots < c_s = 1$. 权值参数则与节点参数满足条件 $B(2s-2)$ 关系, 其中条件 $B(p)$ 定义为

$$B(p):\ \sum_{j=1}^{s} b_j c_j^{k-1} = \frac{1}{k},\ k = 1, \cdots, p \tag{3–17}$$

由条件 $B(p)$ 的性质可以求出权值参数的显式表达

$$b_j = \frac{1}{s(s-1)P_{s-1}(2c_j-1)^2} > 0,\ j = 1, \cdots, s \tag{3–18}$$

其中 $P_k(x) = \frac{1}{k!2^k}\frac{d^k}{dx^k}((x^2-1)^k)$ 为 k 阶的勒让德多项式. 由节点参数和权值参数的显式表达式可知它们有对称性质

$$b_{s+1-j} = b_j,\ c_{s+1-j} = 1 - c_j,\ j = 1, \cdots, s \tag{3–19}$$

将这里得到的权值参数与节点参数和上一节提到的 Runge-Kutta 方法的参数矩阵 $\boldsymbol{\beta}^T$ 和 $\boldsymbol{\xi}$ 相对应, 则得到了一个 Lobatto IIIC 方法的必要条件. 余下两个必要条件分别为条件 $C(s-1)$ 与条件 $D(s-1)$. 其中两种条件分别定义为

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k},\ i = 1, \cdots, s,\ k = 1, \cdots, q \tag{3–20}$$

$$\sum_{i=1}^{s} b_i c_i^{k-1} a_i j = \frac{b_j}{k}(1-c_j^k),\ j=1,\cdots,s,\ k=1,\cdots,r \tag{3–21}$$

其中两种条件中的 a_{ij} 即为 Runge-Kutta 方法参数矩阵 $\boldsymbol{A}$ 中的元素. 对于 Runge-Kutta 方法的误差与参数矩阵的关系, 在文献[57-58] 中提到了如下定理.

定理 3.1 一个参数矩阵满足条件 $B(p), C(q), D(r)$ 的 Runge-Kutta 方法, 其阶数为 $\min(p, 2q+2, q+r+1)$.

从而 Lobatto IIIC 的阶数为 $2s-2$. 一些低阶的 Lobatto IIIC 的参数矩阵为

$$s=2:\ \boldsymbol{A}=\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},\ \boldsymbol{\beta}^T=(\frac{1}{2},\frac{1}{2}),\ \boldsymbol{\xi}=(0,1) \tag{3–22}$$

$$s=3:\ \boldsymbol{A}=\begin{bmatrix} \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix},\ \boldsymbol{\beta}^T=(\frac{1}{6},\frac{2}{3},\frac{1}{6}),\ \boldsymbol{\xi}=(0,\frac{1}{2},1) \tag{3–23}$$

$$s=4:\ \boldsymbol{A}=\begin{bmatrix} \frac{1}{12} & -\frac{\sqrt{5}}{12} & \frac{\sqrt{5}}{12} & -\frac{1}{12} \\ \frac{1}{12} & \frac{1}{4} & \frac{10-7\sqrt{5}}{60} & \frac{\sqrt{5}}{60} \\ \frac{1}{12} & \frac{10+7\sqrt{5}}{60} & \frac{1}{4} & -\frac{\sqrt{5}}{60} \\ \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{bmatrix},\ \boldsymbol{\beta}^T=(\frac{1}{12},\frac{5}{12},\frac{5}{12},\frac{1}{12}),$$
$$\boldsymbol{\xi}=(0,\frac{1}{2}-\frac{\sqrt{5}}{10},\frac{1}{2}+\frac{\sqrt{5}}{10},1) \tag{3–24}$$

3.3 奇异情况

当核函数在原点奇异 (或者近似奇异) 的时候, 考虑将卷积积分分为两部分以移除奇点. 对于给定的 $t_0 \ll 1$ 与 $T=t-t_0$, 卷积积分可以拆分为

$$y(t)=\int_0^{t_0} f(\tau)g(t-\tau)d\tau+\int_0^{T} f(t-\tau)g(\tau)d\tau := I_1+I_2. \tag{3–25}$$

这里注意到 I_2 部分的核函数不再奇异, 因而可以用上一小节提到的方法予以求解. 至于 I_1 的估计, 考虑对 $g(\tau)$ 进行多项式插值, 并且对 $f(\tau)$ 使用广义 Taylor 展开, 即

$$f(\tau)=a_0\tau^{-\alpha}+a_1\tau+a_2\tau^2+\ldots \tag{3–26}$$

其中 $a_0\tau^{-\alpha}$ 为奇异阶数的决定项. 对于弱奇异核或近似奇异核, 围绕奇点进行一些先验渐近分析是很重要的. 利用这种方法, 奇异部分的贡献被简化为一个多项式和插值函数的卷积积分. 取 $t_0=O(h)$ 并设 $G(\tau)$ 为 $g(\tau)$ 的多项式插值, 有

$$I_1 \approx \int_0^{t_0}\left[a_0\tau^{-\alpha}+a_1\tau+a_2\tau^2+\cdots\right]G(t-\tau)d\tau, \tag{3–27}$$

从而可以显式计算. 尽管 I_1 部分的运算消耗会很大, 但是由于 t_0 足够小, 其代价足以接受. 整个算法过程总结在算法3–1中.

注意到在整个计算过程中, 除了第 5,6,9,10 步, 均可以预计算. 并且 I_1 部分的计算与 t 无关, 仅计算了一次. 因而如果设时间步数为 N, 总运算复杂度为 $O(NP)$. 由于 SOE 的项数 P 可以看做一个预设参数, 并且其数量级小于时间步数, 因此可以认为该算法服从线性运算复杂度.

算法 3–1 利用 SOE 的卷积积分快速计算

输入: 时间 t，核函数 $f(\tau)$ 与卷积函数 $g(\tau)$

输出: 公式(3–1)所给出的 $y(t)$

```
选择合适的时间步长 h 与 Runge-Kutta 方法;
利用公式(3–12)预计算 v_n(z);
if f 在原点不奇异 then
    求 f 在 [0,t] 上的 SOE;
    计算 {g_j}_{j=0}^N;
    利用公式(3–13)估计 y(t);
else
    对于选择后的 t_0，将 y 分为 I_1 与 I_2;
    由公式(3–27)显式计算 I_1;
    利用 4-6 步估计 I_2;
    将 I_1 与 I_2 加和;
```

3.4 误差估计

本小节将对耦合 VPMR 算法的时间卷积积分的快速计算进行误差分析. 这里讨论更为复杂的情况，即假设核函数 $f(\tau)$ 在原点奇异. 非奇异情况可以看做为此情况的一个特例. 由公式(3–25)给出的分拆方法，可以将数值解 $y_h(t)$ 写做

$$y_h(t) = I_1^h + I_2^h, \tag{3–28}$$

其中 I_1^h 与 I_2^h 分别为 I_1 和 I_2 的数值解.

首先考虑 I_1^h 部分的估计. 假设 $g(t-\tau) \in C^\gamma([0,t_0]), \gamma \in \mathbb{Z}$，且 $G(t-\tau)$ 为 $g(t-\tau)$ 的 L 阶插值估计 $(L \le \gamma)$. 显然插值误差可以写做

$$|G(t-\tau) - g(t-\tau)| \le C_0 \|g^{(L)}\|_\infty h^L, \ \forall \tau \in [0,t_0]. \tag{3–29}$$

如果在公式(3–26)中的广义 Taylor 展开取 M 阶截断 $f(\tau) \approx f_M(\tau)$，则对于 I_1 部分的误差估计为,

$$\begin{aligned}
|I_1 - I_1^h| &= \left| \int_0^{t_0} \left(f(\tau) - f_M(\tau)\right) g(t-\tau) d\tau + \int_0^{t_0} f_M(\tau) \left(g(t-\tau) - G(t-\tau)\right) d\tau \right| \\
&\le \int_0^{t_0} \left|f(\tau) - f_M(\tau)\right| |g(t-\tau)| d\tau + \int_0^{t_0} \left|f_M(\tau)\right| |g(t-\tau) - G(t-\tau)| \, d\tau \\
&\le C_1 t_0^{M+1} \int_0^{t_0} |g(t-\tau)| d\tau + C_0 \|g^{(L)}\|_\infty h^L \int_0^{t_0} \left|f_M(\tau)\right| d\tau \\
&\le C_1 (n_0 h)^{M+1} \|g\|_{L^1} + C_0 C_{f_M,t_0} \|g^{(L)}\|_\infty h^L
\end{aligned} \tag{3–30}$$

其中 $C_{f_M,t_0} = \int_0^{t_0} |f_M(\tau)| d\tau$ 有界，因为公式(3–1)的卷积是良定义的, C_0 与 C_1 为常数.

对于 I_2^h 部分，核函数 $f(t-\tau)$ 由其在 $[0,T]$ 上的 SOE 展开 $f_{\text{es}}(t-\tau)$ 估计，误差阈值为

ε，误差为，

$$\begin{aligned}\left|I_2 - I_2^h\right| &= \left|\int_0^T (f(t-\tau) - f_{\mathrm{es}}(t-\tau))g(\tau)d\tau + \sum_{l=1}^{P} m_l E_{\mathrm{RK}}^l(t)\right| \\ &\le \varepsilon\|g\|_{L^1} + Pm_{\max}\left|E_{\mathrm{RK}}^{\max}(t)\right|\end{aligned} \tag{3–31}$$

其中 E_{RK}^{ℓ} 为求解公式(3–6)的第 ℓ 个常微分方程采用 Runge-Kutta 所引起的误差. 设 $m_{\max} = \max\{|m_\ell|\}_{\ell=1}^{P}$, $E_{\mathrm{RK}}^{\max}(t) = \max\left\{\left|E_{\mathrm{RK}}^{\ell}(t)\right|\right\}_{\ell=1}^{P}$. 对于 $E_{\mathrm{RK}}^{\max}(t)$ 有如下定理,

定理 3.2 假设一个 q 级的隐式的 Runge-Kutta 方法. 设 p 为 Runge-Kutta 方法的误差阶，且 $S \le p-1$，并满足公式(3–10). 设 h 为时间步长. 如果 $g(\tau) \in C^{(\gamma)}([t_0, t])$, $\gamma \ge p$ 与 $\max\limits_{\ell} |s_\ell h| \le 1$，则误差 $E_{\mathrm{RK}}^{\max}(t)$ 上界为

$$\left|E_{\mathrm{RK}}^{\max}(t)\right| \le Ch^p\left(\sum_{\ell=0}^{p-1}\|g^{(\ell)}(0)\|_\infty + \max_{0\le\tau\le t-t_0}\|g^{(p)}(\tau)\|_\infty\right), \tag{3–32}$$

其中 C 为一个常数.

这个定理的证明在文献[56] 的定理 3.2 和引理 5.2 中有详细的证明，因而本文略去. 结合公式(3–30)，公式(3–31)和公式 (3–32)有 $|y(t) - y_h(t)| = O\left(h^d + \varepsilon\right)$，其中 $d = \min\{M+1,\ L,\ p\}$. 这表明最后的误差主要分为两部分，一是 VPMR 算法所产生的 SOE 估计误差，二是 Runge-Kutta 方法所产生的误差. 后者主要由 Runge-Kutta 的格式与插值格式所决定. 在后续的实验中可以看到，在小步长是 Runge-Kutta 方法的误差为主要误差，因为整体误差随步长减小服从 h^d 收敛阶.

在有关时间卷积积分的文献中可以明显看到，大部分研究均投入在卷积积分相关的方程而非其数值求解. 因而在下一章我们将着重讨论 VPMR 算法耦合下的时间卷积积分相关的方程数值解，以及其相关的误差分析.

第四章　指数和下的卷积积分方程

在本章中，我们扩展了耦合 VPMR 算法的时间卷积积分方法来求解两种与应用问题的解有密切联系的卷积积分方程[59-60]. 如下所示，本算法提供了一种在 N 步时间内时间复杂度和存储空间均为 $O(N)$ 的方法. 在求解分数阶微分方程[43] 和非线性 Volterra 方程[46] 等卷积相关问题时，基于 SOE 的算法可以将卷积积分转化为递归表达式，并可以通过分割手段很好地处理核函数的奇异性. 本章将会基于理论与实验讨论时间卷积积分方程数值方法与 VPMR 算法耦合的优势所在.

4.1　线性方程

考虑一类有如下形式的线性时间卷积积分方程,

$$(1-\varpi)g(t)+H(t)=\int_0^t f(t-\tau)g(\tau)d\tau \tag{4–1}$$

其中 $H(\tau)$ 为一个给定的已知函数, $f(\tau)$ 为核函数, ϖ 为一个实参数, 初值条件为 $g(0)=g_0$ 已知, 目标求取函数 $g(\tau)$. 这类积分方程在科学计算领域被广泛研究, 例如, 它在求解线性抛物线和双曲方程有重要作用[52]. 当核函数取幂核函数时, 即 $f(t-\tau)=(t-\tau)^{-\alpha}, 0<\alpha<1, \varpi=1$ 情形公式(4–1)被称作 Abel 积分方程, $\varpi\neq 1$ 情形公式(4–1)被称作广义 Abel 积分方程. 简化 Abel 问题的解本质上与广义微积分理论相同[60-61].

如果核函数非奇异，则可利用 VPMR 算法得到 $f(t-\tau)$ 的 SOE 展开估计. 公式(4–1)应用 Runge-Kutta 方法 (如公式(3–13)) 后得到的离散格式为

$$(1-\varpi)g_N+H_N=\sum_{\ell=1}^{P} m_\ell\left[r(z_\ell)Y_\ell^{N-1}+h\boldsymbol{\psi}_\ell \boldsymbol{g}_{N-1}\right] \tag{4–2}$$

其中 $t=Nh, H_j=H(jh), \boldsymbol{\psi}_\ell$ 和 $\boldsymbol{g}_{N-1}$ 如前文的定义. 为方便起见，规定整数时间步上的点如 $g(t_j)$ 称为整点，其它时间上的点如 $g(t_j+ch), 0<c<1$ 称为内点. 显然在构造方程时我们目标为求出整点上的函数值，而内点上的函数值此时成为了未知数. 为了减少未知数，我们用 m 点插值，用整点上的函数值来近似内点上的函数值.

$$g(t_j+c_ih)=\sum_{\ell=1}^{m}\alpha_\ell^i g(t_{j-\ell+2}) \tag{4–3}$$

其中 α_ℓ^i 为插值参数. 然后通过在每个时间步长上求解只含有一个未知量的一次线性方程来构建一个递归格式，从而可以显式地计算出该解. 这样的构造使得每一步均只需处理一个一元一次方程而非一个庞大的线性方程组，大大减小了存储所需空间与计算复杂度，简化了方程求解器，并且无需讨论可能出现的病态矩阵等线性方程组的常见问题. 更重要的是，迭代的过程不会引发误差累积，从而使得数值方法对于大时间尺度或小时间步长的不耐受. 因为迭代过程会消去之前数值解引起的误差，从而保证误差不会累积.

如果核函数在原点处有奇异，则按照公式(3–25)的形式对其进行分裂以移去奇点. 设 $t_0=n_0h\ll 1, n_0\in\mathbb{N}^+$. 类似地当 $t<t_0$ 时，对 $f(t-\tau)$ 进行广义 Taylor 展开，并用多项式插

值估计 $g(\tau)$. 这样卷积积分方程会退化成一个简单的线性方程组. $t \geq t_0$ 时, 公式(4–1)的卷积积分方程化为

$$(1-\varpi)g(t)+H(t)=\int_0^{t_0} f(\tau)g(t-\tau)d\tau+\int_0^{T} f(t-\tau)g(\tau)d\tau. \tag{4–4}$$

在 $[0,T]$ 上对 $f(t-\tau)$ 进行 SOE 估计, 从而公式(4–4)的形式可以用非奇异的同样方法求解.

下面首先考虑奇异情况的误差估计, 非奇异的情形则可以看做奇异情况的一个特殊形式. 对于公式(4–4)右端的奇异部分, $f(\tau)$ 由 $f_M(\tau)$ 估计, 其中 $\tau^{-\alpha}$ 表达了奇异的阶数. 而 $g(t-\tau)$ 由 L 阶的多项式插值估计. 不难证明在奇异部分 $[0,t_0]$ 中的误差为 $O(h^{M+1}+h^L)$. 而对于公式(4–4)右端的非奇异部分, 核函数 $f(t-\tau)$ 是由其 P 项 SOE 逼近 $f_{\mathrm{es}}(t-\tau)$ 估计的, 误差控制在 ε 内. $g(\tau)$ 的内点是由 m 点整点插值完成估计的 (如公式(4–3)). 对于一个满足 L-稳定性的 p 阶 q 级 $S \leq p-1$ 阶级的 Runge-Kutta 方法, 有

$$\begin{aligned}(1-\varpi)g(t)+H(t)&=\sum_{\ell=1}^{P} m_\ell e^{-s_\ell t_0}\int_0^T e^{-s_\ell(T-\tau)}g(\tau)d\tau+\int_0^T f_\varepsilon(t-\tau)g(\tau)d\tau\\&\quad+\int_0^{t_0} f(\tau)g(t-\tau)d\tau\\&=\sum_{\ell=1}^{P} M_\ell Y_\ell(T)+\int_0^{t_0} f(\tau)g(t-\tau)d\tau+O(\varepsilon),\end{aligned} \tag{4–5}$$

其中 $M_\ell=m_\ell e^{-s_\ell t_0}$, $f_\varepsilon=f-f_{\mathrm{es}}$, 且 $Y_\ell(T)$ 为常微分方程 $Y_\ell'(\tau)=-s_\ell Y_\ell(\tau)+g(\tau)$ with $Y_\ell(0)=0$. 的解. 设 $T=N_T h$. 为了清晰起见, 使用公式(4–2)对公式(4–5)予以改写, 有

$$(1-\varpi)g(t)+H(t)=\sum_{\ell=1}^{P} M_\ell Y_\ell^{N_T}+\int_0^{t_0} f_M(\tau)G(t-\tau)d\tau+O(h^d+\varepsilon), \tag{4–6}$$

其中 $Y_\ell^{N_T}$ 为 $Y_\ell(T)$ 通过 Runge-Kutta 方法得到的数值解. 注意到 $c_q=1$, 且 Runge-Kutta 的稳定性方程 $r(z_\ell)$ 满足 $r(z_\ell)=e^{z_\ell}+O(h^{p+1})$[62], 因而对任意 ℓ 其均为 $O(1)$ 复杂度.

再考虑公式(4–6)的中间一项, 即 $\int_0^{t_0} f_M(\tau)G(t-\tau)d\tau=R(t)+\kappa g(t)$, 其中 $g(t)$ 的系数 κ 为 $O(h^{1-\alpha})$ 的. $g(t)$ 的解形式为

$$\begin{aligned}g(t)&=\frac{1}{1-\varpi-\kappa}\left[h\sum_{\ell=1}^{P}\sum_{j=0}^{N_T-1}\sum_{s=1}^{q} M_\ell v_{N_T-1-j}^{s\ell} g_j^s+R(t)-H(t)+O(h^d+\varepsilon)\right]\\&=\frac{1}{1-\varpi-\kappa}\left[h\sum_{j=0}^{N_T-1}\sum_{k=1}^{m}\xi_{jk}g(t_{j-k+2})+R(t)-H(t)+O(h^d+\varepsilon)\right],\end{aligned} \tag{4–7}$$

其中 $v_{N_T-1-j}^{s\ell}$ 为 $\boldsymbol{v}_{N_T-1-j}(z_\ell)$ 的第 s 个组分, $g_j^s=g(t_j+c_s h)$, 且

$$\xi_{jk}=\sum_{\ell=1}^{P}\sum_{s=1}^{q} M_\ell v_{N_T-1-j}^{s\ell}\alpha_k^s \tag{4–8}$$

为一个与 Runge-Kutta 方法和 SOE 展开有关的 $O(1)$ 系数. 由于内点的插值误差为 $O(h^L)$, $g(t)$ 的误差有上界

$$\gamma_N \leq \left|\frac{1}{1-\varpi-\kappa}\left[\sum_{\ell=1}^{N_T} h\Xi_\ell(\gamma_\ell+O(h^L))+O(h^d+\varepsilon)\right]\right|, \tag{4–9}$$

其中 $\gamma_\ell = \left|g(t_\ell) - g_\ell\right|$ 且 $\Xi_\ell = \sum_{j-k+2=\ell} \xi_{jk}$.

显然如果 $\varpi = 1$, $g(t)$ 的收敛阶为 $O(h^{d-1+\alpha} + \varepsilon h^{\alpha-1})$. 另外，如果 $\varpi \neq 1$，收敛阶为 $O(h^d + \varepsilon)$. 注意到初始值内产生的误差不会影响一段时间后的收敛速度，也不会引起误差累积，因为之前整点上数值解的误差在迭代的过程中被消去，没有传递到之后整点的数值解. 类似地，误差也由 VPMR 算法的 SOE 误差和 Runge-Kutta 方法的误差组成. 在 $\varpi = 1$ 时误差的分配更为复杂，但仍然是两部分误差的形式.

4.2 非线性 Volterra 方程

考虑另一种经典的问题，非线性 Volterra 方程，

$$u(t) = a(t) + \int_0^t f(t-\tau) g(\tau, u(\tau)) d\tau, \quad t \geq 0 \tag{4–10}$$

其中 $g(\tau, u(\tau))$ 为一个光滑的非线性函数, $a(\tau)$ 为一个已知函数，未知函数 $u(\tau)$ 的初值条件 $u(0) = u_0$ 已知. 非线性 Volterra 方程(4–10)出现在各种应用中，包括连续体力学、势位理论、电磁学等[63-65] 方面.

我们使用与处理原点奇异的时间卷积方程相同的处理方法. 即设置统一的时间步长 h，满足 $0 < t_0 \ll 1$ 的参数 $t_0 = n_0 h$. 当 $t < t_0$ 时，类似地，分别用广义 Taylor 展开和多项式插值估计 $f(t-\tau)$ 和 $g(t, u(t))$. 与之前直接求解方程的解不同，这里 $u(t)$ 在 $[0, t_0]$ 中插值点上的数值解需要由 Newton 迭代法等方程的数值求解器求解. $t \geq t_0$ 时，核函数 $f(t-\tau)$ 由其在 $[0, T]$ 上的 SOE 估计，从而解改写为已知函数与指数和和非线性函数的卷积的和. 我们仍然采用 Runge-Kutta 方法求解卷积积分方程，得到一个解 $u(t)$ 的递推形式. 在时间 $t = Nh$ 的非线性 Volterra 方程的离散格式写为

$$u(t) \approx u_N = a_N + \sum_{\ell=1}^{P} M_\ell \left[r(z_\ell) Y_\ell^{N_T-1} + h \boldsymbol{\psi}_\ell \boldsymbol{g}_{N_T-1} \right] + \int_0^{t_0} f_M(\tau) G(t-\tau, u(t-\tau)) d\tau. \tag{4–11}$$

与之前的方法一样，内点上的函数值 $u(t_j + c_i h)$ 由整点上的函数值插值估计.

下面对非线性 Volterra 方程的数值求解进行误差分析. 核函数 $f(\tau)$ 由 $[0, t_0]$ 上的 M 阶广义 Taylor 展开 $f_M(\tau)$ 与 $[0, T]$ 上的 SOE 展开 $f_{\mathrm{es}}(t-\tau)$ 估计. 其中后者的误差控制在 ε 以内. $g(t-\tau, u(t-\tau))$ 则由 $[0, t_0]$ 上的 L 阶插值 $G(t-\tau, u(t-\tau))$ 估计. 其 $[0, T]$ 上的内点函数值由整点函数值的 m 阶插值估计. 从而误差估计为

$$|u(t) - u_N| = \left| E_{t_0} + \int_0^T \left(f(t-\tau) - f_{\mathrm{es}}(t-\tau) \right) g(\tau, u(\tau)) d\tau + \sum_{\ell=1}^{P} m_\ell E_{\mathrm{RK}}^\ell(t) \right| \tag{4–12}$$

其中

$$E_{t_0} = \int_0^{t_0} \left[f(\tau) g(t-\tau, u(t-\tau)) - f_M(\tau) G(t-\tau, u(t-\tau)) \right] d\tau, \tag{4–13}$$

且 $E_{\mathrm{RK}}^{\ell}(t)$ 为求解常微分方程过程中使用 Runge-Kutta 产生的误差. E_{t_0} 有估计

$$
\begin{aligned}
|E_{t_0}| &= \left|\int_0^{t_0}(f(\tau)-f_M(\tau))g(t-\tau,u(t-\tau))+f_M(\tau)(g(t-\tau,u(t-\tau))-G(t-\tau,u(t-\tau)))d\tau\right| \\
&\le C_0 t_0^{M+1}\left|\int_0^{t_0} g(\tau,u(\tau))d\tau\right| + C_1 h^L \sum_{i=0}^{L}\|\partial_t^i\partial_u^{L-i}g(\tau,u(\tau))\|_\infty \int_0^{t_0}\left|f_M(t-\tau)\right|d\tau \\
&\le C_0(n_0h)^{M+1}\|g\|_{L_1} + C_1 C_{f_M,t_0} h^L \sum_{i=0}^{L}\|\partial_t^i\partial_u^{L-i}g(\tau,u(\tau))\|_\infty
\end{aligned}
\tag{4–14}
$$

其中 $C_{f_M,t_0}=\int_0^{t_0}\left|f_M(t-\tau)\right|d\tau$ 有界, C_0 与 C_1 为常数, 指数和那一项的误差为

$$
\left|\int_0^{T}(f(t-\tau)-f_{\mathrm{es}}(t-\tau))g(\tau,u(\tau))d\tau\right| \le \varepsilon\|g\|_{L_1}. \tag{4–15}
$$

对于 $E_{\mathrm{RK}}^{l}(t)$, 我们有如下定理对其进行刻画.

定理 4.1 对于非线性函数 $g(\tau,u(\tau))$, 假设对于任意的 $\eta<0$ 均有如下的 Lipschitz 条件成立,

$$
|g(\tau,v_1)-g(\tau,v_2)| \le C(\eta)\cdot|v_1-v_2| \ \text{ for } |v_1|\le\eta,\ |v_2|\le\eta,\ 0<\tau<t. \tag{4–16}
$$

考虑一个 q 级 p 阶 $S\le p-1$ 阶级的 Runge-Kutta 方法, 满足 L-稳定性条件且 $\max\limits_{\ell}|s_\ell h|\le 1$. 则 Runge-Kutta 方法产生的误差有上界

$$
\left|E_{\mathrm{RK}}^{l}(t)\right| \le C_2\left(|u^{(p)}(0)|+\int_0^{t-t_0}|u^{(p+1)}(\tau)|d\tau\right)h^p. \tag{4–17}
$$

这个定理的证明同样可以在文献[56] 中的定理 4.1 中找到, 因此本文将其省略. 结合公式(4–14), (4–15)和(4–17), 总误差有

$$
|u(t)-u_n| = O(h^d+\varepsilon) \tag{4–18}
$$

对其的详细分析见下一小节.

4.3 误差实例分析

与之前时间卷积积分的数值求解误差类似, 时间卷积积分方程的误差也源自与 VPMR 算法的 SOE 带来的误差与 Runge-Kutta 方法的误差. 下面将以非线性 Volterra 方程为例, 举一个基础的算例, 分析误差的主体部分.

在方程(4–10)中取

$$
f(x)=\frac{1}{\sqrt{x+0.5}},\quad g(x,u(x))=(u(x)-x)^2,\quad u(x)=sin(2x) \tag{4–19}
$$

按公式(4–19)中的函数求得 $a(t)$ 作为已知函数, 这样可以保证目标函数有正弦已知解对照. 取 2 阶 Runge-Kutta 方法, 测试不同参数下步长与 $t=1$ 处总相对误差之间的关系. 如图4–1所示, 红色、绿色、蓝色与黑色曲线分别表示 20 项、30 项、50 项和 100 项 SOE 展开的实

验结果. 从图中可以看出在小步长时, 误差主要由 Runge-Kutta 方法引起, 因为四条曲线均显示出明显服从 Runge-Kutta 方法的收敛阶. 而大步长时, Runge-Kutta 方法的误差已经足够小, 使得 VPMR 算法的误差成为了主体. 最终在排除机器误差之外的因素下, 曲线会收敛于 VPMR 算法的误差. 并且 VPMR 算法的误差越小, 曲线收敛所需的最小步长也越小. 这与我们之前理论计算的误差分析相吻合. 从算例中可以发现, 该算法是一种高精度算法. 但需要注意, 这里由于方程求解器采用了 Newton 迭代法, 在时间步的推进过程中会产生误差累积, 因为 Newton 迭代法的误差无法被完全消去.

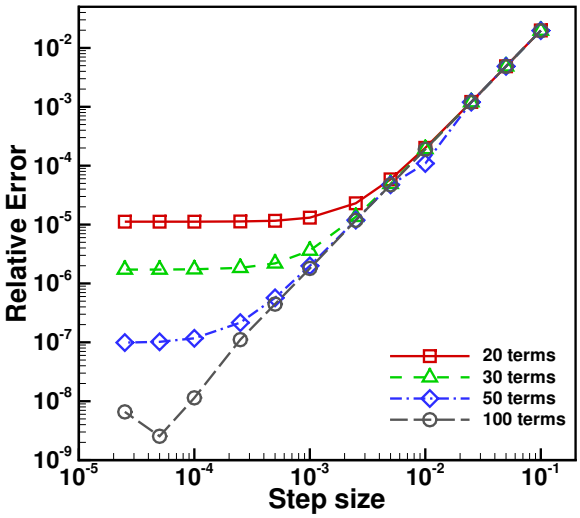

图 4–1 不同参数下步长与总相对误差之间的关系

第五章　多重快速高斯变换

本章主要介绍一种名为快速高斯变换 (Fast Gauss Transform,FGT) 的算法，并分析 VPMR 算法所得到的 SOG 在 FGT 算法中的应用.

在过去的几十年里，核求和问题在许多科学领域都具有重要的意义，包括偏微分方程的数值解、粒子模拟、机器学习和数据科学[66-71]. 核求和问题的快速求和问题是一个被广泛研究的课题，其中解决这个问题的最著名算法为 $O(N)$ 复杂度的快速多极子算法 (Fast Multipole Method,FMM). FMM 算法最初是由 L. Greengard 和 V.Rokhlin 在粒子模拟环境中发展起来的，在粒子模拟环境中，核函数或者其导数在原点处具有奇点[72-73]. FMM 的优点在于它能有效地实现复杂的数据结构，并对新核函数进行自适应[74-76]. 但有时过为复杂的数据结构对于一些光滑核函数在实际操作中是不必要的. 作为一个典型例子，高斯核函数的单能级 FMM 算法的特殊情况, FGT 算法，利用一个我们称为 Hermite 函数的级数构造了多极子和局部展开，从而放弃了自适应网格. 然而快速高斯变换面临着网格与带宽的高度相关性，即网格的设计很大程度上取决于带宽. 并且其只能处理高斯势函数的一大特点大大限制了其广泛应用. 据我们所知，目前不存在一种修正的、与核函数无关且带宽可控的快速高斯变换. 并且对于许多复杂核函数问题 (即使有时候具有很强的各向异性)，我们仍然需要一种简单数据结构保证精度的快速求和算法. 因而本章所提出的多重快速高斯变换很好地解决了这一问题. 与 FGT 相比，该算法的效率几乎相同，并且其可以处理一般光滑核函数.

5.1　快速高斯变换

考虑一个具有 N 个源点与 M 个目标点的系统. 源点在目标点产生的势仅与两点之间的距离和源点电荷量有关，且为一个高斯函数. 因而在目标点 y_j 处所有源点对其产生的势的和为

$$\Phi(y_j) = \sum_{i=1}^{N} q_i e^{-||x_i - y_j||^2/h^2} \tag{5–1}$$

其中 $\{x_i\}$ 为源点的集合, q_i 为对应源点的电荷量. 如果对这个问题直接计算，显然运算复杂度为 $O(MN)$. 由于 M 和 N 的数量级往往接近，故可近似为 $O(N^2)$，即平方复杂度. 这个复杂度显然是不可接受的. FGT 则用线性复杂度 $O(M+N)$ 解决了这一问题[77]. 这一问题通常会在分子模拟 (如 Monte Carlo 方法，分子动力学方法等) 中需要处理.

下进行 FGT 算法的推导. 首先考虑 Hermite 多项式，其定义为

$$H_n(y) = (-1)^n e^{y^2} \frac{d^n}{dy^n}(e^{-y^2}), \quad y \in \mathbb{R} \tag{5–2}$$

例如前几个 Hermite 多项式为

$$\begin{aligned} H_0(y) &= 1 \\ H_1(y) &= 2y \\ H_2(y) &= 4y^2 - 2 \\ H_3(y) &= 8y^3 - 12y \end{aligned} \tag{5–3}$$

由 Hermite 多项式的定义以及 Taylor 展开可以得到

$$e^{2yx-x^2} = \sum_{n=0}^{\infty} \frac{x^n}{n!} H_n(y) \tag{5–4}$$

对公式5–4两端同乘因子 e^{-y^2} 有

$$e^{-(y-x)^2} = \sum_{n=0}^{\infty} \frac{x^n}{n!} H_n(y) e^{-y^2} \tag{5–5}$$

由此可以给出一个定义.

定义 5.1 定义 $h_n(x)$

$$h_n(x) = e^{-y^2} H_n(y) \tag{5–6}$$

从而对于二元高斯函数 $e^{-\frac{(y-x)^2}{h^2}}$ 按变量 x 用 $h_n(x)$ 展开有

$$e^{-\frac{(y-x)^2}{h^2}} = \sum_{n=0}^{\infty} \frac{1}{n!} (\frac{x-x_0}{h})^n h_n(\frac{y-x_0}{h}) \tag{5–7}$$

这个展开称之为 Hermite 展开. 类似地按变量 y 展开有

$$e^{-\frac{(y-x)^2}{h^2}} = \sum_{n=0}^{\infty} \frac{1}{n!} (\frac{y-y_0}{h})^n h_n(\frac{x-y_0}{h}) \tag{5–8}$$

这个展开称之为 Taylor 展开.

定义5.1给出了源和目标之间势函数的展开方式. 除此之外 $h_n(y)$ 有与 Hermite 多项式类似的递推式,

$$h_{n+1}(y) = 2yh_n(y) - 2nh_{n-1}(y), \quad y \in \mathbb{R} \tag{5–9}$$

为了处理高维问题, 引入多维指标 (Multi-index) 来改写展开式. 考虑一个 d 维向量 $x = (x_1, x_2, \cdots, x_d)$ 与一个 d 维多维指标 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d)$, 其中 $\alpha_1, \alpha_2, \cdots, \alpha_d$ 为已知参数, x 与 α 之间的多维指标运算为

$$\alpha! = \alpha_1! \alpha_2! \cdots \alpha_n! \tag{5–10}$$

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n \tag{5–11}$$

$$x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \tag{5–12}$$

$$\frac{d^\alpha}{dx^\alpha} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} \cdots \frac{\partial^{\alpha_n}}{\partial x_n^{\alpha_n}} \tag{5–13}$$

从而按照多维指标的定义, 可得到多重指标 Hermite 多项式的定义:

定义 5.2

$$h_\alpha(y) = e^{-||y||^2} H_\alpha(y) = h_{\alpha_1}(y_1) h_{\alpha_2}(y_2) \cdots h_{\alpha_d}(y_d) \tag{5–14}$$

一致地, Hermite 展开和 Taylor 展开改写为

$$e^{-\frac{(y-x)^2}{h^2}} = \sum_{|\alpha| \geq 0} \frac{1}{\alpha!} (\frac{x-x_0}{h})^\alpha h_\alpha(\frac{y-x_0}{h}) \tag{5–15}$$

$$e^{-\frac{(y-x)^2}{h^2}} = \sum_{|\alpha| \geq 0} \frac{1}{\alpha!} (\frac{y-y_0}{h})^\alpha h_\alpha(\frac{x-y_0}{h}) \tag{5–16}$$

在得到源点和目标点的坐标之后，将点所在空间分割为均匀网格并记录下所有网格中心的坐标. 判定每一个点 (包括源点和目标点) 所在的网格. 之后在公式(5–15)和(5–16)中的 x_0 与 y_0 均取网格中心坐标，从而有如下定义

定义 5.3 对势函数进行展开，定义 S-展开为

$$\Phi(y, x_i) = \sum_{|\alpha|\geq 0} \frac{1}{\alpha!}(\frac{x_i - x_c^l}{h})^\alpha h_\alpha(\frac{y - x_c^l}{h}) \tag{5–17}$$

定义 R-展开为

$$\Phi(y, x_i) = \sum_{|\beta|\geq 0} \frac{1}{\beta!} h_\beta(\frac{x_i - x_c^l}{h})(\frac{y - x_c^l}{h})^\beta \tag{5–18}$$

其中 x_c^l 表示空间内第 l 个网格中心点的坐标.

方便起见，我们设 $a_\alpha(x_i, x_c^l) = \frac{1}{\alpha!}(\frac{x_i - x_c^l}{h})^\alpha$，$S_\alpha(y - x_c^l) = h_\alpha(\frac{y - x_c^l}{h})$，$b_\beta(x_i, x_c^l) = \frac{1}{\beta!} h_\beta(\frac{x_i - x_c^l}{h})$，$R_\beta(y - x_c^l) = (\frac{y - x_c^l}{h})^\beta$. 同时设 $I(l)$ 表示第 l 个网格，从而用这种方法便可把目标点的受到的总势分为每个网格势的和. 第 l 个网格内所有源点产生的势为

$$\begin{aligned}\Phi^l(y) &= \sum_{x_i \in I(l)} q_i \Phi(y, x_i) \\ &= \sum_{x_i \in I(l)} q_i [\sum_{|\alpha|\geq 0} a_\alpha(x_i, x_c^l) S_\alpha(y - x_c^l)] \\ &= \sum_{|\alpha|\geq 0} [\sum_{x_i \in I(l)} q_i a_\alpha(x_i, x_c^l)] S_\alpha(y - x_c^l)\end{aligned} \tag{5–19}$$

从而有如下引理[78].

引理 5.1 $\Phi^l(y)$ 有 S-展开，

$$\Phi^l(y) = \sum_{|\alpha|\geq 0} A_\alpha^l h_\alpha(\frac{y - x_c^l}{h}) \tag{5–20}$$

其中 $A_\alpha^l = \frac{1}{\alpha!} \sum_{x_i \in I(l)} q_i a_\alpha(x_i, x_c^l)$.

由引理5.1可以推导出 FGT 算法的核心计算式，即如下定理.

定理 5.1 $\Phi^l(y)$ 有 SR-展开，

$$\Phi^l(y) = \sum_{|\beta|\geq 0} C_\beta^{lm} (\frac{y - x_c^m}{h})^\beta \tag{5–21}$$

其中 $C_\beta^{lm} = \frac{1}{\beta!} \sum_{|\alpha|\geq 0} A_\alpha^l (-1)^{|\alpha|} h_{\alpha+\beta}(\frac{x_c^l - x_c^m}{h})$，$x_c^l$ 代表第 l 个网格的中心坐标，x_c^m 代表目标点所在网格的中心坐标.

证明 考虑 $h_\alpha(y)$ 关于点 $y_0 \in \mathbb{R}^d$ 的 Taylor 展开，

$$h_\alpha(y) = \sum_{|\beta|\geq 0} \frac{(y - y_0)^\beta}{\beta!} D^\beta h_\alpha(y_0) \tag{5–22}$$

由其定义有

$$h_\alpha(y) = (-1)^\alpha D^\alpha e^{-||y||^2} \tag{5–23}$$

因此

$$D^\beta h_\alpha(y) = (-1)^\beta h_{\alpha+\beta}(y) \tag{5–24}$$

代入到 Taylor 展开有

$$h_\alpha(y) = \sum_{|\beta|\geq 0} \frac{(y-y_0)^\beta}{\beta!}(-1)^\beta h_{\alpha+\beta}(y_0) \tag{5–25}$$

根据引理5.1与公式(5–25)有

$$\begin{aligned}
\Phi^l(y) &= \sum_{|\alpha|\geq 0} A^l_\alpha h_\alpha(\frac{y-x^l_c}{h}) \\
&= \sum_{|\alpha|\geq 0} A^l_\alpha [\sum_{|\beta|\geq 0} \frac{(-1)^\beta}{\beta!}(\frac{y-x^m_c}{h})^\beta h_{\alpha+\beta}(\frac{x^m_c-x^l_c}{h})] \\
&= \sum_{|\beta|\geq 0} [\frac{(-1)^\beta}{\beta!} \sum_{|\alpha|\geq 0} A^l_\alpha (-1)^{|\alpha|+|\beta|} h_{\alpha+\beta}(\frac{x^l_c-x^m_c}{h})](\frac{y-x^m_c}{h})^\beta \\
&= \sum_{|\beta|\geq 0} C^{lm}_\beta (\frac{y-x^m_c}{h})^\beta
\end{aligned} \tag{5–26}$$

□

实际计算中，公式(5–26)中的级数取有限和即可．实验结果表明，较小的截断项 (如 $|\alpha| \leq 4$) 就可以达到一个很高的近似精度．整理 FGT 的算法步骤如下：在算法5–1中，前两步属于

算法 5–1 快速高斯变换 FGT

输入: 源点坐标集合 $\{x_i\}$，目标点坐标集合 $\{y_i\}$ 与网格划分方式

输出: 目标点的势集合 $\Phi(y_i)$

1 确定所有网格的中心点;

2 计算 $\frac{1}{\beta!}(-1)^{|\alpha|} h_{\alpha+\beta}(\frac{x^l_c-x^m_c}{h}), \forall \alpha, \beta, l, m$;

3 计算 C^{lm}_β;

4 **for** $\forall y_i, l$ **do**

5 计算 $\Phi^l(y_i) = \sum_{|\beta|\geq 0} C^{lm}_\beta (\frac{y-x^m_c}{h})^\beta$;

6 计算 $\Phi(y_i) = \sum_l \Phi^l(y_i)$;

预计算部分．对于 N 个源点和 M 个目标点，第二步的运算复杂度为 $O(N)$，循环部分与第六步均为 $O(M)$(这里认为分得网格数的数量级远小于 M 与 N)．从而整个算法的复杂度为 $O(M+N) \sim O(N)$.

但实际上由于展开的独特性与多维指标的设计，FGT 只能应用于势函数为高斯函数的情形．并且由误差分析可知，较大的带宽会有更快的收敛速度和更稀疏的网格分割．这显然给 FGT 算法带来了极大的局限性．因此考虑使用 VPMR 算法与 FGT 的保线性性质，将非高斯函数的核函数进行大带宽高斯函数和的近似，从而使用线性复杂度的算法解决平方复杂度的一般势的计算问题．并且利用其带宽可控的性质，获得更好的收敛精度.

5.2 高斯和的应用

由于 FGT 的保线性性质, P 项 SOG 的计算可以使用 P 次 FGT 解决. 本节将利用 FGT 的计算时, 推导一种更简单的计算方式.

设势函数为 $f(x) \sim f_{eq}(x) \sum_{k=1}^{P} e^{-\frac{x^2}{h_k^2}}$. 则根据公式(5–26)有

$$
\begin{aligned}
f_{eq}(x) &= \sum_{k=1}^{P} e^{-\frac{x^2}{h_k^2}} \\
&= \sum_{k=1}^{P}[\sum_{|\beta|\geq 0} C_\beta^{lmk}(\frac{y-x_c^m}{h_k})^\beta] \\
&= \sum_{k=1}^{P}[\sum_{|\beta|\geq 0}[\frac{1}{\beta!}\sum_{|\alpha|\geq 0} A_\alpha^{lk}(-1)^{|\alpha|}h_{\alpha+\beta}(\frac{x_c^l-x_c^m}{h_k})(\frac{y-x_c^m}{h_k})^\beta]] \\
&= \sum_{|\beta|\geq 0}[\frac{1}{\beta!}\sum_{|\alpha|\geq 0}[(-1)^{|\alpha|}\alpha!\sum_{k=1}^{P}[h_{\alpha+\beta}(\frac{x_c^l-x_c^m}{h_k})\frac{1}{h^{|\alpha+\beta|}}]\sum_{x_i\in I} q_i(x_i-x_c^l)^\alpha(x-x_c^m)^\beta]]
\end{aligned}
\tag{5–27}
$$

从而需要进行计算 P 遍的部分仅有 $\sum_{k=1}^{P}[h_{\alpha+\beta}(\frac{x_c^l-x_c^m}{h_k})\frac{1}{h^{|\alpha+\beta|}}]$, 而剩余的部分仅需计算一次. 从而大大降低了运算复杂度. 在后续章节中给出了相关的精度算例与时间算例, 以体现耦合之后的高精度与线性时间复杂度.

5.3 误差分析

历史文献中对 FGT 的误差分析有很多版本[79-80], 其中一个主要原因为 Greengard 在其发表的第一篇关于快速高斯变换的文章[77] 中对其误差分析出现了错误. 本文所推导的误差分析主要来自于文献[81], 并对其进行部分改动, 使得最终得到的误差上界结果相较原论文更优.

使用公式(5–27)所代表的多重快速高斯变换, 可以对一般核函数应用 FGT 算法. 显然误差来源于两部分: 一是 VPMR 算法得到 SOG 近似产生的误差, 二是 FGT 算法所造成的误差. 与前文所提到的 Runge-Kutta 方法与 VPMR 算法耦合的情况一致, 在 VPMR 算法与 FGT 算法耦合的过程中误差主体由 FGT 算法导致.

首先考虑单指数项 FGT 的误差. 势函数为 $e^{-\frac{(x-y)^2}{h^2}}$, 设公式(5–26)中的级数截断项数为 p. 设

$$
u_p^i(x_i, y_i, c_i) = \sum_{n_i=0}^{p-1} \frac{1}{n_i!}(\frac{y_i-c_i}{h})^{n_i} h_{n_i}(\frac{x_i-c_i}{h}) \tag{5–28}
$$

$$
v_p^i(x_i, y_i, c_i) = \sum_{n_i=p}^{\infty} \frac{1}{n_i!}(\frac{y_i-c_i}{h})^{n_i} h_{n_i}(\frac{x_i-c_i}{h}) \tag{5–29}
$$

这里 i 指代某一个维度. 这里以 3 维系统为例, 即 $x=(x_1,x_2,x_3), y=(y_1,y_2,y_3)$ 有

$$
e^{\frac{-||x-y||^2}{h^2}} = \prod_{i=1}^{3}(u_p^i + v_p^i) \tag{5–30}
$$

方便起见，设 $A_i = e^{-\frac{(x_i-y_i)^2}{h^2}}$，则有

$$A_i = u_p^i + v_p^i \tag{5–31}$$

记误差为 E 则

$$\begin{aligned} E &= |e^{\frac{-||x-y||^2}{h^2}} - u_p^1 u_p^2 u_p^3| \\ &= |(u_p^1 + v_p^1)(u_p^2 + v_p^2)(u_p^3 + v_p^3) - u_p^1 u_p^2 u_p^3| \\ &= |v_p^1 v_p^2 v_p^3 - v_p^1 v_p^2 A_3 - v_p^1 v_p^3 A_2 - v_p^2 v_p^3 A_1 + v_p^1 A_2 A_3 + v_p^2 A_1 A_3 + v_p^3 A_1 A_2| \end{aligned} \tag{5–32}$$

由于

$$e^{-\frac{1}{h^2}} \le A_i \le 1,\ \ 0 < v_p^i \le 1 \tag{5–33}$$

从而

$$\begin{aligned} E &\le v_p^1 v_p^2 v_p^3 + v_p^1 + v_p^2 + v_p^3 - e^{-\frac{1}{h^2}}(v_p^1 v_p^2 + v_p^1 v_p^3 + v_p^2 v_p^3) \\ &\le (1 - 3e^{-\frac{1}{h^2}}) v_p^1 v_p^2 v_p^3 + v_p^1 + v_p^2 + v_p^3 \end{aligned} \tag{5–34}$$

于是考虑单独对 v_p^i 的上界估计，

$$\begin{aligned} v_p^i &= \sum_{n_i=p}^{\infty} \frac{1}{n_i!} (\frac{y_i - c_i}{h})^{n_i} h_{n_i}(\frac{x_i - c_i}{h}) \\ &\le \sum_{n_i=p}^{\infty} \frac{1}{n_i!} (\frac{r}{2h})^{n_i} 2^{\frac{n_i-p}{2}} \sqrt{\frac{n_i!}{p!}} h_p(\frac{x_i - c_i}{h}) \\ &\le 2^{-\frac{p}{2}} \sqrt{\frac{1}{p!}} h_p(\frac{x_i - c_i}{h}) \sum_{n_i=p}^{\infty} r^{n_i} (2\pi)^{-0.25} n_i^{-\frac{n_i}{2-0.25}} e^{\frac{n_i}{2}} \\ &\le 2^{-\frac{p}{2}} \sqrt{\frac{1}{p!}} h_p(\frac{x_i - c_i}{h}) (2\pi p)^{-0.25} \frac{r_p^p}{1 - r_p} \\ &\le TK \end{aligned} \tag{5–35}$$

其中 $T = 2^{-\frac{p}{2}} \sqrt{\frac{1}{p!}} \max_{[0,\frac{1}{h}]} h_p(x), K = (2\pi p)^{-0.25} \frac{r_p^p}{1 - r_p}, r_p = r\sqrt{\frac{e}{p}}$，且 r 代表分割的网格边长. 注意公式(5–35)用到了 Stirling 公式与一个事实，即对 $\forall n \ge p, x \in \mathbb{R}$ 有

$$h_n(x) \le 2^{-\frac{n-p}{2}} \sqrt{\frac{n!}{p!}} h_p(x) \tag{5–36}$$

由此可见误差上界与带宽 h 有密切关系. 因而可控带宽的 VPMR 算法对于 FGT 算法的耦合可以有效减小误差.

由于 FGT 本身具有一些其它的的局限性，例如高维会严重陷入维度诅咒，网格利用率可能不高等，即使 VPMR 算法耦合之后也无法解决这些固有问题. FGT 目前有许多改进的版本，如可以自适应网格的改进快速高斯变换 (Improved Fast Gauss Transform, IFGT)[82]，结合树结构的 FGT 算法 (Tree-FGT)[83] 等. 未来这一方面的一个拓展方向即为将 VPMR 算法耦合到这些改进后的 FGT 的算法中去.

第六章　数值算例

本章所有算例都是在具有 32GB 内存, 2.50GHz 的 IntelTM 核心处理器的个人计算机上进行的. 计算平台为 MATLAB2019a. 所有代码皆为单线程串行计算.

6.1　高斯和展开

在本节中，我们给出数值结果来说明 VPMR 算法对于非奇异目标函数 SOG 估计的性能. 这里采用四种不同的核函数来度量 VPMR 算法的性能. 它们分别为具有小带宽的高斯核函数 f_{gau}，反二次核函数 f_{imq}, Ewald 分裂核函数 f_{ewd} 和 Matérn 核函数 f_{mat}，写做

$$f_{\text{gau}}(x) = e^{-x^2/h^2}, \tag{6–1}$$

$$f_{\text{imq}}(x) = \frac{1}{\sqrt{1/2 + x^2}}, \tag{6–2}$$

$$f_{\text{ewd}}(x) = \frac{\text{erf}(\alpha x)}{x}, \tag{6–3}$$

$$f_{\text{mat}}(x) = \frac{(\sqrt{2\nu}|x|)^{\nu} K_{\nu}(\sqrt{2\nu}|x|)}{2^{\nu-1}\Gamma(\nu)}, \tag{6–4}$$

其中 $\text{erf}(x) = (2/\sqrt{\pi})\int_0^x \exp(-u^2)du$ 为误差函数, K_ν 为 ν 阶的第二类修正的贝塞尔函数, Γ 为 Gamma 函数. 计算中取参数 $h = 0.1$, $\alpha = 1$ 和 $\nu = 2$. 我们注意到，高斯核函数和反二次核函数是广泛使用的径向基函数，用于许多数据科学和工程问题[84-85]. Ewald 分裂核函数则是来自于著名的库仑相互作用的 Ewald 求和算法中的长程部分，即

$$\frac{1}{x} = \frac{\text{erf}(\alpha x)}{x} + \frac{\text{erfc}(\alpha x)}{x} \tag{6–5}$$

中的前半部分，常研究其在 Fourier 空间中的性质，参数 α 描述了截断半径的倒数[86-88]. 最后, Matérn 核函数也是一个常用的具有复杂形式的径向基函数，常用作建模高斯过程的协方差函数，其中参数 ν 描述核的光滑性[10]. 其常用性质有 $f_{mat}(0) = 1, \forall\nu$ 而且在 ν 较大时, $f_{mat}(x)$ 在原点具有高阶可微的性质.

首先探究项数 P 与 SOG 逼近过程中的误差关系. 为了衡量 SOG 逼近的精度，我们定义误差公式为

$$\epsilon_\infty = \frac{\max\left\{|f_p(x_i) - f(x_i)|, i = 1, \cdots, M\right\}}{\max\left\{|f(x_i)|, i = 1, \cdots, M\right\}}, \tag{6–6}$$

其中 $f_p(x)$ 为 SOG 展开, $p = 2n$ 且这个展开没有经过 MR 技术处理. $\{x_i, i = 1, \cdots, M\}$ 为区间 $[0, 1]$ 内的取样点. 这里取 $M = 1000$. 该误差可以看作是连续 L^∞ 范数的近似.

误差实验的结果如6–1所示. 子图 a 中给出了固定最小带宽下，误差与项数之间的关系. 参数取 $n_c = \lceil n/4 \rceil$，即最小带宽固定为 $s_p \approx \sqrt{1/8}$. 子图 b 中则给出了固定项数下，误差与最小带宽之间的关系. 参数取 $p = 10000$. 两幅图中 Gauss, IMQ, Ewald 和 Matern 分别代表着高斯核函数 f_{gau}，反二次核函数 f_{imq}, Ewald 分裂核函数 f_{ewd} 和 Matérn 核函数 f_{mat}，对应着红色，绿色，蓝色以及灰色曲线. 在子图 a 中我们观察到，对于所有四个核函数的 SOG 近似都有很高的精度，并且收敛速度是非常快的. 不同核函数的收敛速度不一致，没有明显的共

同特点. 这里要指出高斯核函数的原带宽 $h = 0.1$, 但经过 SOG 处理之后 $s_j \gg h$, 且收敛情况良好. 因而给出了一种大带宽高斯信号逼近小带宽高斯信号的构造. 这对于信号处理有着十分重要的意义. 而在子图 b 中发现对于高斯核函数和 Matérn 核函数, 当最小带宽降低时, 估计的精度显著提高. 然而, 另外两个核函数的 SOG 近似似乎对最小带宽的变化并不敏感, 因为它们是更平滑的函数, 而且傅里叶空间中的高频分量非常小. 从而说明误差与最小带宽没有明显的关系, 更多与函数性质有关.

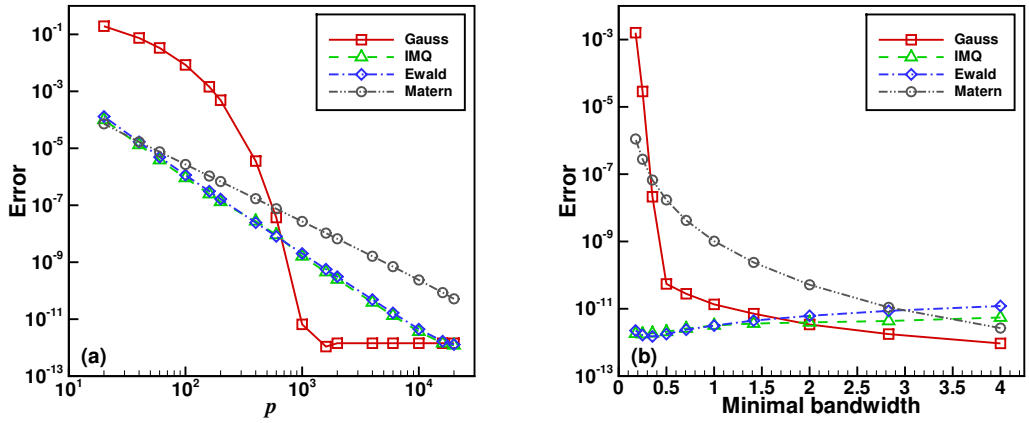

图 6–1 误差 ϵ_∞ 与项数和最小带宽的关系

在实际计算中, 高斯系数的大小与舍入误差密切相关. 有必要探究 SOG 展开最小带宽的变化对系数大小的影响. 图6–2中展示了 SOG 中系数的最大绝对值 $w_{\max} = \max\{|w_j|, j = 0, 1, \cdots, p-1\}$ 与最小带宽 s_p 之间的关系. 子图 a 和 b 分别固定项数 $p = 20$ 和 40, 红色, 绿色, 蓝色以及灰色曲线分别表示高斯核函数 f_{gau}, 反二次核函数 f_{imq}, Ewald 分裂核函数 f_{ewd} 和 Matérn 核函数 f_{mat} 的结果. 如图6–2两幅图的比较中可以看出, 最大系数的绝对值会随着用于 SOG 近似的项数的增加而增加. 而最小带宽对于系数最大绝对值没有明显的影响规律. 而且两幅子图均显示出明显的大系数, 直接应用会引起极大的机器误差而导致实验失败. 本实验采用了多精度工具箱保存了每一个数据足够多的有效数字 (如 300 位), 从而可以测量出精确的误差结果. 因而在实际应用中 MR 技术是十分必要的, 因为处理之后的数据只用 double 双精度数据结构便可以保存全部信息. 后续会有实验显示 MR 技术的实际效果.

下面将展示 VPMR 算法与最小二乘法 (Least square method,LSM) 的比较. 对于 LSM, 我们使用完全正交分解来计算拟合矩阵的低秩近似, 以对抗病态矩阵的存在. 如图6–3所示, 四副子图分别展示了高斯核函数 f_{gau}, 反二次核函数 f_{imq}, Ewald 分裂核函数 f_{ewd} 和 Matérn 核函数 f_{mat} 的 VPMR 算法的结果和 LSM 的结果, 定义域为 $x \in [0, 1]$. 其中灰色, 蓝色, 红色以及绿色曲线分别代表 200 项 VPMR 算法结果, 800 项 VPMR 算法结果, 200 项 LSM 结果与 800 项 LSM 结果. 显然无论哪一种核函数, VPMR 算法都提供了更为精确的结果, 误差基本可以达到 10^{-9}. 甚至 200 项 VPMR 算法结果的精度要高于 800 项 LSM 结果的精度. 由于 LSM 较难探测到大系数的结果, 因而最后结果更弱. 如果强行拓宽 LSM 的探测区间, 其运算时间是不可接受的.

最后考虑 MR 技术在 VPMR 算法中的作用. 由 VP 和产生的 SOG 估计面临着线性系数过大以及冗余信息过多等问题, 无法直接应用到其它算法中去. 于是探究 MR 方法对于 VP 和这两个问题的解决能力. 如表格6–1与表格6–2所示, 它们分别计算了 f_{imq} 与 f_{mat}100 项

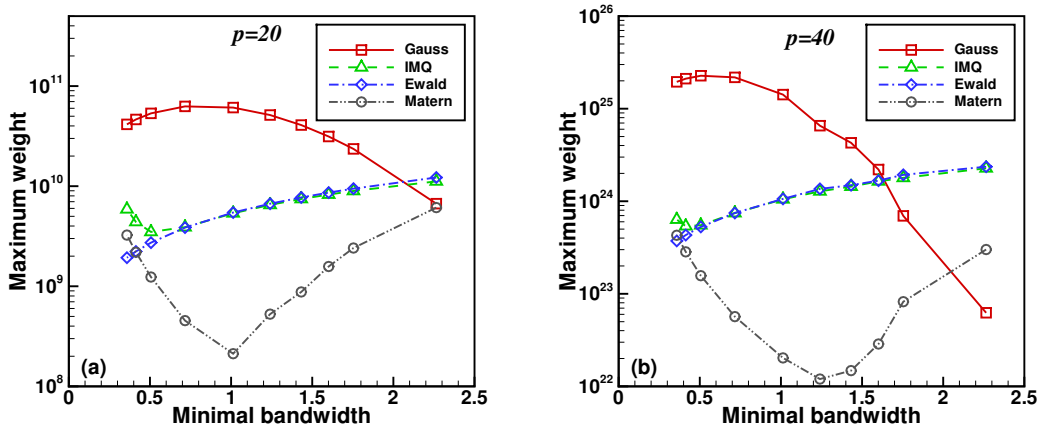

图 6–2 系数的最大绝对值 $w_{\max}$ 与最小带宽 s_p 之间的关系

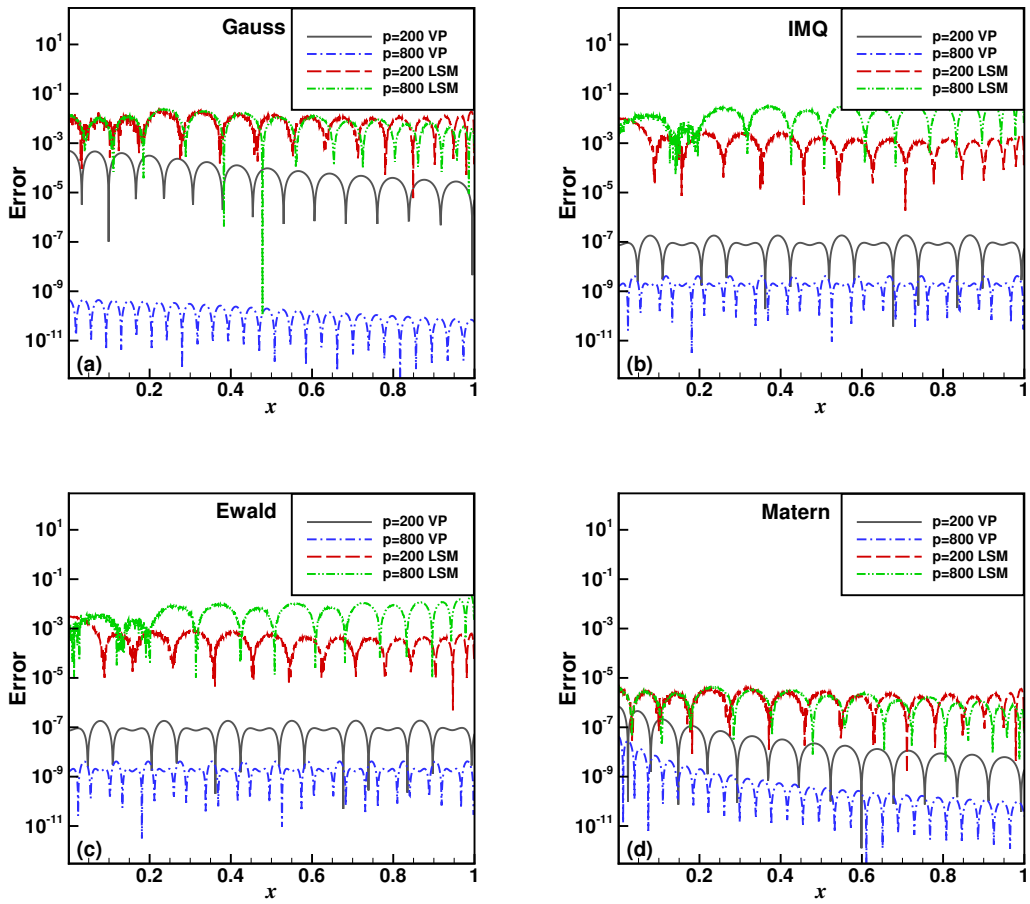

图 6–3 四种不同核函数的 VPMR 结果与 LSM 结果的比较

SOG 的模型降阶结果. 统计了降阶到 100 项 (未降阶情形), 90 项, 70 项, 50 项, 30 项以及 10 项的线性系数绝对值的最大值 $\widetilde{w}_{\max}$, 最小带宽 s_q 以及降阶后产生的误差 ϵ_∞. 从结果中可以看出进行了降阶之后, 线性系数绝对值最大值有非常明显的减少, 而最小带宽变化幅度不大. 这与第二章的理论结果高度一致. 并且降阶后产生的误差有时可以在降阶项数较小时也可以保持很好, 例如 f_{imq} 可以降阶到 70 项以保留原有精度, 约简率约为 30%. 而 f_{mat} 甚至可以降阶到 30 项就可以保留原有精度, 约简率达到了 70%. 实际上模型降阶的效果与函数性质有密切关系, 但无论何种函数, 模型降阶均可以大幅降低其线性系数绝对值, 保证最小带宽没有太大变化, 从而完美弥补 VP 和所带来线性系数绝对值过大的问题. 由此可以证明 MR 技术在 VPMR 算法中具有显著效果. 后续实验所使用的 VPMR 算法的结果都是测试的保证误差不变的最优约简项数结果.

表 6–1 f_{imq}100 项 SOG 的模型降阶结果

降阶项数 q	$\widetilde{w}_{\max}$	s_q	ϵ_∞
100	5.96e+68	0.361	2.36e-6
90	37.5	0.201	2.36e-6
70	13.7	0.346	2.66e-6
50	6.90	0.363	2.34e-5
30	2.31	0.421	1.87e-4
10	2.31	0.665	1.03e-2

表 6–2 f_{mat}100 项 SOG 的模型降阶结果

降阶项数 q	$\widetilde{w}_{\max}$	s_q	ϵ_∞
100	5.70e+64	0.361	3.87e-6
90	0.335	0.131	3.87e-6
70	0.467	0.122	3.88e-6
50	0.309	0.113	3.89e-6
30	0.246	0.116	5.68e-6
10	0.274	0.153	1.84e-5

6.2 指数和展开

在本节中, 我们给出数值结果来说明 VPMR 算法对于目标函数 SOE 估计的性能. 与上一节类似, 本节也将对四种不同的核函数进行实验: Matérn 核函数, 幂核函数, Ewald 分裂核函数和 Helmholtz 核函数. 第一种与第三种核函数已经于上一节介绍, 下面介绍其余两种核函数,

这里考虑的幂核函数 $f(x) = x^{\alpha-1}$ 是弱奇异的, 在计算物理中有着广泛的应用[3, 11]. 幂核函数有如下式的逆 Laplace 变换的形式[12],

$$x^{\alpha-1} = \frac{1}{\Gamma(1-\alpha)} \int_{-\infty}^{\infty} e^{-e^t x + (1-\alpha)t} dt. \tag{6–7}$$

公式(6–7)的数值积分产生了一个显式的离散化方法来得到一个 SOE. 虽然用积分表示构造 SOE 近似可以达到给定的精度，但带宽的不可控制限制了实际的使用. 而 Helmholtz 核函数是来自于 Helmholtz 方程的 Green 函数，它具有很强的震荡性质. 在二维和三维系统里, Helmholtz 核函数的形式分别为

$$f_{2\mathrm{D}}(x) = \frac{i}{4}H_0^{(1)}(kx), \quad f_{3\mathrm{D}}(x) = \frac{e^{ikx}}{4\pi x}, \tag{6–8}$$

其中 $H_0^{(1)}$ 为第一类 Hankel 函数. 高波数的 Helmholtz 方程很难用数值来求解，因为波数 k 的值越大, Helmholtz 核函数的振荡就越强. 由于这个问题, Helmholtz 核函数尝试用一个有效的 SOE 近似非常困难.

如图所示展示了四种不同核函数在不同参数下使用 VPMR 算法的 SOE 结果. 子图 abcd 分别展示了 Matérn 核函数，幂核函数, Ewald 分裂核函数和 Helmholtz 核函数的结果. 一副子图中不同颜色的曲线代表着不同的实验参数. 子图 a 中蓝色，绿色和红色曲线分别表示 Matérn 核函数 $\nu = 1.0, 2.0$ 与 4.0 的结果. 子图 b 中蓝色，绿色和红色曲线分别表示幂核函数 $\alpha = 0.1, 0.5$ 与 0.9 的结果. 子图 c 中蓝色，绿色和红色曲线分别表示 Ewald 分裂核函数截断半径的倒数为 1.0, 2.0 与 4.0 的结果. 子图 d 中蓝色与绿色曲线分别表示 Helmholtz 核函数二维体系与三维体系下的结果. 可以从图中观察到随着项数的增加，所有核函数的逼近误差均在减小，并达到一个较高的精度. 由于不同函数的本身性质不同，其收敛速度也会有明显的不同.

6.3 卷积积分

本节将展示卷积积分计算求值的相关算例. 首先考虑非奇异核函数的情形. 在公式(3–4)中取核函数为高斯函数 $f(\tau) = e^{-\tau^2/4}$，卷积函数为正弦函数 $g(\tau) = \sin\tau$，计算卷积积分

$$y(t) = \int_0^t e^{-\frac{(t-\tau)^2}{4}} \sin\tau d\tau. \tag{6–9}$$

在处理误差的时候，本文认为" 真实解" 来自于自适应 Gauss-Kronrod 数值积分所得到的绝对误差不超过 10^{-14} 的数值解. 表格6–3中展示了不同时间步长下在时间 $t = 1, 4, 10$ 的绝对误差与收敛阶. 采用的 SOE 参数为 $\varepsilon = 8.1e-14$, $n_c/(2n-1) = 1/8$, 项数为 $P = 20$. 从表格中可以清楚看出，在误差允许的范围内，可以认为收敛阶为 4 阶，这与采用的 4 阶 Runge-Kutta 方法所一致，说明 VPMR 算法的误差不为误差的主体. 除此之外也发现误差不会随着时间的推移而积累，这是一个非常好的性质. 图6–5则展示了 CPU 时间算例. 从图中可以看出该算法是一个线性时间复杂度的算法，其中散点为实验真实数据，直线为对散点的拟合曲线，红色点，绿色点与蓝色点分别表示了时间步长为 0.05,0.01 与 0.005 的结果. 这与我们的时间复杂度分析理论高度一致. 并且收敛阶的稳定代表误差主体由 Runge-Kutta 方法提供，而非 VPMR 算法. 最终误差可以达到 10^{-13} 数量级，也体现出高精度的特点.

再考虑奇异核函数的情形. 取核函数为幂核函数 $f(\tau) = \tau^{\alpha-1}$，其中 $0 < \alpha < 1$，卷积函数为余弦函数 $g(\tau) = \cos\tau$. 这个卷积积分也称为 Riemann-Liouville 分数阶积分[60]，通常写做

$$y(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \cos\tau d\tau \tag{6–10}$$

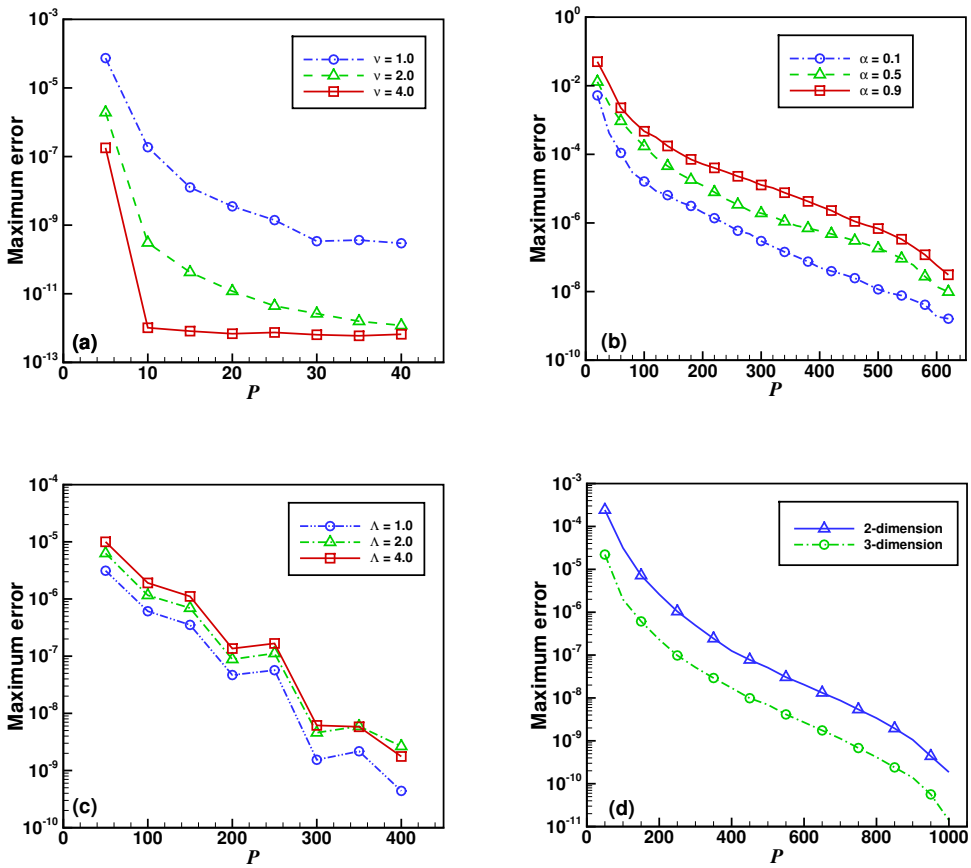

图 6–4 四种不同核函数在不同参数下的 SOE 结果

其中本文所选用的卷积积分有解析解

$$y(t) = \frac{2^{1-\alpha}\sqrt{\pi}t^{\alpha}}{\alpha\Gamma\left(\frac{\alpha}{2}\right)\Gamma\left(\frac{1+\alpha}{2}\right)}\hat{\mathrm{H}}\left(1, \left[\frac{1}{2}(1+\alpha), 1+\frac{1}{2}\alpha\right], -\frac{t^2}{4}\right), \tag{6–11}$$

其中 $\hat{\mathrm{H}}(\{a_i\}_{i=1}^{\eth_1}, \{b_j\}_{j=1}^{\eth_2}, \tau)$ 为广义超几何函数，定义为

$$\hat{\mathrm{H}}(\{a_i\}_{i=1}^{\eth_1}, \{b_j\}_{j=1}^{\eth_2}, \tau) = \sum_{\ell=0}^{\infty}\left(\frac{\prod_{i=1}^{\eth_1}(a_i)_\ell}{\prod_{j=1}^{\eth_2}(b_j)_\ell}\right)\left(\frac{\tau^\ell}{\ell!}\right), \tag{6–12}$$

其中 $\eth_1$ 和 $\eth_2$ 为两个正整数，$(\cdot)_\ell = \Gamma(\cdot+\ell)/\Gamma(\cdot)$ 为 Pochhammer 符号. 在 SOE 估计中，分别取参数 $n_c/(2n-1) = 0.15, 0.2$ 与 $\alpha = 0.1, 0.5, 0.9$. 项数为 $P = 640$，保证 SOE 误差控制在 $\sim 10^{-9}$. 按照公式(3–25)，卷积积分被分裂为 I_1 与 I_2. 这里对 I_1 采用四阶格式，对 I_2 采用四阶 Lobatto IIIC 方法. 表格6–4中展示了不同时间步长与不同 α 下在时间 $t = 1, 4, 8$ 的绝对误差与收敛阶. 所有数据均显示出四阶收敛性，与理论分析一致. 在步长为 0.025 时即可达到 10^{-9} 左右的精度，可以说明该算法的高精度特性.

研究指出，幂核函数作为核函数往往需要一个大的项数 P 来实现高精度，文献中已经对它的 SOE 方法进行了研究[3, 11-12]. 通常，应使用数百个指数项来实现 8 ~ 9 位有效数字

表 6–3 对于不同时间 t 求解公式(6–9)所产生的绝对误差与收敛阶

步长 h	$t=1$	阶数	$t=4$	阶数	$t=10$	阶数
0.5	$6.60e-5$	-	$3.47e-5$	-	$4.08e-5$	-
0.25	$4.49e-6$	3.88	$3.31e-6$	3.39	$3.53e-6$	3.53
0.1	$1.19e-7$	3.93	$1.03e-7$	3.62	$1.06e-7$	3.70
0.05	$7.46e-9$	3.95	$6.79e-9$	3.71	$6.90e-9$	3.77
0.025	$4.68e-10$	3.96	$4.36e-10$	3.77	$4.41e-10$	3.82
0.01	$1.20e-11$	3.97	$1.14e-11$	3.82	$1.15e-11$	3.86
0.005	$7.21e-13$	3.98	$6.96e-13$	3.85	$7.10e-13$	3.88

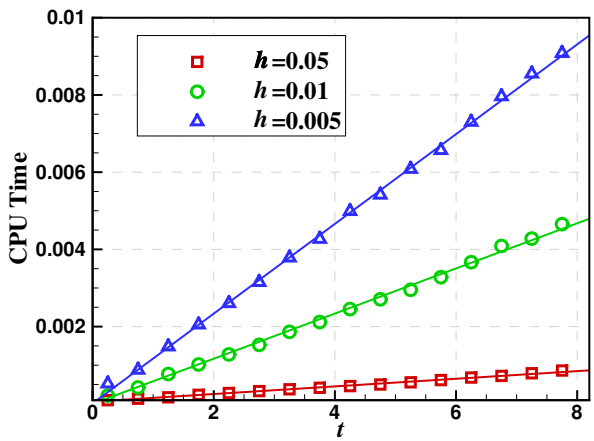

图 6–5 不同参数下消耗的 CPU 时间

的误差，但最大指数为 $\sim 10^3$，这大大降低了 Runge-Kutta 方法的收敛精度. 在这一精度水平上, VPMR 算法的指数项数更少，并且由于带宽可控，因此 Runge-Kutta 方法会具有更好的性能. 我们注意到，如果引入更好的技术，诸如 Ewald 分裂技术[88]，并通过 SOE 延拓近似光滑部分，卷积积分的计算将得到大幅改进.

6.4 卷积积分方程

本节将会计算如公式(4–1)的时间卷积积分方程. 首先考虑核函数非奇异的算例. 取核函数为高斯函数，即

$$g(t)+\frac{\sqrt{\pi}}{2e}[f_1(t)+f_2(t)]-\cos(t)=\int_0^t e^{-\frac{(t-\tau)^2}{4}}g(\tau)d\tau, \tag{6–13}$$

其中

$$f_1(t)=\left[\mathrm{erf}\left(\frac{1}{2}(t-2i)\right)+\mathrm{erf}\left(\frac{1}{2}(t+2i)\right)\right]\cos(t), \tag{6–14}$$

$$f_2(t)=\left[-\mathrm{erfi}\left(1-\frac{it}{2}\right)-\mathrm{erfi}\left(1+\frac{it}{2}\right)+2\mathrm{erfi}(1)\right]\sin(t), \tag{6–15}$$

这里 $\mathrm{erf}(\cdot)$ 与 $\mathrm{erfi}(\cdot)$ 分别表示误差函数与误差函数的虚部. 卷积方程(6–13)具有真实解 $g(t)=\cos t$. 算例中所采用的 Runge-Kutta 方法仍为四阶 Lobatto IIIC 方法. SOE 的参数为 $\varepsilon=8.1e-$

表 6–4 不同时间 t 与不同 α 下计算公式(6–10)的误差与收敛阶

α	步长 h	$t=1$	阶数	$t=4$	阶数	$t=8$	阶数
0.1	0.25	$4.11e-5$	-	$4.11e-5$	-	$7.97e-5$	-
	0.1	$4.61e-6$	3.64	$1.73e-6$	3.45	$3.04e-6$	3.57
	0.0625	$7.80e-7$	3.69	$3.10e-7$	3.52	$5.29e-7$	3.62
	0.05	$3.32e-7$	3.71	$1.35e-7$	3.55	$2.28e-7$	3.64
	0.025	$2.25e-8$	3.76	$9.62e-9$	3.63	$1.58e-8$	3.70
0.5	0.25	$4.22e-5$	-	$1.02e-5$	-	$2.41e-5$	-
	0.1	$1.40e-6$	3.72	$3.95e-7$	3.55	$8.26e-7$	3.68
	0.0625	$2.31e-7$	3.76	$6.85e-8$	3.61	$1.39e-7$	3.72
	0.05	$9.75e-8$	3.77	$2.94e-8$	3.63	$5.92e-8$	3.73
	0.025	$6.55e-9$	3.81	$2.40e-9$	3.63	$4.34e-9$	3.74
0.9	0.25	$5.54e-6$	-	$1.55e-6$	-	$3.74e-6$	-
	0.1	$1.69e-7$	3.81	$4.82e-8$	3.78	$1.09e-7$	3.86
	0.0625	$2.72e-8$	3.84	$7.49e-9$	3.84	$1.58e-8$	3.94
	0.05	$1.14e-8$	3.84	$2.94e-9$	3.89	$5.41e-9$	4.06
	0.025	$8.88e-10$	3.80	$1.96e-10$	3.90	$1.47e-9$	3.41

14, $n_c/(2n-1)=1/8$ 与 $P=20$. 实验结果如表格6–5所示. 从实验结果来看, 误差阶为 4 阶, 与我们的理论推导高度一致. 且最终误差数量级很小, 反映了该算法的高精度.

表 6–5 对于不同时间 t 求解公式(6–13)所产生的绝对误差与收敛阶

步长 h	$t=1$	阶数	$t=4$	阶数	$t=8$	阶数
0.1	$3.25e-6$	-	$1.47e-5$	–	$1.71e-4$	–
0.05	$2.17e-7$	3.91	$9.50e-7$	3.95	$1.12e-5$	3.94
0.025	$1.41e-8$	3.92	$6.16e-8$	3.95	$7.27e-7$	3.94
0.01	$3.73e-10$	3.94	$1.62e-9$	3.96	$1.92e-8$	3.95
0.005	$2.35e-11$	3.95	$1.02e-10$	3.96	$1.21e-9$	3.96
0.0025	$1.71e-12$	3.92	$6.86e-12$	3.95	$8.27e-11$	3.94

再考虑奇异核函数的情形. 考虑具有奇异核的 Abel 方程

$$3g(t)+H(t)=\int_0^t (t-\tau)^{-\alpha}g(\tau)d\tau, \tag{6–16}$$

其中取合适的 $H(t)$ 使得待解函数 $g(\tau)=\cos\tau$, 从而原 Abel 方程有真实解余弦函数. SOE 所采用的的参数为 $\varepsilon=1.4e-8$, $n_c/(2n-1)=0.2$ 和 $P=600$. 算例中取 $\alpha=0.5$. 运算结果如表格6–6所示. 同样地, 误差收敛阶为 4, 与理论计算相同.

最后一类方程考虑非线性 Volterra 积分方程. 同样地也分为核函数非奇异情况和奇异情况第一种情况出现在具有抑制后反弹的神经网络的分析中[89], 形式为

$$u(t)=1+\int_0^t (t-\tau)^3(4-t+\tau)e^{-t+\tau}\frac{u^4(\tau)}{1+2u^2(\tau)+2u^4(\tau)}d\tau. \tag{6–17}$$

表 6–6 对于不同时间 t 求解公式(6–16)所产生的绝对误差与收敛阶

步长 h	$t=2$	阶数	$t=6$	阶数	$t=10$	阶数
0.025	$2.60e-8$	-	$9.80e-6$	–	$3.95e-7$	–
0.01	$1.13e-8$	3.74	$4.25e-8$	3.74	$1.71e-7$	3.74
0.00625	$1.47e-9$	4.14	$5.20e-9$	4.24	$1.97e-8$	4.33
0.005	$5.30e-10$	4.25	$1.90e-9$	4.30	$6.80e-9$	4.33

方程(6–17)在 $t=10$ 的精确解为 $u(10)=1.25995582337$[89]. 我们用 VPMR 算法计算 $t=10$ 的数值解. 插值阶数取 4 阶, SOE 的参数为 $\varepsilon=10^{-12}$, $n_c/(2n-1)=2.25$ 和 $P=170$. 牛顿迭代法所产生的误差控制在 10^{-12} 以内. 表格6–7展示了绝对误差, 收敛阶以及相应的 CPU 时间. 同理论分析, 数值算例显示了该算法的四阶收敛性. 由于牛顿迭代法在不同时间步有不同的复杂程度, CPU 时间近似于线性增长, 这表现出非常优良的性能.

表 6–7 求解 Volterra 方程(6–17)的绝对误差, 收敛阶以及相应的 CPU 时间

步长 h	误差	阶数	CPU 时间
1.25	$5.76e-2$	–	$1.0e-4$
1	$2.65e-2$	3.48	$1.2e-4$
0.625	$3.91e-3$	3.88	$1.8e-4$
0.5	$1.44e-3$	4.02	$2.7e-4$
0.25	$4.64e-5$	4.43	$4.7e-4$
0.0625	$2.48e-7$	4.12	$1.3e-3$
0.05	$1.43e-7$	4.01	$1.8e-3$
0.01	$1.90e-10$	4.05	$7.4e-3$

第二种非线性 Volterra 方程的算例是有一个奇异的核函数, 通常产生于超流体的理论中[65]. 方程为

$$u(t)=-\int_0^t \frac{(u(\tau)-\sin\tau)^3}{\sqrt{\pi(t-\tau)}}d\tau. \tag{6–18}$$

在计算过程中, 采用参数 $t_0=0.05$, $\varepsilon=1.40e-8$, $n_c/(2n-1)=0.4$ 和 $P=640$, 插值阶数取 4 阶. 真实解用步长为 $h=0.0001$ 时的数值解代替. 牛顿迭代法产生的误差控制在 10^{-10} 以内. 表格6–8给出了数值结果, 可见其收敛阶仍然控制在 4 阶, 表现出较好的数值精度.

6.5 多重快速高斯变换

本算例探究 VPMR 算法对于 FGT 算法的作用. 首先考虑精度算例. 测试势函数为不同参数下的各向同 Matérn 核函数. 探究在多重快速高斯变换中, 最大相对误差与 SOG 展开项数之间的关系. 其中最大相对误差定义为

$$\epsilon_\infty := \frac{\max\limits_{i=1,\cdots,N}|\Phi_n(x_i)-\Phi(x_i)|}{\max\limits_{i=1,\cdots,N}|\Phi(x_i)|} \tag{6–19}$$

表 6–8 求解 Volterra 方程(6–18)的绝对误差与收敛阶

步长 h	$t=2$	阶数	$t=6$	阶数	$t=10$	阶数
0.025	$3.33e-8$	-	$1.31e-7$	–	$7.39e-8$	–
0.0125	$2.00e-9$	4.06	$8.39e-9$	3.97	$4.14e-9$	4.16
0.01	$8.78e-10$	3.97	$3.63e-9$	3.92	$1.74e-9$	4.09
0.00625	$1.84e-10$	3.75	$6.75e-10$	3.80	$1.72e-10$	4.37
0.005	$1.33e-10$	3.43	$4.34e-10$	3.55	$1.04e-10$	4.08

其中 N 代表目标点数和源点数. 不失一般性, 在二维单位正方形中随机生成目标点与源点. 如图6–6所示, 我们测试了 $\nu=1.0$ 与 1.5 最小带宽为 1 与 4 的情况. 蓝色, 红色, 黄色和紫色线则表示了一共四种不同的参数设计. 由于其最小带宽不同, 其相应网格分割为 4×4 与 3×3, 分别对应最小带宽为 1 与 4. 可以看到, 无论哪一种参数, 误差收敛速度均非常迅速, 在项数不到 100 项的时候, 已经可以保证最大相对误差不大于 10^{-5}, 某些参数甚至可以达到 10^{-7}. 并且其网格分割较为稀疏. 这一算例可以充分体现出多重快速高斯变换所具备的高精度, 带宽可控, 收敛速度快等优良特点.

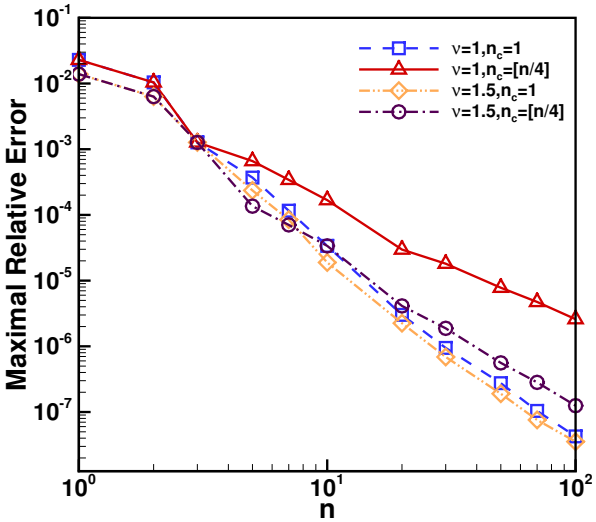

图 6–6 多重 FGT 中 SOG 项数与误差之间的关系

再考虑 CPU 时间算例. 选定 2 维的单位正方形作为全空间, 测试势函数为不同参数下的各向异 Matérn 核函数, 网格分割保证每一个目标点的势相对误差不大于 10^{-5}. 设定目标点数与源点数始终相等为 N, 且两种点均服从均匀分布. 简便起见, 设每个源的电荷量均为 1. 时间测试采用多次测试的平均结果. 如图6–7给出了点数 N 与 CPU 时间的关系. 子图 A 和 B 分别测试了 $\nu=1.0$ 与 1.5 的情况. 红色线代表各向同性, 而黄色线代表强各向异性的情况, 即 x 方向与 y 方向对势的贡献不同. 可见在点数达到一定数量时, 多重 FGT 算法保持了非常好的线性时间复杂度. 并且多重 FGT 算法的时间曲线与直接计算曲线交点的横坐标 (Breakeven 点) 很小, 不到 100. 这代表着该算法在很小的体系就可以代替直接计算以节约运算时间, 可见 VPMR 算法与 FGT 算法共同作用效果良好.

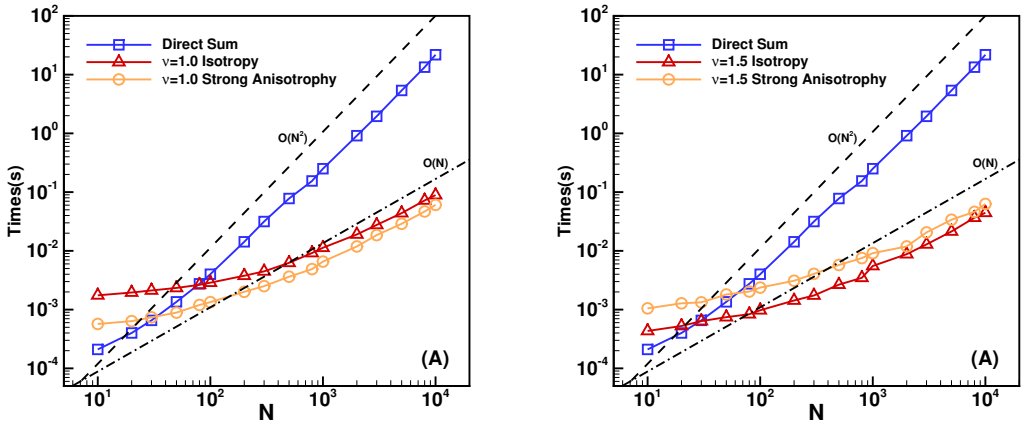

图 6–7 点数与 CPU 时间的关系

综上所述的所有算例，我们首先证明了 VPMR 算法对于 SOG 估计和 SOE 估计的高精度效应，后从精度和运算复杂度两个方面证明了 VPMR 算法得到的 SOG 或 SOE 在其它成熟算法如 Runge-Kutta 方法, FGT 算法等较优秀的耦合性质. 通过把问题改写为若干个高斯函数或指数函数的问题，大幅简化了原有问题，提高了相应精度并降低了运算复杂度. 并且 VPMR 算法自身的高精度与带宽可控的优良性质也使得其耦合适配性很强. 除此之外, SOG 和 SOE 所代表的高并行适配性也使得其在耦合过程中可以设计高效率的并行方式，大大提高了算法的运行效率. 这所有结果都体现出原创的 VPMR 算法在 SOG 和 SOE 估计方面有着极其优秀的效果.

全文总结

本文研发了一种可以做到高精度 SOG/SOE 估计的算法，并命名为 VPMR 算法. 该算法首先利用 VP 和得到基础的 SOG/SOE, 之后应用 MR 技术进行约简，得到项数减少系数降低的结果. 通过与历史上的 SOG/SOE 方法进行比较，可以发现 VPMR 算法在精度，收敛速度，核独立特点以及带宽控制上都有明显的优越性. VPMR 算法中最大带宽的可控性保证了该算法的效率. 并且该算法对并行化很友好，以后可以开发相应的并行化算法来对展开过程予以加速.

利用 VPMR 算法得到的高度 SOG/SOE 估计有许多用途，文中介绍了它在时间卷积积分，时间卷积积分方程上面的应用，并给出了详细的算例予以佐证. 不同的数值结果表明了它的效率、精度和普遍性，证明了在许多问题中潜在应用的吸引力. 由于奇异函数的特殊性，我们无法在奇点处对奇异目标函数进行 SOG/SOE 展开. 于是采用奇点平移截断来消除奇点带来的影响，这种截断可以把目标函数分为奇异与非奇异两部分. 非奇异部分正常使用 VPMR 算法而奇异部分采用多项式插值来计算. 由于截断点距离奇异点较近，即使采用多项式插值其运算消耗也可以在一个可接受范围内. 并且这种耦合可以使得数值格式可以以一种递推形式表达出来，不仅降低了运算复杂度，更缩小了所需的数据存储空间. 在卷积积分计算与线性卷积积分方程的求解中，由于递推格式可以完全消去前一时间步上数值解的误差，因而还可以保证误差不会随着时间推移而积累.

VPMR 算法的一个重要作用是可以与其它算法相耦合，例如快速高斯变换. VPMR 算法的出现使得 FGT 算法突破了势函数原有的仅可以为高斯函数的限制，从而可以以线性复杂度解决一般核函数的问题. 并且 VPMR 算法结果的带宽可控，又解决了 FGT 算法所面对的误差与带宽高度关联的问题. 尽管这种耦合目前无法解决 FGT 算法自身所具备的例如无法处理高维问题，网格利用率可能过低的问题，但是可以将 VPMR 算法与其它改进过的 FGT 算法，如 IFGT，树结构 FGT 等线性算法相耦合. 从而达到既拓宽了原算法的适用范围，又保证了其原有的优点.

采用 VPMR 算法耦合的耦合算法的误差往往由两部分构成，一是 VPMR 算法产生 SOG(SOE) 逼近所导致的误差，二是原算法的误差. 在数值实验中发现，前者的误差只有在步长极小等情况才会成为总误差的主要作用. 因而 VPMR 算法的高精度保证了其可以在与其它算法耦合时不会引起误差干扰. 至于运算复杂度方面，由于 VPMR 算法在产生 SOG/-SOE 展开是可以认为是预计算过程，不统计其计算消耗，耦合算法的复杂度往往是 P 个独立原算法的运算复杂度之和，其中 P 为 SOG(SOE) 的项数. 由于项数的数量级远低于原算法运算复杂度的数量级，可以认为 VPMR 算法的耦合不会引起复杂度的变化. 又因为这种耦合往往导致需要进行 P 次原算法的计算，采用并行优化可以大幅提高耦合效率. 从而论证了 VPMR 算法具有并行程度优秀的特点.

至于 VPMR 算法进一步的研究，一是可以从 SOG/SOE 的实际用途入手，探究 VPMR 算法与其它成熟算法的耦合，并与原有经典算法进行比较，凸显该算法的优越性. 二是可以从并行优化入手，减少算法运算时间，充分利用计算资源. 三是可以往深层次探究 VP 和的更优形式构造与 MR 技术的优化. 这些都将会是 VPMR 算法提升的空间.

参考文献

[1] Beylkin G, Kurcz C, Monzón L. Fast convolution with the free space Helmholtz Green's function[J]. Journal of Computational Physics, 2009, 228(8): 2770-2791.

[2] Cerioni A, Genovese L, Mirone A, et al. Efficient and accurate solver of the three-dimensional screened and unscreened Poisson's equation with generic boundary conditions[J]. The Journal of Chemical Physics, 2012, 137(13): 134108.

[3] Greengard L, Jiang S, Zhang Y. The anisotropic truncated kernel method for convolution with free-space Green's functions[J]. SIAM Journal on Scientific Computing, 2018, 40(6): A3733-A3754.

[4] Cheng H, Greengard L, Rokhlin V. A fast adaptive multipole algorithm in three dimensions[J]. Journal of Computational Physics, 1999, 155(2): 468-498.

[5] Yarvin N, Rokhlin V. An improved fast multipole algorithm for potential fields on the line[J]. SIAM Journal on Numerical Analysis, 1999, 36(2): 629-666.

[6] Alpert B, Greengard L, Hagstrom T. Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation[J]. SIAM Journal on Numerical Analysis, 2000, 37(4): 1138-1164.

[7] Jiang S, Greengard L. Fast evaluation of nonreflecting boundary conditions for the Schrödinger equation in one dimension[J]. Computers & Mathematics with Applications, 2004, 47(6-7): 955-966.

[8] Jiang S, Greengard L. Efficient representation of nonreflecting boundary conditions for the time-dependent Schrödinger equation in two dimensions[J]. Communications on Pure and Applied Mathematics, 2008, 61(2): 261-288.

[9] Lubich C, Schädle A. Fast convolution for nonreflecting boundary conditions[J]. SIAM Journal on Scientific Computing, 2002, 24(1): 161-182.

[10] Chen J, Wang L, Anitescu M. A fast summation tree code for Matérn kernel[J]. SIAM Journal on Scientific Computing, 2014, 36(1): A289-A309.

[11] Beylkin G, Monzón L. On approximation of functions by exponential sums[J]. Applied and Computational Harmonic Analysis, 2005, 19(1): 17-48.

[12] Beylkin G, Monzón L. Approximation by exponential sums revisited[J]. Applied and Computational Harmonic Analysis, 2010, 28(2): 131-149.

[13] Braess D. Asymptotics for the approximation of wave functions by exponential sums[J]. Journal of Approximation Theory, 1995, 83(1): 93-103.

[14] Braess D, Hackbusch W. On the efficient computation of high-dimensional integrals and the approximation by exponential sums[G]. in: Multiscale, Nonlinear and Adaptive Approximation. Springer, 2009: 39-74.

[15] Braess D, Hackbusch W. Approximation of $1/x$ by exponential sums in $[1, \infty)$[J]. IMA Journal of Numerical Analysis, 2005, 25(4): 685-697.

[16] Evans J W, Gragg W B, LeVeque R J. On least squares exponential sum approximation with positive coefficients[J]. Mathematics of Computation, 1980, 34(149): 203-211.

[17] Rodŕıguez A F, de SANTIAGO R L., Guillén E L, et al. Coding Prony's method in MATLAB and applying it to biomedical signal filtering[J]. BMC bioinformatics, 2018, 19(1): 1-14.

[18] Gonchar A A, Rakhmanov E A. Equilibrium distributions and degree of rational approximation of analytic functions[J]. Mathematics of the USSR-Sbornik, 1989, 62(2): 305.

[19] Kammler D W. Least squares approximation of completely monotonic functions by sums of exponentials[J]. SIAM Journal on Numerical Analysis, 1979, 16(5): 801-818.

[20] Varah J M. On fitting exponentials by nonlinear least squares[J]. SIAM Journal on Scientific and Statistical Computing, 1985, 6(1): 30-44.

[21] Wiscombe W J, Evans J W. Exponential-sum fitting of radiative transmission functions[J]. Journal of Computational Physics, 1977, 24(4): 416-444.

[22] Glover K. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds[J]. International Journal of Control, 1984, 39(6): 1115-1193.

[23] Greengard L, Strain J. The fast Gauss transform[J]. SIAM Journal on Scientific and Statistical Computing, 1991, 12(1): 79-94.

[24] Roussos G, Baxter B J. Rapid evaluation of radial basis functions[J]. Journal of Computational and Applied Mathematics, 2005, 180(1): 51-70.

[25] De La Vallée-Poussin C J. Leçons sur l'approximation des fonctions d'une variable réelle[M]. Paris, 1919.

[26] Natanson I P. Constructive Function Theory[M]. Ungar, 1964.

[27] Jiang S. A fast Gauss transform in one dimension using sum-of-exponentials approximations[J]. ArXiv: 1909.09825, 2019.

[28] Liang J, Gao Z, Xu Z. A kernel-independent sum-of-Gaussians method by de la Vallée-Poussin sums[J]. Advances in Applied Mathematics and Mechanics, in press.

[29] Albtoush R, Al-Khaled K. Approximation of periodic functions by Vallée Poussin sums[J]. Hokkaido Mathematical Journal, 2001, 30(2): 269-282.

[30] Boyer R P, Goh W M Y. Generalized Gibbs phenomenon for Fourier partial sums and de la Vallée-Poussin sums[J]. Journal of Applied Mathematics and Computing, 2011, 37(1-2): 421-442.

[31] Moore B. Principal component analysis in linear systems: controllability, observability, and model reduction[J]. IEEE Transactions on Automatic Control, 1981, 26(1): 17-32.

[32] Xu K, Jiang S. A bootstrap method for sum-of-poles approximations[J]. Journal of Scientific Computing, 2013, 55(1): 16-39.

[33] Barrowes B. Multiple Precision Toolbox for MATLAB[J]. MATLAB Central File Exchange, Retrieved August 10, 2020.

[34] Spivak M, Veerapaneni S K, Greengard L. The fast generalized Gauss transform[J]. SIAM Journal on Scientific Computing, 2010, 32(5): 3092-3107.

[35] López-Fernández M, Palencia C, Schädle A. A spectral order method for inverting sectorial Laplace transforms[J]. SIAM Journal on Numerical Analysis, 2006, 44(3): 1332-1350.

[36] Trefethen L N, Weideman J A C, Schmelzer T. Talbot quadratures and rational approximations[J]. BIT Numerical Mathematics, 2006, 46(3): 653-670.

[37] Talbot A. The accurate numerical inversion of Laplace transforms[J]. IMA Journal of Applied Mathematics, 1979, 23(1): 97-120.

[38] Weideman J, Trefethen L. Parabolic and hyperbolic contours for computing the Bromwich integral[J]. Mathematics of Computation, 2007, 76(259): 1341-1356.

[39] Weideman J. Improved contour integral methods for parabolic PDEs[J]. IMA Journal of Numerical Analysis, 2010, 30(1): 334-350.

[40] Gao Z, Liang J, Xu Z. Kernel-independent sum-of-exponentials with application to convolution quadrature[J]. ArXiv:2012.13477, preprint.

[41] Lopez-Marcos M. A difference scheme for a nonlinear partial integro-differential equation[J]. SIAM Journal on Numerical Analysis, 1990, 27(1): 20-31.

[42] Sanz-Serna J M. A numerical method for a partial integro-differential equation[J]. SIAM Journal on Numerical Analysis, 1988, 25(2): 319-327.

[43] Cao J, Xu C. A high order schema for the numerical solution of the fractional ordinary differential equations[J]. Journal of Computational Physics, 2013, 238: 154-168.

[44] Cuesta E, Palencia C. A fractional trapezoidal rule for integro-differential equations of fractional order in Banach spaces[J]. Applied Numerical Mathematics, 2003, 45(2-3): 139-159.

[45] Diethelm K, Ford N J. Analysis of fractional differential equations[J]. Journal of Mathematical Analysis and Applications, 2002, 265(2): 229-248.

[46] Zakeri G A, Navab M. Sinc collocation approximation of non-smooth solution of a nonlinear weakly singular Volterra integral equation[J]. Journal of Computational Physics, 2010, 229(18): 6548-6557.

[47] Lubich C. Fractional linear multistep methods for Abel-Volterra integral equations of the second kind[J]. Mathematics of Computation, 1985, 45(172): 463-469.

[48] Miller R K. Volterra integral equations in a Banach space[J]. Funkcial. Ekvac, 1975, 18(2): 163-193.

[49] Mohammadi F. A wavelet-based computational method for solving stochastic Itô – Volterra integral equations[J]. Journal of Computational Physics, 2015, 298: 254-265.

[50] Lubich C. Convolution quadrature revisited[J]. BIT Numerical Mathematics, 2004, 44(3): 503-514.

[51] Schädle A, López-Fernández M, Lubich C. Fast and oblivious convolution quadrature[J]. SIAM Journal on Scientific Computing, 2006, 28(2): 421-438.

[52] López-Fernández M, Sauter S. Generalized convolution quadrature with variable time stepping[J]. IMA Journal of Numerical Analysis, 2013, 33(4): 1156-1175.

[53] March W B, Xiao B, Tharakan S, et al. A kernel-independent FMM in general dimensions[C]. in: SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2015: 1-12.

[54] Lubich C. Convolution quadrature and discretized operational calculus. I[J]. Numerische Mathematik, 1988, 52(2): 129-145.

[55] Lubich C. Convolution quadrature and discretized operational calculus. II[J]. Numerische Mathematik, 1988, 52(4): 413-425.

[56] Lubich C, Ostermann A. Runge-Kutta methods for parabolic equations and convolution quadrature[J]. Mathematics of Computation, 1993, 60(201): 105-131.

[57] Butcher J C, Goodwin N. Numerical Methods for Ordinary Differential Equations[M]. Wiley Online Library, 2008.

[58] Brasey V, Hairer E. Half-explicit Runge–Kutta methods for differential-algebraic systems of index 2[J]. SIAM Journal on Numerical Analysis, 1993, 30(2): 538-552.

[59] Corduneanu C. Integral Equations and Stability of Feedback Systems[M]. Academic Press, 1973.

[60] Srivastava H M, Buschman R G. Theory and Applications of Convolution Integral equations[M]. Springer Science & Business Media, 2013.

[61] Tamarkin J D. On integrable solutions of Abel's integral equation[J]. Annals of Mathematics, 1930, 31(2): 219-229.

[62] Wanner G, Hairer E. Solving Ordinary Differential Equations[M]. Springer Berlin Heidelberg, 1996.

[63] Jaswon M A. Integral Equation Methods in Potential Theory and Elastostatics[Z]. 1977.

[64] Jiang S, Rokhlin V. Second kind integral equations for the classical potential theory on open surfaces II[J]. Journal of Computational Physics, 2004, 195(1): 1-16.

[65] Levinson N. A nonlinear Volterra equation arising in the theory of superfluidity[J]. Journal of Mathematical Analysis and Applications, 1960, 1(1): 1-11.

[66] Gray A G, Moore A W. N-body'problems in statistical learning[C]. in: Advances in neural information processing systems. 2001: 521-527.

[67] Hofmann T, Schölkopf B, Smola A J. Kernel methods in machine learning[J]. The Annals of Statistics, 2008: 1171-1220.

[68] Liang J, Liu P, Xu Z. A high-accurate fast Poisson solver based on harmonic surface mapping algorithm[J]. Commun. Comput. Phys., 2021, 30: 210-226.

[69] Wendland H. Scattered Data Approximation[M]. Cambridge University Press, 2004.

[70] Yang C, Duraiswami R, Davis L S. Efficient kernel machines using the improved fast Gauss transform[C]. in: Advances in Neural Information Processing Systems. 2005: 1561-1568.

[71] Zhao Q, Liang J, Xu Z. Harmonic surface mapping algorithm for fast electrostatic sums[J]. The Journal of Chemical Physics, 2018, 149(8): 084111.

[72] Carrier J, Greengard L, Rokhlin V. A fast adaptive multipole algorithm for particle simulations[J]. SIAM Journal on Scientific and Statistical Computing, 1988, 9(4): 669-686.

[73] Rokhlin V, Greengard L. A fast algorithm for particle simulations[J]. J. Comp. Phys, 1987, 73: 325-348.

[74] Cherrie J B, Beatson R K, Newsam G N. Fast Evaluation of Radial Basis Functions: Methods for Generalized Multiquadrics in $\mathbb{R}^n$[J]. SIAM Journal on Scientific Computing, 2012, 23(5): 1549-1571.

[75] Fong W, Darve E. The black-box fast multipole method[J]. Journal of Computational Physics, 2009, 228(23): 8712-8725.

[76] Yarvin N, Rokhlin V. An Improved Fast Multipole Algorithm for Potential Fields on the Line[J]. SIAM Journal on Numerical Analysis, 36(2): 629-666.

[77] Greengard L, Strain J. The fast Gauss Transform[J]. SIAM Journal on Scientific and Statistical Computing, 1991, 12(1): 79-94.

[78] Rayker V. The fast Gauss transform with all the proofs[J]. Dept. of Computer Science, University of Maryland at College Park, 2006.

[79] Wan X, Karniadakis G E. A sharp error estimate for the fast Gauss transform[J]. Journal of Computational Physics, 2006, 219(1): 7-12.

[80] Greengard L, Sun X. A new version of the fast Gauss transform[J]. Documenta Mathematica, 1998, 3: 575-584.

[81] Baxter B J C, Roussos G. A new error estimate of the fast Gauss transform[J]. SIAM Journal on Scientific Computing, 2002, 24(1): 257-259.

[82] Yang C, Duraiswami R, Gumerov N A, et al. Improved fast gauss transform and efficient kernel density estimation[C]. in: Null. 2003: 464.

[83] Lee D, Moore A W, Gray A G. Dual-tree fast Gauss transforms[C]. in: Advances in Neural Information Processing Systems. 2006: 747-754.

[84] Hu X G, Ho T S, Rabitz H. The collocation method based on a generalized inverse multiquadric basis for bound-state problems[J]. Computer Physics Communications, 1998, 113(2-3): 168-179.

[85] Scholkopf B, Kah-Kay Sung, Burges C J C, et al. Comparing support vector machines with Gaussian kernels to radial basis function classifiers[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2758-2765.

[86] Darden T, York D, Pedersen L. Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems[J]. The Journal of Chemical Physics, 1993, 98(12): 10089-10092.

[87] Ewald P P. Die Berechnung optischer und elektrostatischer Gitterpotentiale[J]. Annalen Der Physik, 1921, 369(3): 253-287.

[88] Jin S, Li L, Xu Z, et al. A random batch Ewald method for particle systems with Coulomb interactions[J]. SIAM, J. Sci. Comput., in press.

[89] Heiden. Analysis of Neural Networks[M]. Springer Science & Business Media, 2013.

致　谢

本研究及学位论文是在我的导师徐振礼教授的亲切关怀和悉心指导下完成的. 徐教授严谨的治学精神深深激励感染了我. 从课题的提出, 到瓶颈问题的解决, 再到最终落笔形成一篇文章, 无处不体现出徐教授精益求精严格严谨的特点. 无论是课题提出时的反复讨论, 还是解决问题时的头脑风暴, 亦或是徐教授一字一句斟酌初稿, 都让我意识到认真用心完成一篇原创文章的不易. 徐教授的谆谆教诲指引着我前行, 将来我也将会继续跟随徐教授, 不断在计算数学的领域奋进, 探究原子分子模拟的真理.

数学科学学院博士生梁久阳学长也是我在完成毕业论文过程中给了我许多指导的人. 在起初初入课题组, 对科研一无所知的我很快陷入了不知所措. 是梁学长不厌其烦地为我讲述科研的方法, 培养我的科研精神与习惯, 帮助我在科研道路上起步. "不要总是提出问题, 更要学会如何去解决问题, 至少要给出一个大致思路来. " 他的这番话始终指引着我在科研的海洋里探索, 激励我攻克了一个又一个难题. 梁学长在本篇毕业论文中也给出了许多建设性的指导意见, 并且给了我许多鼓励与劝勉.

感谢两位同门刘𠀤鸣与李水木. 他们为我提供的对于论文 LaTeX 模板帮助使我能够更快完成论文的编辑与写作. 也感谢制作这份 LaTeX 模板的同学们. 希望这份传承精神能够一直保持下去, 造福更多需要帮助的人.

另外, 我还要感谢致远创新研究中心 (ZIRC), 是它给了我一个良好的研究环境与一个资源平台. 有了 ZIRC, 我的论文得以更加顺利地完成. 最后我要感谢致远学院, 是它为我大学本科四年提供了一个非常优秀的平台, 使得我能够接触到许多像徐教授那样优秀的老师, 以及获得许多资源拓宽自己的眼界. 致远学院所渲染的学术氛围也使得我确定了走向科研这条道路. 夜航星空下, 有你不孤单.

在论文即将完成之际, 再次向徐教授, 梁学长等帮助过我, 指导过我的人们表示衷心的感谢.

感谢你们的帮助!

A FAST SUM-OF-GAUSSIAN METHOD

Approximation of interacting kernels by sum of Gaussians (SOG) is frequently required in many applications of scientific and engineering computing in order to construct efficient algorithms for kernel summation or convolution problems. In this paper, we propose a kernel-independent SOG method by introducing the de la Vallée-Poussin sum and Chebyshev polynomials, which is called VPMR. The SOG works for general interacting kernels and the lower bound of Gaussian bandwidths is tunable and thus the Gaussians can be easily summed by fast Gaussian algorithms. The number of Gaussians can be further reduced via the model reduction based on the balanced truncation based on the square root method. Numerical results on the accuracy and model reduction efficiency show attractive performance of the proposed method. Compared with the historical method, VPMR has the advantages of faster convergence speed and higher convergence precision. Both numerical experiments and theoretical results verify the advantages of this method. It cleverly solves the bandwidth problem which is difficult to deal with in SOG estimation process, and gives a scheme in analytic form. This is undoubtedly a breakthrough that has not been made in the historical method.

In order to promote this efficient original algorithm, we implemented VPMR on Matlab. Since the VPMR approach requires high-precision matrix manipulation, we employ the Multiple Precision Toolbox in order to implement the algorithm. These packages are used in both steps of the VP-sum and the model-reduction procedures. The computer code of the VPMR approach is released as open source, which is available at https://github.com/ZXGao97. The visual code of VPMR makes it more convenient for users to apply the algorithm.

In addition to generating SOG, VPMR can also generate sum of exponentials(SOE) with the same high precision. One of important applications of the SOE is to quickly approximate convolution quadrature. We consider the approximation of convolution quadrature between a given kernel $f(t)$ and smooth function $g(t)$ as follows,

$$y(t) = f * g = \int_0^t f(t-\tau)g(\tau)d\tau. \tag{6–20}$$

If the kernel function is non-singular, then VPMR can be used to get the SOE of the kernel function, so that the convolution quadrature problem can be decomposed into several ordinary differential equations to solve, and numerical solvers such as Runge-Kutta method can be used. If the kernel function is singular, we employ the SOE expansion for the finite part of the splitting convolution kernel such that the convolution quadrature can be solved as a system of ordinary differential equations due to the exponential kernels. The remaining part is explicitly approximated by employing the generalized Taylor expansion. The significant features of our algorithm are that the SOE method is efficient and accurate, and works for general kernels with controllable upperbound of positive exponents. We provide numerical analysis for the SOE-based convolution quadrature. Numerical results on different kernels, the convolution quadrature and integral equations demonstrate attractive performance of both accuracy and efficiency of the proposed method. The better precision and lower complexity illustrate the advantages and feasibility of VPMR. This coupling method makes it possible to quickly compute the convolution quadrature of some kernel functions which cannot

obtain the analytic form of Laplace transform. In addition, its high efficiency and high precision in the convolution quadrature equation shows that this method can also play an important role in the mainstream research of convolution quadrature.

Another application of VPMR is in combination with the fast Gauss transform(FGT). As one of the kernel-independent FGT based methods, the fast generalize Gauss transform which combines both the Hermite and plane-wave versions FGT is proved to be efficient in solving the diffusion problems like the unsteady Stokes flow. This method which requires the values of the Fourier transform of the kernel at the plane-wave discretization points focuses mostly on the acceleration of the translation operator (named S2W and W2L) and thus exchanges expansion cost for kernel-independent. The accerlate performance is limited especially when the kernel isn't a Gaussian type and both the sources and the targets are not belong to a tensor-product grid. As the author argued in, the original FGT is faster when the number of boxes is small (i.e., with a large bandwidth). Another methods reported in is worked when the kernel is radially symmetric and negative definite on all $\mathbb{R}^d$. Even leaving aside the difficulty to find such a Borel measure required in, unmanageable low bandwidth thanks to the discretization of a $[0, +\infty)$ integral in conjunction with the generalised Gauss quadrature rule will lead to costly translate consumption. We first construct an q-terms Gaussian approximation of the kernel by Vallée Poussin (VP) sums and variable substitution. All these terms of Gaussian remain an excellent bandwidth and thus can be evaluated by the FGT or its modified version within the small-scale boxes structure, we style it as the multipole fast Gauss transform. As an significant advantage, the prefactor in front of $O(N + M)$ is independent of q (i.e., the increasing of q will only impact the precompute time), which allows us to compute the whole terms with just one time of multipole FGT. Our numerical results compared with the previous tree code scheme show dependable accuracy and attractive performance. Our algorithm can be easily extended to solve high-dimensional problem by exchanging FGT to IFGT or other modified versions which aim to eliminate the curse of dimensionality.

Both the theoretical analysis and the numerical examples and the comparison with the historical methods show that the VPMR has the advantages of high precision, fast convergence, good coupling property and controllable bandwidth. This original algorithm, VPMR, fills a historical gap in this aspect of the problem. This breakthrough makes many excellent algorithms which are limited to exponential or Gaussian functions can be applied to general functions, thus greatly promoting the development of solving the corresponding problems of these algorithms.

As for the further research of VPMR, the first one is to start from the practical use of SOG/SOE, to explore the coupling of VPMR and other mature algorithms, and compare with the original classical algorithm, highlighting the superiority of VPMR. Second, we can start with parallel optimization to reduce the time of algorithm operation and make full use of computing resources. Thirdly, we can further explore the better form structure of VP sum and the optimization of Model Reduction. These extended studies will improve all aspects of the VPMR, making it more competitive in related problems and more adaptable in coupling problems.