浙江大學

本 科 毕 业 论 文 (设 计)

题目	谱方法某些结果的数值验证
姓名与学号	3080100913 李言迪
指导教师	叶兴德
年级与专业	2008 数学与应用数学
所在学院	理学院

浙江大学本科生毕业论文（设计）诚信承诺书

1. 本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2. 本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

毕业论文（设计）作者签名：　　李言迪

二〇一二年五月二十二日

本科生毕业论文（设计）任务书

一、 题目：谱方法某些结果的数值验证

二、 指导教师对毕业论文（设计）的进度安排及任务要求：

时间	任务	要求
2012年02月13日—2012年03月02日	翻译外文文献	完成翻译
2012年03月05日—2012年03月16日	阅读相关文献	完成文献综述
2012年03月19日—2012年04月01日	准备开题报告	完成开题报告
2012年04月05日—2012年05月04日	程序设计，数值实验	完成数值实验
2012年05月07日—2012年05月30日	毕业论文撰写	完成毕业论文
2012年06月初	论文答辩	完成论文答辩

起讫日期 2012 年 02 月 13 日至 2012 年 06 月 日.

指导教师（签名） 职称

三、 系或研究所审核意见:

同意该计划。

负责人（签名）

年 月 日

目录

摘要

在应用 Chebyshev-谱 tau 方法求解常微分 Helmholtz 方程的边值问题时，文献中存在着两种处理方法：既可以使得未知量是待求函数的谱系数，也可是以使未知量为待求函数二阶导函数的谱系数（2 阶谱）。现有的证明指出，以高阶谱递推低阶谱的“谱积分”过程，误差放大不超过 2.4 倍；而与之相对的“谱微分”过程会放大误差达 $O(N^2)$。据此，有学者猜测，使用“谱微分”的算法会在计算 1 阶谱系数时表现劣势。针对这一问题，本文进行了数值实验，然而得到的结果并没有支持该论断; 在实际计算中, 反而使用“谱微分”的求解顺序表现更好。

关键词: Chebyshev-tau 方法，谱方法，Helmholz 方程

Abstract

On the subject of how to apply Chebyshev-tau spectral method to one-dimensional Helmholtz boundary problem, we find two main strategies in the literature. The difference lies in the choice of the premier target of solution: whether we first solve for the spectral coefficients(SCs) of the unknown function, or of its second-order derivative (which we denote as the 2-order SCs). Study has shown that the process of 'spectral integrating' which transforms high-order SCs to low-order SCs amplifies initial errors by less than a factor of 2.4, whereas the reverse process--'spectral differentiating'--amplifies errors by $O(N^2)$. Thus, some scholar argues that one should expect worse accuracy in the 1-order SCs from the strategy with differentiating. In order to validate such proposition, we conduct a series of numerical experiments. It turns out, however, that the strategy with differentiating performs equally well, if not better, in real computation settings. Analysis is also given regarding this apparent contradiction.

Keywords: Chebyshev-tau approximation, Spectral method, Helmholtz BV problem

1 概述

谱方法求解微分方程的原型是 Fourier 方法，把未知函数作级数展开来表示，然后把谱系数作为要求解的对象。Fourier 方法适用于周期性问题；而对于非周期性问题，一般采用与之关系紧密的 Chebyshev 方法类，包括 Chebyshev-tau 方法和配置法（collocation）。本文采用 Chebyshev-tau 法来处理。在具体实现时，文献中存在两种求解顺序：通过利用 Chebyshev 多项式各阶谱系数间的递推关系，我们可以使得未知量为未知函数 $u(x)$ 的 Chebyshev 展式的系数（0 阶谱系数）；或者是其 2 阶导数 $u_N''(x)$ 的 Chebyshev 展式系数（2 阶谱系数）。对于这两种实现，Greengard[2] 认为，后一种顺序会在求解未知函数的 1 阶导数和 2 阶导数上比前者更加精确。这是因为他曾证明出，以 2 阶谱系数还原 1 阶谱系数的递推过程，误差放大不超过 2.4 倍；而用 0 阶谱系数计算 1 阶谱系数的递推过程，误差会放大 $O(N^2)$ 倍（这里的 N 指的是 N 阶截断的 Chebyshev 展式逼近）。针对这一论断，本文通过简单的一维 Helmholtz 方程的 Dirichlet 问题的一些实例，进行了数值实验。然而我们得到的结果并没有支持 Greengard 一边倒的论断；在实际计算中，反而第一种求解顺序的表现更好。

本文在第2节会介绍 Chebyshev 方法的基本概念和 Chebyshev-tau 逼近的技术细节。接下来在第3节中，给出上述的两种求解过程。我们将看到，这两种方法的核心都需要求解一个“拟三对角方程组”，为了保证求解时的效率，本文设计了相应的算法，在第4节中给出。然后，第5节是数值实验部分，本文以三个实例对比了两种方法的求解结果，并在第6节分析了 Greengard 论断失效的可能原因。最后，程序在附录A附上。

2 Chebyshev-tau 逼近方法

2.1 Chebyshev 多项式

记 k 次的第一类 Chebyshev 多项式为 $T_k(x)$, 它们定义在 $x \in [-1,1]$，满足

$$T_k(x) = \cos(k\cos^{-1}x), k = 0, 1, 2, ... \tag{2.1.1}$$

很明显，其值域为 $|T_k(x)| \leq 1$。对 Chebyshev 多项式来说，在理论和计算中更多用到下面的等价形式，即设 $x = \cos z$，有

$$T_k(z) = \cos kz \tag{2.1.2}$$

由该式不难写出 Chebyshev 多项式的前几项为

$$T_0 = 1,\ T_1 = \cos z = x,\ T_2 = \cos 2z = 2\cos^2 z - 1 = 2x^2 - 1, ...$$

2.2 两点边值问题

我们将应用 Chebyshev-tau 方法考虑如下的 Helmholtz 方程的 Dirichlet 问题，

$$\begin{cases} -\dfrac{d^2u}{dx^2} + \lambda u = f, & x \in (-1,1) \\ u(-1) = a, \quad u(1) = b. \end{cases} \tag{2.2.1}$$

Chebyshev-tau 方法是这样进行的。即首先对微分方程的未知函数 $u(x)$ 和右端非齐次项 $f(x)$，作 N 阶截断的 Chebyshev 展开式，即

$$u_N(x) = \sum_{k=0}^{N} \hat{u}_k T_k(x), \quad f_N(x) = \sum_{k=0}^{N} \hat{f}_k T_k(x),$$

其中,展开式的系数我们称为(0 阶)谱系数。

然后,Chebyshev 多项式构成了一组单位正交基。我们让 $u(x)$ 的逼近 $u_N(x)$ 在弱的意义下满足微分方程,意即在不超过 $N-2$[1]次的 Chebyshev 多项式构成的子空间中成立,即

$$\int_{-1}^{1}\left[-\frac{d^2u_N}{dx^2}+\lambda u_N-f\right]\cdot T_k(x)w(x)dx=0,\quad k=0,1,\ldots,N-2, \tag{2.2.2}$$

其中 $w(x)=(1-x^2)^{-1/2}$ 为内积的权函数。若记 $\hat{u}_k^{(2)}$ 表示 $u_N''(x)$ 的谱系数,即 2 阶谱系数。则上式由 $\{T_k\}_{k=0,\ldots,N}$ 的单位正交性可以写成

$$-\hat{u}_k^{(2)}+\lambda\hat{u}_k=\hat{f}_k,\quad k=0,1,\ldots,N-2, \tag{2.2.3}$$

最后,由 Chebyshev 多项式性质 $T_k(1)=1$,$T_k(-1)=(-1)^k$,得到边界条件表达为

$$\sum_{k=0}^{N}\hat{u}_k=a,\quad \sum_{k=0}^{N}(-1)^k\hat{u}_k=b. \tag{2.2.4}$$

通过 (2.2.3) 和 (2.2.4) 两式求解弱解 $u_N(x)$ 是 Chebyshev-tau 方法下面需要求解的问题。

2.3 Chebyshev 谱系数的数值计算

理论上,一个在 $x\in[-1,1]$ 上有定义的任意函数 $g(x)$,他的 Chebyshev 展式的系数由下面的内积式准确给出

$$\hat{g}_k^*=(g,T_k)_w=\frac{2}{\pi c_k}\int_{-1}^{1}(gT_k)\,wdx,\quad c_k=\begin{cases}2 & \text{当}k=0\text{时}\\ 1 & \text{当}k\geq 1\text{时}\end{cases}. \tag{2.3.1}$$

数值计算此积分,则对用 Gauss-Lobatto 点 $x_k=\cos\pi k/N, k=0,\ldots,N$ 作为插值格点[2],将 Gauss 数值积分公式

$$\int_{-1}^{1}pwdx\cong\frac{\pi}{N}\sum_{k=0}^{N}\frac{p(x_k)}{\bar{c}_k},\quad \bar{c}_k=\begin{cases}2 & \text{当}k=0\text{时}\\ 1 & \text{当}1\leq k\leq N-1\text{时}\\ 2 & \text{当}k=N\text{时}\end{cases},$$

应用于 (2.3.1),记 $g_i=g(x_i)$,得到所谓离散的 Chebyshev 变换,

$$\hat{g}_k=\frac{2}{\bar{c}_kN}\sum_{i=0}^{N}\frac{1}{\bar{c}_i}g_i\cos\frac{k\pi i}{N},k=0,...,N \tag{2.3.2}$$

[1]考虑到我们仅需要 $N+1$ 个方程来确定 $N+1$ 个待定的谱系数 $\{\hat{u}_k\}_{k=0,\ldots,N}$,因此,(2.2.2),(2.2.3) 中的 k 只要在 $0,\ldots,N-2$ 中成立,即构成一个恰好适定的线性方程组。

[2]谱系数可以由插值点处的函数值线性组合得到。相较于 Gauss 点 $x_k=\cos(k+1/2)\pi/N, k=0,\ldots,N-1$,Gauss-Lobatto 点包含了边界点 $x=\pm1$,从而谱系数可以反应出边界点上函数值的信息,故宜选用此种插值点来作积分。

我们后面将应用此变换于微分方程 (2.2.1) 的右端函数以及数值实验的真实解上。可以证明这种近似带来的误差如下 (见 [4])

$$\hat{g}_k = \hat{g}_k^* + \frac{1}{\bar{c}_k}\left[\sum_{\substack{m=1\\2mN>N-k}}^{\infty} \hat{g}^*_{k+2mN} + \sum_{\substack{m=1\\2mN>N+k}}^{\infty} \hat{g}^*_{k-2mN}\right].$$

2.4 谱系数的递推关系

我们需要知道 $u_N(x)$ 的导函数在 Chebyshev 基下展开的谱系数 (高阶谱系数),并将看到,高阶谱系数可以由原函数的谱系数以线性组合的方式给出。设

$$u'_N(x) = \sum_{k=0}^{N} \hat{u}_k T'_k(x) = \sum_{k=0}^{N} \hat{u}_k^{(1)} T_k(x) \tag{2.4.1}$$

利用三角函数公式得到的递推关系 $\frac{T'_{k+1}}{k+1} - \frac{T'_{k-1}}{k-1} = 2T_k$,可以得到

$$T'_k(x) = 2k \sum_{n=0}^{[(k-1)/2]} \frac{1}{c_{k-1-2n}} T_{k-1-2n}(x) \tag{2.4.2}$$

把 (2.4.2) 代入 (2.4.1) 可以导出一个 1 阶谱系数 $\hat{u}_k^{(1)}$ 的表达式:

$$\hat{u}_k^{(1)} = \frac{2}{c_k} \sum_{\substack{p=k+1\\p:=p+2}}^{N} p\hat{u}_p, k = 0, \ldots, N-1, \text{且有} \hat{u}_N^{(1)} = 0. \tag{2.4.3}$$

一般的 p 阶与 $p-1$ 阶谱系数也存在同样的递推关系。这便是由低阶谱系数获得高阶谱系数的“谱微分运算”。该关系也可以写成所谓微分矩阵的形式 $\hat{U}^{(1)} = \hat{D}\hat{U}$。递推用两次,也可以得到 2 阶谱系数计算如下

$$\hat{u}_k^{(2)} = \frac{1}{c_k} \sum_{\substack{p=k+2\\p:=p+2}}^{N} p(p^2-k^2)\hat{u}_p, k = 0, \ldots, N-2, \text{且有} \hat{u}_N^{(2)} = \hat{u}_{N-1}^{(2)} = 0. \tag{2.4.4}$$

同样的公式 (2.4.3),隔项作差,又可以得到由高阶谱系数获得低阶谱系数的“谱积分运算”。

$$\hat{u}_k^{(p-1)} = \frac{1}{2k}\left(c_{k-1}\hat{u}_{k-1}^{(p)} - \hat{u}_{k+1}^{(p)}\right), \quad k \geq 1 \tag{2.4.5}$$

且有末端项初值置为:1 阶谱系数时,

$$\hat{u}_N^{(1)} = 0,\ \hat{u}_{N-1}^{(1)} = 2N\hat{u}_N. \tag{2.4.6}$$

2 阶谱系数时,

$$\hat{u}_N^{(2)} = \hat{u}_{N-1}^{(2)} = 0,\ \hat{u}_{N-2}^{(2)} = 2(N-1)\hat{u}_{N-1}^{(1)} = 4N(N-1)\hat{u}_N. \tag{2.4.7}$$

3 两种求解过程

3.1 方法 I:化未知量为 0 阶谱系数

虽然我们可以利用 (2.4.4) 来直接消去 (2.2.3) 中的 2 阶谱系数,但是这种方式构成的线性系统一方面需要 $O(N^2)$ 阶的较多运算量,一方面也很可能是比较奇异的(Greengard[2])。同样是保留 0 阶谱,一种更有效率的方式如 Canuto 等人 [1] 所述[3]。利用 $q=2$ 时的三项递推关系 (2.4.5)

$$2k\hat{u}_k^{(1)} = c_{k-1}\hat{u}_{k-1}^{(2)} - \hat{u}_{k+1}^{(2)}, \quad k = 1, \ldots, N-1$$

将 (2.2.3)(此式对 $k = 0, \ldots, N-2$ 成立)代入得到

$$2k\hat{u}_k^{(1)} = c_{k-1}\left(-\hat{f}_{k-1} + \lambda\hat{u}_{k-1}\right) - \left(-\hat{f}_{k+1} + \lambda\hat{u}_{k+1}\right), \quad k = 1, \ldots, N-3.$$

再代入 $q=1$ 时的递推 (2.4.5),消去 1 阶谱系数得到

$$\begin{aligned} 2k\hat{u}_k &= \frac{c_{k-1}}{2(k-1)}\left[c_{k-2}(-\hat{f}_{k-2} + \lambda\hat{u}_{k-2}) - (-\hat{f}_k + \lambda\hat{u}_k)\right] \\ &\quad -\frac{1}{2(k+1)}\left[(-\hat{f}_k + \lambda\hat{u}_k) - (-\hat{f}_{k+2} + \lambda\hat{u}_{k+2})\right], \quad k = 2, \ldots, N-4. \end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned} &\frac{c_{k-2}\lambda}{4k(k-1)}\hat{u}_{k-2} + \left(-1 - \frac{\lambda}{2(k^2-1)}\right)\hat{u}_k + \frac{\lambda}{4k(k+1)}\hat{u}_{k+2} \\ = \;& \frac{c_{k-2}}{4k(k-1)}\hat{f}_{k-2} + \left(-\frac{1}{2(k^2-1)}\right)\hat{f}_k + \frac{1}{4k(k+1)}\hat{f}_{k+2}, \quad k = 2, \ldots, N-4. \end{aligned} \tag{3.1.1}$$

下面将扩展 (3.1.1) 的结果到 $k = N-3, \ldots, N$。首先对于 $k = N-3$ 的情况,同样的步骤

$$\begin{aligned} 2(N-3)\hat{u}_{N-3} &= \hat{u}_{N-4}^{(1)} - \hat{u}_{N-2}^{(1)} \\ &= \frac{1}{2(N-4)}\left[\hat{u}_{N-5}^{(2)} - \hat{u}_{N-3}^{(2)}\right] - \frac{1}{2(N-2)}\left[\hat{u}_{N-3}^{(2)} - \hat{u}_{N-1}^{(2)}\right] \end{aligned}$$

由(2.4.7)知其中$\hat{u}_{N-1}^{(2)} = 0$,

$$\begin{aligned} &= \frac{1}{2(N-4)}\left[\left(-\hat{f}_{N-5} + \lambda\hat{u}_{N-5}\right) - \left(-\hat{f}_{N-3} + \lambda\hat{u}_{N-3}\right)\right] \\ &\quad -\frac{1}{2(N-2)}\left[-\hat{f}_{N-3} + \lambda\hat{u}_{N-3}\right] \end{aligned}$$

$\Leftrightarrow$ 形如

$$\begin{aligned} &(\ldots)\,\hat{u}_{N-5} + (\ldots)\,\hat{u}_{N-3} \\ = \;& (\ldots)\,\hat{f}_{N-5} + (\ldots)\,\hat{f}_{N-3} \end{aligned}$$

对比 (3.1.1),仅需去掉 $\hat{u}_{k+2}$ 与 $\hat{f}_{k+2}$ 项就符合了。$k = N-2$ 的结论是类似的。

[3]该书中有一些错误,按此处所写为准。

对于 $k=N-1$ 的情况，

$$\begin{aligned} 2(N-1)\hat{u}_{N-1} &= \hat{u}^{(1)}_{N-2}-\hat{u}^{(1)}_{N}=\frac{\hat{u}^{(2)}_{N-3}-\hat{u}^{(2)}_{N-1}}{2(N-2)}-\hat{u}^{(1)}_{N} \\ \text{由(2.4.6)(2.4.7)，} &= \frac{-\hat{f}_{N-3}+\lambda\hat{u}_{N-3}}{2(N-2)} \end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned} & \tfrac{\lambda}{4(N-1)(N-2)}\hat{u}_{N-3}+(-1)\hat{u}_{N-1} \\ = \quad & \tfrac{1}{4(N-1)(N-2)}\hat{f}_{N-3} \end{aligned}$$

$k=N$ 的结论是类似的。

综合起来，(3.1.1) 和 $k=N-3,\dots,N$ 的情况，可以整合成下面的式子

$$\begin{aligned} & \frac{c_{k-2}\lambda}{4k(k-1)}\hat{u}_{k-2}+\left(-1-\frac{\lambda\beta_k}{2(k^2-1)}\right)\hat{u}_k+\frac{\lambda\beta_{k+2}}{4k(k+1)}\hat{u}_{k+2} \qquad (3.1.2) \\ = \quad & \frac{c_{k-2}}{4k(k-1)}\hat{f}_{k-2}+\left(-\frac{\beta_k}{2(k^2-1)}\right)\hat{f}_k+\frac{\beta_{k+2}}{4k(k+1)}\hat{f}_{k+2}, \quad k=2,\dots,N. \end{aligned}$$

其中

$$\beta_k=\begin{cases}1, & 0\le k\le N-2,\\ 0, & k>N-2.\end{cases}$$

(3.1.2) 加上边界条件 (2.2.4) 的两个方程，正好构成 $N+1$ 个方程，来求解 $N+1$ 个变量 $\{\hat{u}_k\}_{k=0,\dots,N}$。注意到 (3.1.2) 中的奇、偶项可以分开，而边界条件 (2.2.4) 也可以化成奇、偶项分开的形式如下：

$$\sum_{\substack{k=0\\ k\text{为偶数}}}^{N}\hat{u}_k=\frac{b+a}{2}, \qquad \sum_{\substack{k=0\\ k\text{为奇数}}}^{N}\hat{u}_k=\frac{b-a}{2}. \qquad (3.1.3)$$

因此我们分开奇、偶项的谱系数，分别求解。线性系统是这样的：

$$\begin{pmatrix} 1 & 1 & 1 & \dots & & & 1\\ * & * & * & & & & \\ & * & * & * & & & \\ & & & \dots & & & \\ & & & * & * & * & \\ & & & & * & * & \\ & & & & & * & * \end{pmatrix}\begin{pmatrix}\hat{u}_0\\ \hat{u}_2\\ \hat{u}_4\\ \vdots\\ \hat{u}_{N\text{偶}-4}\\ \hat{u}_{N\text{偶}-2}\\ \hat{u}_{N\text{偶}}\end{pmatrix}=\begin{pmatrix}\frac{b+a}{2}\\ \hat{g}_2\\ \hat{g}_4\\ \vdots\\ \hat{g}_{N\text{偶}-4}\\ \hat{g}_{N\text{偶}-2}\\ \hat{g}_{N\text{偶}}\end{pmatrix},\quad \begin{pmatrix} 1 & 1 & 1 & \dots & & & 1\\ * & * & * & & & & \\ & * & * & * & & & \\ & & & \dots & & & \\ & & & * & * & * & \\ & & & & * & * & \\ & & & & & * & * \end{pmatrix}\begin{pmatrix}\hat{u}_1\\ \hat{u}_3\\ \hat{u}_5\\ \vdots\\ \hat{u}_{N\text{奇}-4}\\ \hat{u}_{N\text{奇}-2}\\ \hat{u}_{N\text{奇}}\end{pmatrix}=\begin{pmatrix}\frac{b-a}{2}\\ \hat{g}_3\\ \hat{g}_5\\ \vdots\\ \hat{g}_{N\text{奇}-4}\\ \hat{g}_{N\text{奇}-2}\\ \hat{g}_{N\text{奇}}\end{pmatrix}, \qquad (3.1.4)$$

其中，$N_{\text{偶}}$（$N_{\text{奇}}$）表示最大偶（奇）数项，$*$ 表示系数矩阵中非零项，$\hat{g}_k$ 是 (3.1.2) 的相应 k 时的右端项。我们称系数矩阵的形式为“拟三对角矩阵”，它与一般三对角矩阵不同之处仅在于首行。后面将看到，第二种方法也会化归到一个“拟三对角矩阵”问题上，因此我们放弃运算量为 $O(N^3)$ 的 Gauss 消元法，而设计专门的类似于“追赶法”的 $O(N)$ 的算法来求解，结果在第4节中给出。

小结一下,方法 I 求解 Helmholtz 方程的 Dirichlet 问题 (2.2.1) 的步骤是:首先,确定要作的 Chebyshev 截断的项数 N,对右端函数 f 作相应的离散 Chebyshev 变换 (2.3.2),得到系数 $\{\hat{f}_k\}_{k=0,\dots,N}$。其次,求解拟三对角方程组 (3.1.4),得到 0 阶谱系数 $\{\hat{u}_k\}_{k=0,\dots,N}$。最后,利用谱微分运算 (2.4.3)(2.4.4) 由 $\{\hat{u}_k\}_{k=0,\dots,N}$ 来计算 1 阶谱 $\{\hat{u}_k^{(1)}\}_{k=0,\dots,N}$ 与 2 阶谱 $\{\hat{u}_k^{(2)}\}_{k=0,\dots,N}$,并可以进而通过这些谱系数计算函数、1 阶 / 2 阶导函数的函数值。

3.2 方法 II:化未知量为 2 阶谱系数

针对谱空间中离散的方程 (2.2.3) 的另一种处理方式(见 Canuto 等 [1][4]),是利用低阶谱系数与高阶谱系数的递推关系在 (2.2.3) 中消去 0 阶谱系数,而保留 2 阶谱系数。

调用两次谱积分运算 (2.4.5),可以得到这个递推关系:

$$\hat{u}_k = P_k\hat{u}_{k-2}^{(2)} + Q_k\hat{u}_k^{(2)} + R_k\hat{u}_{k+2}^{(2)}, \quad 2 \le k \le N, \tag{3.2.1}$$

其中 $P_k = \dfrac{c_{k-2}}{4k(k-1)}$, $Q_k = \dfrac{-e_{k+2}}{2(k^2-1)}$, $R_k = \dfrac{e_{k+4}}{4k(k+1)}$, $e_j = \begin{cases} 1, & j \le N, \\ 0, & j > N. \end{cases}$

当 $k=1$ 时,用如下方式来处理:

$$\begin{aligned} \hat{u}_1 &= \frac{2\hat{u}_0^{(1)} - \hat{u}_2^{(1)}}{2} = \hat{u}_0^{(1)} - \frac{1}{2}\left(\frac{\hat{u}_1^{(2)} - \hat{u}_3^{(2)}}{2\times 2}\right) \\ &= \hat{u}_0^{(1)} - \frac{1}{8}\hat{u}_1^{(2)} + \frac{1}{8}\hat{u}_3^{(2)}. \end{aligned} \tag{3.2.2}$$

当 $k=0$ 时,保持不作变形。将 (3.2.1),(3.2.2) 代入方程 (2.2.3),化简可得:

$$\begin{aligned} \lambda\hat{u}_0 - \hat{u}_0^{(2)} &= \hat{f}_0, \\ \lambda\hat{u}_0^{(1)} + \left(-1 - \frac{\lambda}{8}\right)\hat{u}_1^{(2)} + \frac{\lambda}{8}\hat{u}_3^{(2)} &= \hat{f}_1, \\ \frac{\lambda}{4k(k-1)}\hat{u}_{k-2}^{(2)} + \left(-1 - \frac{\lambda}{2(k^2-1)}\right)\hat{u}_k^{(2)} + \frac{\lambda\beta_{k+2}}{4k(k+1)}\hat{u}_{k+2}^{(2)} &= \hat{f}_k, \quad k = 2,\dots,N-2. \end{aligned} \tag{3.2.3}$$

类似地,将 (3.2.1),(3.2.2) 代入奇、偶项分离的边界条件 (3.1.3),整理得到:

$$\begin{aligned} \hat{u}_0 + \frac{1}{4}\hat{u}_0^{(2)} - \frac{7}{48}\hat{u}_2^{(2)} + \sum_{\substack{k=4 \\ k\text{为偶}}}^{N-2} \frac{3}{(k^2-1)(k^2-4)}\hat{u}_k^{(2)} &= \frac{b+a}{2}, \\ \hat{u}_0^{(1)} - \frac{1}{12}\hat{u}_1^{(2)} + \sum_{\substack{k=3 \\ k\text{为奇}}}^{N-2} \frac{3}{(k^2-1)(k^2-4)}\hat{u}_k^{(2)} &= \frac{b-a}{2}. \end{aligned} \tag{3.2.4}$$

有两点说明:

1. (3.2.3) 和 (3.2.4) 构成的方程组中含有 $\hat{u}_0$ 与 $\hat{u}_0^{(1)}$ 作为未知数的一员;而相应的,2 阶谱系数中有两个是已经确定的了,即 $\hat{u}_N^{(2)} = \hat{u}_{N-1}^{(2)} = 0$,不出现在未知数中。因此看到,仍是 $N+1$ 个方程求解 $N+1$ 个未知数

[4]该书中有一些错误,按此处所写为准。

$\left\{\hat{u}_0, \hat{u}_0^{(1)}, \{\hat{u}_k^{(2)}\}_{k=0,1,\ldots,N-2}\right\}$。

2. 从 (3.2.3) 注意到，实际上，方法 II 与方法 I 除了边值条件和两行方程不同外，系数矩阵的通项表达是相同的。

我们分开奇、偶项来构造拟三对角方程组。排列顺序为：以两边界条件分别作为奇、偶的首行，以 (3.2.3) 的前两行分别构成奇、偶的次行，其余通项按序下排即可。得到如下系统，

$$\begin{pmatrix} 1 & \frac{1}{4} & \frac{-7}{48} & \cdots & * & * & * \\ * & * & & & & & \\ & * & * & * & & & \\ & & & \cdots & & & \\ & & & * & * & * & \\ & & & & * & * & * \\ & & & & & * & * \end{pmatrix} \begin{pmatrix} \hat{u}_0 \\ \hat{u}_0^{(2)} \\ \hat{u}_2^{(2)} \\ \vdots \\ \hat{u}_{N偶-6}^{(2)} \\ \hat{u}_{N偶-4}^{(2)} \\ \hat{u}_{N偶-2}^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{b+a}{2} \\ \hat{f}_0 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{N偶-6} \\ \hat{f}_{N偶-4} \\ \hat{f}_{N偶-2} \end{pmatrix}, \begin{pmatrix} 1 & \frac{-1}{12} & * & \cdots & * & * & * \\ * & * & * & & & & \\ & * & * & * & & & \\ & & & \cdots & & & \\ & & & * & * & * & \\ & & & & * & * & * \\ & & & & & * & * \end{pmatrix} \begin{pmatrix} \hat{u}_0^{(1)} \\ \hat{u}_1^{(2)} \\ \hat{u}_3^{(2)} \\ \vdots \\ \hat{u}_{N奇-6}^{(2)} \\ \hat{u}_{N奇-4}^{(2)} \\ \hat{u}_{N奇-2}^{(2)} \end{pmatrix} = \begin{pmatrix} \frac{b-a}{2} \\ \hat{f}_1 \\ \hat{f}_3 \\ \vdots \\ \hat{f}_{N奇-6} \\ \hat{f}_{N奇-4} \\ \hat{f}_{N奇-2} \end{pmatrix}, \tag{3.2.5}$$

其中，$N_{偶}$（$N_{奇}$）表示最大偶（奇）数项，$*$ 表示系数矩阵中非零项。

小结一下，方法 II 求解 Helmholtz 方程的 Dirichlet 问题 (2.2.1) 的步骤是：首先，确定要作的 Chebyshev 截断的项数 N，对右端函数 f 作相应的离散 Chebyshev 变换 (2.3.2)，得到系数 $\{\hat{f}_k\}_{k=0,\ldots,N}$。其次，求解拟三对角方程组 (3.2.5)，得到 2 阶谱系数 $\{\hat{u}_k^{(2)}\}_{k=0,\ldots,N}$ 和 $\hat{u}_0, \hat{u}_0^{(1)}$。最后，利用谱积分运算 (2.4.5) 计算其余 1 阶谱 $\{\hat{u}_k^{(1)}\}_{k=1,\ldots,N}$ 与 0 阶谱 $\{\hat{u}_k\}_{k=1,\ldots,N}$，并可以进而通过这些谱系数计算函数、1 阶 / 2 阶导函数的函数值。

4 设计拟三对角方程的求解算法

我们考虑的问题中出现的“拟三对角方程组”可以写成下面的一般形式（以六阶为例表示），

$$\begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & a_{33} & a_{34} & & \\ & & a_{43} & a_{44} & a_{45} & \\ & & & a_{54} & a_{55} & a_{56} \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{pmatrix}.$$

假设系数矩阵 A 可以作 LU 分解，即有

$$A = \begin{pmatrix} 1 & & & & & \\ l_{21} & 1 & & & & \\ & l_{32} & 1 & & & \\ & & l_{43} & 1 & & \\ & & & l_{54} & 1 & \\ l_{61} & l_{62} & l_{63} & l_{64} & l_{65} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & a_{12} & & & & \\ & u_{22} & a_{23} & & & \\ & & u_{33} & a_{34} & & \\ & & & u_{44} & a_{45} & \\ & & & & u_{55} & a_{56} \\ & & & & & u_{66} \end{pmatrix},$$

我们按下面的顺序进行一个比较系数的过程，可以依次确定 L 和 U。

迭代器	行,列	对应关系		被决定的对象
$i=1$	$1,1$	$u_{11}=a_{11}$	$\Rightarrow$	$u_{11}=a_{11}$
$i=2,\ldots,n-1$	$i,i-1$	$l_{i,i-1}\times u_{i-1,i-1}=a_{i,i-1}$	$\Rightarrow$	$l_{i,i-1}=\dfrac{a_{i,i-1}}{u_{i-1,i-1}}$
	i,i	$l_{i,i-1}\times a_{i-1,i}+u_{i,i}=a_{i,i}$	$\Rightarrow$	$u_{i,i}=a_{i,i}-l_{i,i-1}\times a_{i-1,i}$
$i=n,j=1$	$n,1$	$l_{n,1}\times u_{11}=t_1$	$\Rightarrow$	$l_{n,1}=\dfrac{t_1}{u_{11}}$
$j=2,\ldots,n-1$	n,j	$l_{n,j-1}\times a_{j-1,j}+l_{n,j}\times u_{j,j}=t_j$	$\Rightarrow$	$l_{n,j}=\dfrac{t_j-l_{n,j-1}\times a_{j-1,j}}{u_{j,j}}$
$j=n$	n,n	$l_{n,n-1}\times a_{n-1,n}+1\times u_{n,n}=t_n$	$\Rightarrow$	$u_{n,n}=t_n-l_{n,n-1}\times a_{n-1,n}$

有了 A 的 LU 分解,我们可以分两步走求 $Ax=b$ 的解:$Ly=b$ 和 $Ux=y$。可依下面顺序确定:

迭代器	被决定的对象
$i=1$	$y_1=b_1$
$i=2,\ldots,n-1$	$y_i=b_i-l_{i,i-1}\times y_{i-1}$
$i=n$	$y_n=b_n-\sum_{j=1}^{n-1}l_{n,j}\times y_j$

迭代器	被决定的对象
$j=n$	$x_n=\dfrac{y_n}{u_{n,n}}$
$j=n-1,\ldots,1$	$x_j=\dfrac{y_j-a_{j,j+1}\times x_{j+1}}{u_{j,j}}$

此拟三对角方程算法的运算量为,乘除法 $8n-10$,加减法 $5n-6$(作为参考,三对角方程的追赶法的运算量为,乘除法 $5n-4$,加减法 $3n-3$)。因而可知方法 I 和方法 II 的方程可以很高效地求解。

5 数值实验

现在我们考虑对上述两种求解过程进行一个比较。方法 I 是先求解“拟三对角方程”获得 0 阶谱系数，然后作两次离散的谱微分依次得到 1、2 阶的谱系数。方法 II 是先求解“拟三对角方程”获得 2 阶谱系数，然后作两次离散的谱积分依次得到 1、0 阶的谱系数。假设第一步解方程的误差是 $E^{(0)}$，这个误差会在谱微分或者谱积分中进一步累积。对于这两种运算的误差放大倍数，Greengard[2] 进行了研究。他证明得到，设 D 表示谱微分算子，J 表示谱积分算子，$\mathbf{a}=(a_0,a_1,\dots,a_N),\hat{\mathbf{a}}=(a_0+\epsilon,a_1+\epsilon,\dots,a_N+\epsilon)$，则[5]

$$\frac{\|D(\hat{\mathbf{a}})-D(\mathbf{a})\|_\infty}{\|\hat{\mathbf{a}}-\mathbf{a}\|_\infty}=O(N^2),\quad \frac{\|J(\hat{\mathbf{a}})-J(\mathbf{a})\|_\infty}{\|\hat{\mathbf{a}}-\mathbf{a}\|_\infty}<2.4.$$

因此他认为，以谱积分方法主导的方法 II，会在未知函数的 1 阶谱系数上[6]，比谱微分方法主导的方法 I 更加精确。然而，本次测试发现，初始误差的不同以及有限项截断方法所引入的误差，会与上述效应共同作用，而最终的结果并不如 Greengard 所期望的那样支持方法 II。本次测试展示了三个较有代表性的例子。实验在 MATLAB 7.11(UNIX) 版本中进行，采用的是双精度（64 位）浮点运算（精度达到 $2^{-52}\approx 2\times 10^{-16}$）。

检验过程是这样进行的：首先选定 15 个按对数意义均匀分布在 $[10,10^{3.1}]$ 的 N。其次，计算真实解的 0、1、2 阶谱系数：为了尽量减少其中数值积分带来的误差，由误差的收敛性，我们选用最大的 $N=10^{3.1}$ 为积分插值点，用 Gauss-Lobatto 公式 (2.3.2) 来近似 N 阶以下的真实的 0、1、2 阶谱系数。再次，对每个不同的 N，调用上述两方法求解，获得数值解的 0、1、2 阶谱系数。最后，计算两种方法的结果的于真实解的相对误差，取各阶谱系数的最大相对误差作图，以蓝线表示方法 I，以红线表示方法 II（重合时，后者的图线覆盖前者）。

示例 1. 第一个例子来自于 [5]，两点边值问题 (2.2.1) 如下给出

$$\begin{cases}-u''+400u=-400\cos^2\pi x-2\pi^2\cos 2\pi x, & x\in(-1,1),\\ u(-1)=\dfrac{e^{-40}+e^{20}}{1+e^{-20}}-1, & u(1)=0.\end{cases}$$

真实解为：

$$g(x)=\frac{e^{-20}}{1+e^{-20}}e^{20x}+\frac{1}{1+e^{-20}}e^{-20x}-\cos^2\pi x.$$

这个函数的特点是：有细微的抖动；在边界 $x=-1$ 处是光滑的，但数量级比较大。算法运行结果如图5.1所示。

结果中两种方法几乎看不出差别。这可能是由于机器精度比较高，初始误差 $E_r^{(0)}$ 太小，以致 $E_r^{(0)}N^2$ 项不显著（见注释5）。为了让 N^2 阶显示出来，我开始对拟三对角方程的解作如下意义的扰动：

$$\text{在方法一中作}\hat{u}_k:=\hat{u}_k\cdot(1+error),\quad \text{在方法二中作}\hat{u}_k^{(2)}:=\hat{u}_k^{(2)}\cdot(1+error).$$

[5]Greengard 研究的是 Chebyshev（无穷项）级数，而我们的 Chebyshev-tau 逼近用的是截断的级数，相应的运算就要加上个“离散”二字；然而类似的结论仍然成立。此时的离散的谱微分运算为一个矩阵设为 $\hat{D}$，仍然很容易看到 $E^{(1)}=|U^{(1)}-U^{(1)}_{real}|=|\hat{D}U-(\hat{D}U_{real}+U^{(1)}_{real}$的末元)$|\le\hat{D}E^{(0)}+\epsilon$，从而 $\left\|E^{(1)}\right\|\le O(N^2)\left\|E^{(0)}\right\|+\epsilon$. 类似地可以证明，离散的谱积分矩阵 $\hat{J}$ 只有系数与 N 无关的线性误差放大。

$$\hat{D}=\begin{bmatrix}0&1&0&3&0&5&\dots\\ &0&4&0&8&0&\dots\\ &&0&6&0&10&\dots\\ &&&\ddots&&&\end{bmatrix}_{N\times N}$$

[6]由于方法 I 直接求解 0 阶谱，没有误差放大，直观上会比方法 II 放大两次后的结果精确；同理，方法 II 在 2 阶谱上会精确。因此，我们比较关心 1 阶谱的计算上哪种方法更优。

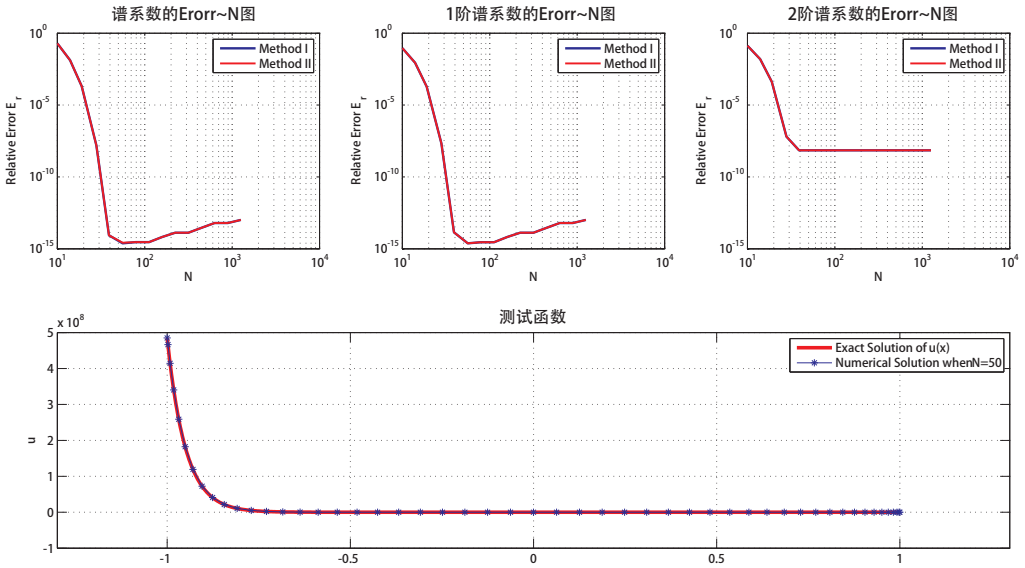

图 5.1. 测试函数 1 及两方法在数值实验中的表现

测试环境的机器精度为 16 位有效数字——当令 $error$ 为 10^{-k} 阶的随机数时，结果的第 k 位有效数字会受到影响，近似模拟了精度为 k 位有效数字的机器下的结果——因此，我取 $error$ 为阶 $10^{-16} \sim 10^{-10}$ 的随机数，重做实验。此时谱微分和谱积分的效应之差别就显现出来了，如图5.2，

蓝线：从左往右看，1 阶和 2 阶谱系数的误差随 N 线性增长：这正是 $E_r^{(1)} = E_r^{(0)}O(N^2)$, $E_r^{(2)} = E_r^{(0)}O(N^4)$ 在双对数坐标系下的相应表现。

红线：从右往左看，1 阶和 0 阶谱系数的误差只是向上平移：这也是 $E_r^{(1)} = Const{\cdot}E_r^{(0)}$, $E_r^{(2)} = Const{\cdot}E_r^{(0)}$ 在双对数坐标系下的相应表现。

此例中，在 1 阶谱的计算上，方法 II 的结果略优于方法 I。

示例 2. 下面这个例子体现的是边界层的影响，两点边值问题 (2.2.1) 如下给出，

$$\begin{cases} u'' - 10^5 u = 0, \\ u(-1) = 1, \quad u(1) = 2. \end{cases}$$

真实解为：

$$g(x) = c_1 e^{\sqrt{10^5}x} + c_2 e^{-\sqrt{10^5}x},$$

其中参数 c_1, c_2 可以由两个边界条件解出。算法运行结果如图5.3，

然后，对初始误差作 $10^{-16} \sim 10^{-10}$ 的扰动后，得到的结果如图5.4所示。仍然，$O(N^2)$ 的误差放大效应以及常数倍的效应也慢慢地显示出来了。但是，在误差随着 N 开始上涨很多之前，方法 II 反而并不如方法 I。

示例 3. 最后一个例子是高度振动的函数。一般地，谱方法在这种问题上比有限差分法要好；因此，我们有必要

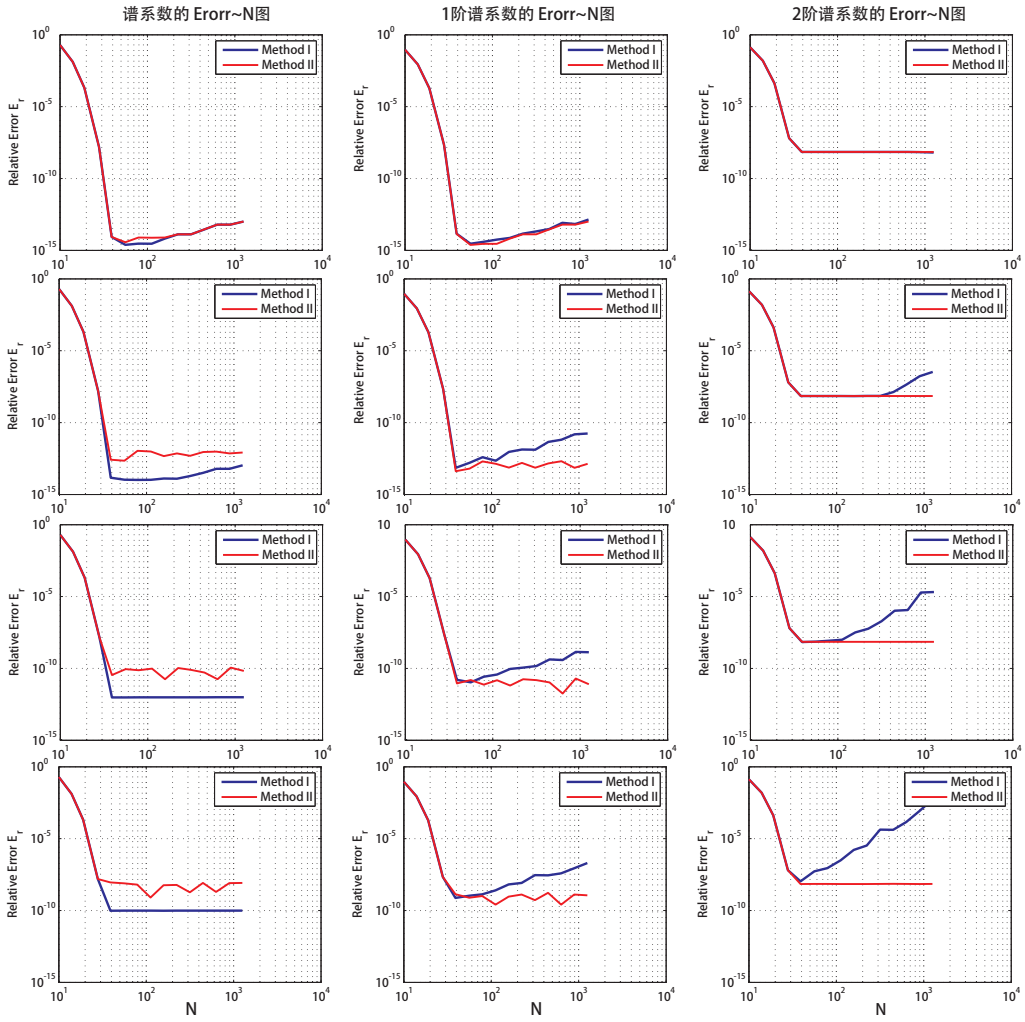

图 5.2. 例 1 扰动后各阶谱系数的相对误差。扰动比例的数量阶，从上至下依次为 $10^{-16}, 10^{-14}, 10^{-12}$和$10^{-10}$.

看一下在这种函数的计算上，方法 I 与方法 II 的差异。两点边值问题 (2.2.1) 如下给出，

$$\begin{cases} -u'' + (\frac{25}{4} - 2500)u = 250\cos(50x + 50)e^{-\frac{5}{2}(x+1)} \\ u(-1) = 0, \quad u(1) = \sin 100e^{-5}. \end{cases}$$

真实解为：

$$g(x) = \sin(50x + 50)e^{-\frac{5}{2}(x+1)}$$

算法运行结果如图5.5，这次，对初始误差作 $10^{-16} \sim 10^{-13}$ 的扰动，得到的结果如图5.6所示。我们看到，在 1 阶谱的计算上，两种方法并没有很实质性的差异。

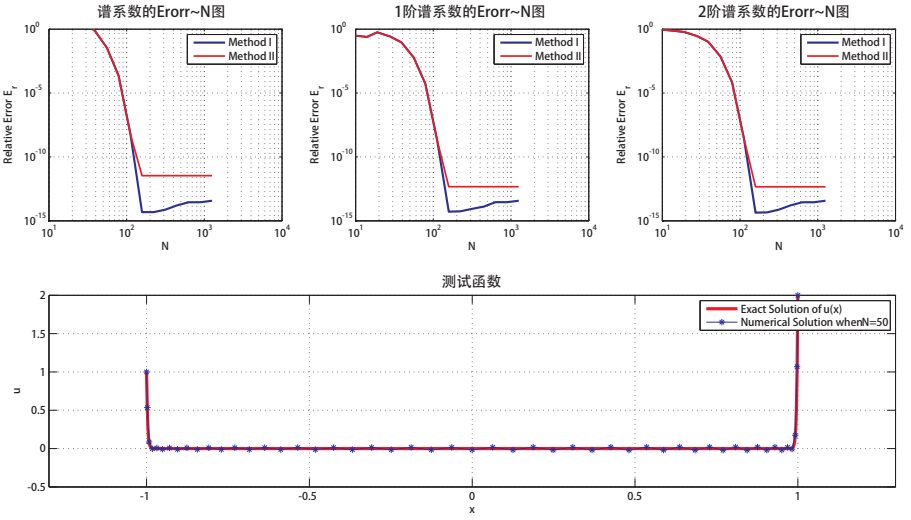

图 5.3. 测试函数 2 及两方法在数值实验中的表现

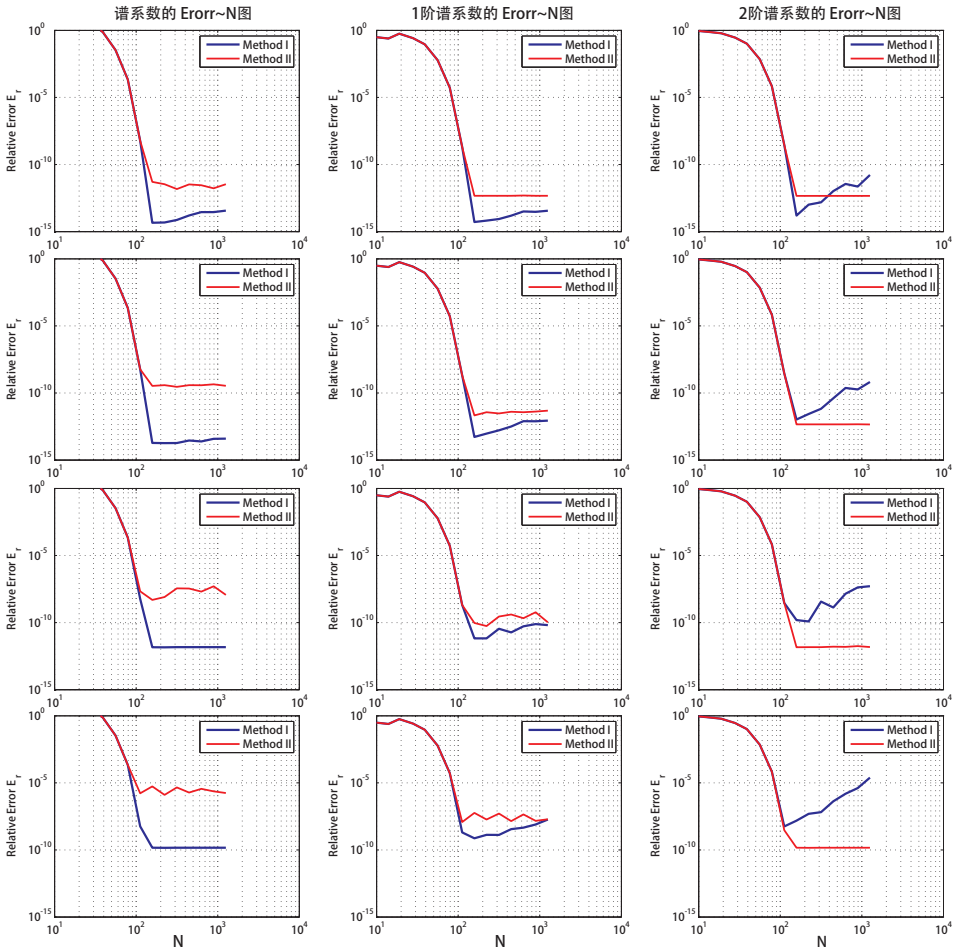

图 5.4. 例 2 扰动后各阶谱系数的相对误差。扰动比例的数量阶，从上至下依次为 $10^{-16}, 10^{-14}, 10^{-12}$和$10^{-10}$.

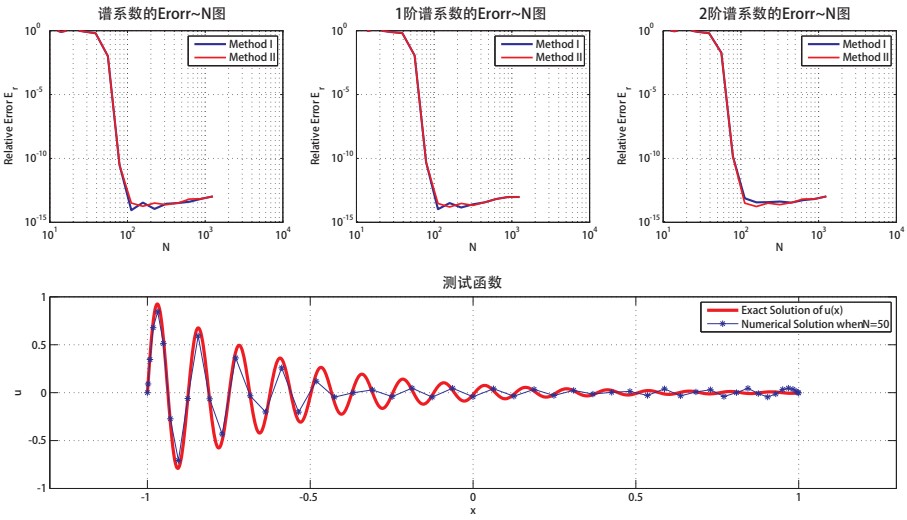

图 5.5. 测试函数 3 及两方法在数值实验中的表现

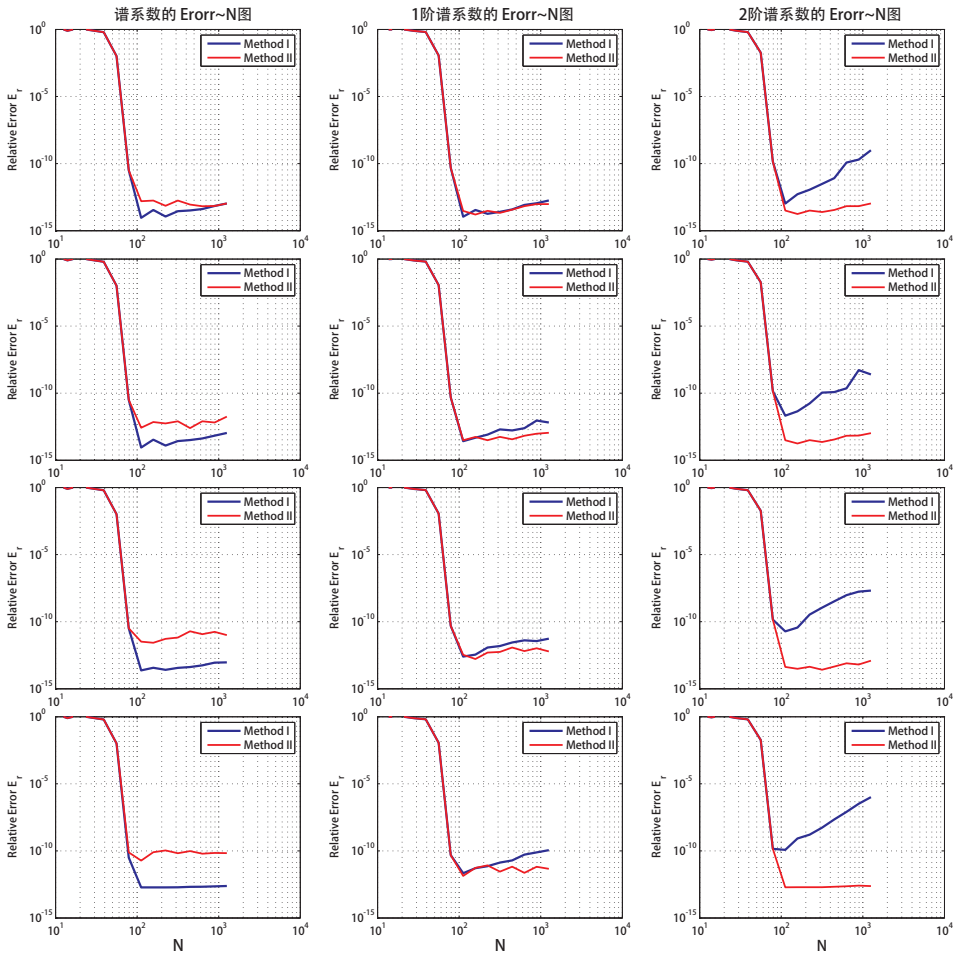

图 5.6. 例 3 扰动后各阶谱系数的相对误差。扰动比例的数量阶，从上至下依次为 $10^{-16}, 10^{-15}, 10^{-14}$和$10^{-13}$.

6 结果分析

根据实验结果，我们作出如下的分析：

首先，根据“N 较大时 $2.4 < O(N^2)$”就判断方法 II 优于方法 I，是一个并不全面的论断。因为，事实上，谱方法的精度并不是随着 N 增大而无限减小的，而是在 10^2 阶左右已达最优，之后误差便增加了。因此，在 N 取很大时作比较并没有多少意义；而应该比较的是 $O(N^2_{\text{最优}})$ 与 2.4 的值孰大孰小，来判定有实际意义的误差放大比率。而在我们的结果中，蓝线的最低点不仅与红线相持，而且在示例 2 中还有很大的优势。

其次，在示例 2 中，方法 II 在 2 阶谱系数上竟不如方法 I 两次误差放大后的结果，这是有悖直观的。这说明解“拟三对角方程”时，方法 II 的误差过大。事实上，认真比较方法 I 与方法 II 的“拟三对角矩阵”就会发现，这两个矩阵只有最前两行是不同的（其中一行是边界）；然而，方法 II 中这两行的不均匀性却大大加重了矩阵的奇异性。图6.1和6.2为前两个例子中的条件数对比。可以看到方法 II 的条件数比方法 I 要大 $10^3 \sim 10^5$ 阶之多。尤其是对于有边界层的示例 2，这个条件数是灾难性的大。这可能说明了方法 II 的初始误差较大的原因。

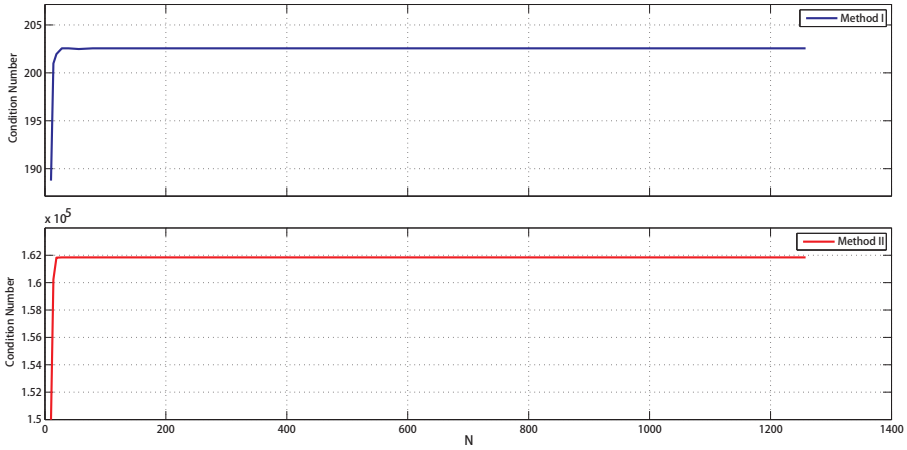

图 6.1. 示例 1 中两方法的条件数 $\sim N$

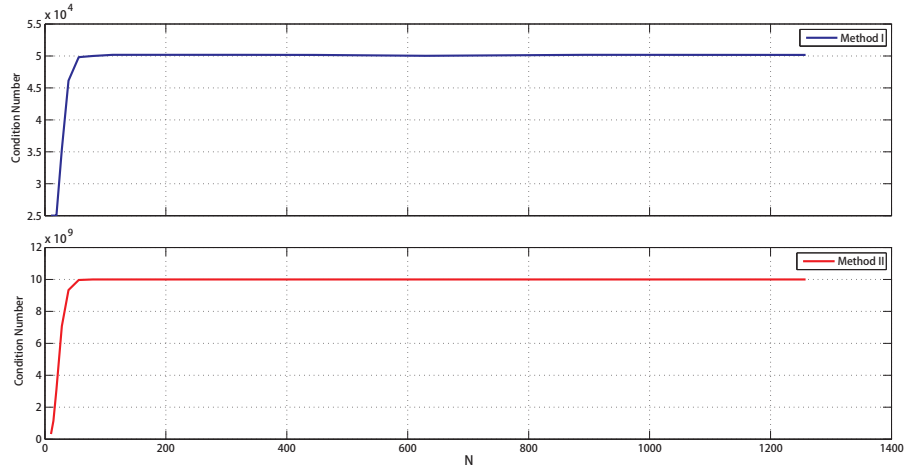

图 6.2. 示例 2 中两方法的条件数 $\sim N$

最后，根据第二点分析，可以尝试通过降低方法 II 的条件数而改善它。为此，我开始尝试对这样的拟三对角系统进行一些预处理。例如对于 $Ax = b$，作预处理 $PAx = Pb$。一种可行的取法是如果

$$A = \begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ & a_{32} & a_{33} & a_{34} & & \\ & & a_{43} & a_{44} & a_{45} & \\ & & & a_{54} & a_{55} & a_{56} \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{pmatrix}, \text{则取} P = \begin{pmatrix} N & & & & & \\ & N & & & & \\ & & N & & & \\ & & & N & & \\ & & & & N & \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{pmatrix},$$

这个预处理会把三个例子的方法 II 的条件数下降 $10 \sim 10^2$ 阶。然而，重做实验并没有看到第二个例子的明显改善。更多的工作也许可以留待今后来做。

感谢

在此特别感谢叶兴德老师，他的合理选题以及适时、耐心的指导，使我学会一步一步完成毕业论文的方法；同时感谢的是室友章瑞，感谢他期间同我的讨论；最后，Leader 的书 [3] 让我学会了数值分析的珍贵思想。

参考文献

[1] C. Canuto, MY Hussaini, A. Quarteroni, and TA Zang. *Spectral methods: fundamentals in single domain*, chapter 4, pages 173--177. Springer, 2006. 6, 8

[2] L. Greengard. Spectral integration and two-point boundary value problems. *SIAM Journal on Numerical Analysis*, pages 1071--1080, 1991. 3, 6, 11

[3] J.J. Leader. *Numerical analysis and scientific computation.* Pearson Addition Wesley, Boston, San Francisco, New York, 2004. 17

[4] R. Peyret. *Spectral methods for incompressible viscous flow*, volume 148, chapter 3, pages 39--53. Springer Verlag, 2002. 5

[5] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12. Springer Verlag, 2002. 11

[6] 叶兴德，程晓良，陈明飞，薛莲. *数值分析基础*. 浙江大学出版社, 2008.

A 程序

- 主程序 TwiceErrorTest.m 脚本
- 离散的 Chebyshev 变换函数 ChebyTransform.m
- 求解拟三对角方程组的函数 qtrid.m
- 方法 I 计算各阶谱系数的函数 Method1.m

- 方法 II 计算各阶谱系数的函数 Method2.m
- 测试函数及误差比较的绘制函数 plotError.m
- 条件数对比绘制函数 plotCond.m
- 测试函数集 Test_functions.m，可拷贝代替 TwiceTestError.m 的测试函数代码

程序 1. 主程序 TwiceErrorTest.m 脚本

```
%% Prescription
clear all;
global N disturb;
% N is the number of Chebyshev grid points
% as well as the number of terms in the truncated Chebyshev expansion
% disturb is to add a slight error in uc in Method I and D2uc in Method II
% to test stability of these methods. Default to be 0.
global a b lamda;
_____________________________________________________________________________
%% Place our test functions here, copied from Test_functions.m
% e.g.Example2: Dual tipping
lamda=1e5;
slamda=sqrt(lamda);
f=@(x)(0);
a=1;b=2;
Coef=[exp(-slamda) exp(slamda);exp(slamda) exp(-slamda)]\[a;b];
g=@(x)(Coef(1)*exp(slamda*x)+Coef(2)*exp(-slamda*x));
D1g=@(x)(Coef(1)*slamda*exp(slamda*x)-Coef(2)*slamda*exp(-slamda*x));
D2g=@(x)(lamda*g(x));
clear slamda;
_____________________________________________________________________________
%% Generate N~sample
times=15; % number of the sample of N;
n=floor(logspace(1,3.1,times)); % sample of N
_____________________________________________________________________________
%% Calculate exact Chebyshev coefficients using n(end) grid points
gc=ChebyTransform(g,n(end));
D1gc=ChebyTransform(D1g,n(end));
D2gc=ChebyTransform(D2g,n(end));
_____________________________________________________________________________
%% Calibrate relative errors in uc, D1uc and D2uc, which change as functions of N
%--- Calculate absolute errors--------%
                      % Three columns, storing error
error1=zeros(times,3);%for uc, D1uc and D2uc respectively
error2=zeros(times,3);
                      % Storing the condition number of the unique linear
cond1=zeros(times,1); %system involved that contributes to our intial error
cond2=zeros(times,1);
for i=1:times
    N=n(i);
    %disturb=(rand(N+1,1)-0.5)*10^(-12); %For exploring with added disturbance
    disturb=0;
    fc=ChebyTransform(f,N);% Spectral Coefficient for f when N, prerequisite for all that follows
    [uc,D1uc,D2uc,conduc]=Method1(fc);% Call Method1.m
    error1(i,:)=[norm(uc-gc(1:N+1),inf),norm(D1gc(1:N+1)-D1uc,inf),norm(D2uc-D2gc(1:N+1),inf)];
    cond1(i)=conduc;
    [uc,D1uc,D2uc,condD2uc]=Method2(fc);% Call Method2.m
    error2(i,:)=[norm(uc-gc(1:N+1),inf),norm(D1gc(1:N+1)-D1uc,inf),norm(D2uc-D2gc(1:N+1),inf)];
    cond2(i)=condD2uc;
end
%----To get the relative errors, perform a column transformation----%
error1=error1*[1/norm(gc,inf) 0 0;0 1/norm(D1gc,inf) 0;0 0 1/norm(D2gc,inf)];
error2=error2*[1/norm(gc,inf) 0 0;0 1/norm(D1gc,inf) 0;0 0 1/norm(D2gc,inf)];
_____________________________________________________________________________
%% Plot Errors and Condition Numbers
plotError(n,error1,error2,f,g); %Call plotError.m
```

```
57  plotCond(n,cond1,cond2); % Call plotCond.m
```

程序 2. 离散的 Chebyshev 变换函数 ChebyTransform.m

```
1  function fc=ChebyTransform(f,N)
2  %Performing the Chebyshev transformation on the inhomogeneous RHS term f(x)
3  %in order to get the coefficients of its truncated Chebyshev series
4  % Input:function f; Output:(N+1)-column-vector fc
5  fc=zeros(N+1,1);
6  cbar=ones(N+1,1); %This is parameter c̄_k, all 1s but 2 when k=N
7  cbar([1 end])=2;
8  xgrid=cos(pi/N.*(0:N))';% Generate N+1 Chebyshev points from 1 to -1
9  for k=0:N
10     ki=k+1;%Matlab index modification k-index
11     sum=0;
12     for i=0:N
13         ii=i+1;%i-index
14         sum=sum+f(xgrid(ii))/cbar(ii)*cos(k*pi*i/N);
15     end
16     fc(ki)=2/N/cbar(ki)*sum;
17 end
18 % Thresholding fc
19 fc(abs(fc)<norm(fc,inf)*5*eps)=0;
```

程序 3. 求解拟三对角方程组的函数 qtrid.m

```
1  function x=qtrid(Top,Sub,Diag,Sup,b)
2  % Function to solve a quasi-tridiagonal linear system
3  % Ax=b, where:
4  % A is like:
5  %   [T1 T2 T3 T4 T5 T6]->Top
6  %   [A1 B1 C1         ]
7  %   [   A2 B2 C2      ]
8  %   [      A3 B3 C3   ]
9  %   [         A4 B4 C4]
10 %   [            A5 B5]
11 %                Sub Diag Sup
12 % Input:
13 %   Top: n-column-vector [T1 T2 T3 T4 T5 T6]'
14 %   Sub: (n-1)-column-vector [A1 A2 A3 A4 A5 A6]'
15 %   Diag: (n-1)-column-vector [B1 B2 B3 B4 B5 B6]'
16 %   Sup: (n-2)-column-vector [C1 C2 C3 C4]'
17 %   b: n-column-vector
18 %   (All these vectors correspond natually and orderedly (left to right)
19 %   to the above Wilkinson diagram)
20 % Return:
21 %   x: n-column-vector
22 % Description of the Algorithm
23 % 1.We solve the equation based on the LU decomposition and save time by
24 %   removing redundent null multiplications.
25 % 2.The main process is based on the method of correlating
26 %   undetermined coefficents. (Details in the scripts)
27 ------------------------------------------------------------------------
28 % Formal Program Begin:
29 n=length(b);
30 if length(Top)~=n || length(Sub)~=(n-1) || ...
31         length(Diag)~=(n-1) || length(Sup)~=(n-2)
32     error('qtrid input error:Dimension dismatch.');
33 end
34 L=zeros(2*n-3,1);
35 U=zeros(n,1);
36 ------------------------------------------------------------------------
37 %% Commence the LU Decomposition
38 U(1)=Diag(n-1);
39 for i=2:n-1
40     k=i-1;
```

```
    L(k)=Sup(n-i)/U(i-1);
    U(i)=Diag(n-i)-L(k)*Sub(n-i+1);
end
%Now that we have n-2 diagonal items of L and n-1 item of U, we are
%heading for the remaining n-1 baseline items of L and the last item of U
L(n-1)=Top(n)/U(1);
for i=2:n-1
    k=i+n-2;%Storage position of L
    L(k)=(Top(2*n-1-k)-L(k-1)*Sub(n-i+1))/U(i);
end
U(n)=Top(1)-L(2*n-3)*Sub(1);
_____________________________________________________________________________
%% Ly=b with Forward Subsititution
x=flipud(b); % We work in the same single vector for compactness
for k=2:n-1
    x(k)=x(k)-L(k-1)*x(k-1);
end
x(n)=x(n)-dot(L(n-1:2*n-3),x(1:n-1));
_____________________________________________________________________________
%% Ux=y with Backward Subsititution
x(n)=x(n)/U(n);% That said, we started with the implied x=y for compactness;
for k=n-1:-1:1
    x(k)=(x(k)-Sub(n-k)*x(k+1))/U(k);
end
x=flipud(x);
```

程序 4. 方法 I 计算各阶谱系数的函数 Method1.m

```
function [uc,D1uc,D2uc,conduc]=Method1(fc)
% Method I
% We pseudo-construct the quasi-tridiag coefficient matrix and then solve
% the Chebyshev coefficients of u. Later, we retrieve first and second
% order spectral coefficents from the differentiation recursion formula.
%
% Input: (N+1)-column-vector fc
% Call: qtrid(Top,Sub,Diag,Sup,b) to solve quasi-tridiag linear system
% Output: (N+1)-column-vectors uc, D1uc, D2uc (i.e.0,1,2-order Spectral Coeff.)
%      conduc:condition number of the involved linear system solving for uc
_____________________________________________________________________________
global a b lamda N %'disturb' claimed later
if(mod(N,2)==0)% Determine the largest even number and odd number below N
    Neven=N;Nodd=N-1;
else    Neven=N-1;Nodd=N;
end
_____________________________________________________________________________
%% Part I: Solve the Chebyshev coefficients of u indexed 0,2,...,Neven.
%----------- Construct the quasi-tridiag matrix
leng=Neven/2;% length of Sub, Diag
size=leng+1;% size of the quasi-tridiag system
Sub=zeros(leng,1);Diag=zeros(leng,1);Sup=zeros(leng-1,1);
B=zeros(leng,1);% Keep in mind that we'll later insert a boundary condition as the first row
%-- k=2 ----------------%
k=2;
tmp=[2/4/k/(k-1);-1/2/(k^2-1);1/4/k/(k+1)];
Sub(k/2)=tmp(1)*lamda;
Diag(k/2)=tmp(2)*lamda-1;
Sup(k/2)=tmp(3)*lamda;
B(k/2)=dot(tmp,fc([k-2+1,k+1,k+2+1]));
%-- k=4:2:Neven-4 -------%
for k=4:2:Neven-4
    tmp=[1/4/k/(k-1);-1/2/(k^2-1);1/4/k/(k+1)];
    Sub(k/2)=tmp(1)*lamda;
    Diag(k/2)=tmp(2)*lamda-1;
    Sup(k/2)=tmp(3)*lamda;
    B(k/2)=dot(tmp,fc([k-2+1,k+1,k+2+1]));
end
%-- k=Neven-2 -----------%;
```

```
k=Neven-2;
tmp=[1/4/k/(k-1);-1/2/(k^2-1)];
Sub(k/2)=tmp(1)*lamda;
Diag(k/2)=tmp(2)*lamda-1;
B(k/2)=dot(tmp,fc([k-2+1,k+1]));
%-- k=Neven -------------%;
k=Neven;
tmp=[1/4/k/(k-1);0];
Sub(k/2)=tmp(1)*lamda;
Diag(k/2)=tmp(2)*lamda-1;
B(k/2)=dot(tmp,fc([k-2+1,k+1]));
%----------- Complete B with boundary condition
B=[a/2+b/2;B];
%----------- Solve the quasi-tridiag linear system
x1=qtrid(ones(size,1),Sub,Diag,Sup,B); % Call qtrid.m
%----------- Record the condition number
A=gallery('tridiag',Sub,[1;Diag],[1;Sup]);% Built-in function to construct a sparse-form tridiagonal matrix
A(1,:)=ones(1,size);conduc1=condest(A);

%% Part II: Solve the Chebyshev coefficients of u indexed 1,3,...,Nodd.
%----------- Construct the quasi-tridiag matrix
leng=(Nodd-1)/2;% length of Sub, Diag
size=leng+1;% size of the quasi-tridiag system
Sub=zeros(leng,1);Diag=zeros(leng,1);Sup=zeros(leng-1,1);
B=zeros(leng,1);% Keep in mind that we'll later insert a boundary condition as the first row
%-- k=3:2:Nodd-4 ------------%
for k=3:2:Nodd-4
    tmp=[1/4/k/(k-1);-1/2/(k^2-1);1/4/k/(k+1)];
    Sub((k-1)/2)=tmp(1)*lamda;
    Diag((k-1)/2)=tmp(2)*lamda-1;
    Sup((k-1)/2)=tmp(3)*lamda;
    B((k-1)/2)=dot(tmp,fc([k-2+1,k+1,k+2+1]));
end
%--  k=Nodd-2  --------------%
k=Nodd-2;
tmp=[1/4/k/(k-1);-1/2/(k^2-1)];
Sub((k-1)/2)=tmp(1)*lamda;
Diag((k-1)/2)=tmp(2)*lamda-1;
B((k-1)/2)=dot(tmp,fc([k-2+1,k+1]));
%--  k=Nodd -----------------%
k=Nodd;
tmp=[1/4/k/(k-1);0];
Sub((k-1)/2)=tmp(1)*lamda;
Diag((k-1)/2)=tmp(2)*lamda-1;
B((k-1)/2)=dot(tmp,fc([k-2+1,k+1]));
%----------- Complete B with boundary condition-%
B=[b/2-a/2;B];
%----------- Solve the quasi-tridiag linear system
x2=qtrid(ones(size,1),Sub,Diag,Sup,B);% Call qtrid.m
%----------- Record the condition number
A=gallery('tridiag',Sub,[1;Diag],[1;Sup]);% Built-in function to construct a sparse-form tridiagonal matrix
A(1,:)=ones(1,size);conduc2=condest(A);
conduc=max(conduc1,conduc2);% output

%% Part III:Catenate even-index uc(i.e.x1) and odd-index uc(i.e.x2)
uc=zeros(N+1,1);
uc((0+1):2:(Neven+1))=x1;
uc((1+1):2:(Nodd+1))=x2;
clear x1 x2 Sub Diag Sup B tmp leng size k;

%% Part IV:Obtain D1uc from uc by spectral differentiation
%D1uc is the (N+1)-column-vector that stores 1-order spectral coeff.
global disturb;
uc=uc+disturb.*norm(uc);% This is a slight disturb to test stability.
D1uc=zeros(N+1,1);
for k=0:N-1
    sum=0;
    p=k+1;
```

```
108      % Calculate sum part:
109      while p<=N
110          pi=p+1;
111          sum=sum+p*uc(pi);
112          p=p+2;
113      end %
114      ki=k+1;
115      D1uc(ki)=2*sum;
116  end
117  D1uc(1)=D1uc(1)/2;% modification for c0=2
118  D1uc(N+1)=0;
119  ____________________________________________________________________________
120  %% Part V:Obtain D2uc from D1uc by spectral differentiation
121  %D2uc is the (N+1)-column-vector that stores 2-order spectral coeff.
122  D2uc=zeros(N+1,1);
123  for k=0:N-1
124      sum=0;
125      p=k+1;
126      % Calculate sum part:
127      while p<=N
128          pi=p+1;
129          sum=sum+p*D1uc(pi);
130          p=p+2;
131      end %
132      ki=k+1;
133      D2uc(ki)=2*sum;
134  end
135  D2uc(1)=D2uc(1)/2;% modification for c0=2
136  D2uc(N+1)=0;
```

程序 5. 方法 II 计算各阶谱系数的函数 Method2.m

```
 1  function [uc,D1uc,D2uc,condD2uc]=Method2(fc)
 2  % Method II
 3  % Description:
 4  % In Method II we dropped 0-order spectral coeff. in favor of the 2-order.
 5  % The solution should include 2-order spectral coeff. (denoted as D2uc)
 6  % indexed with 0,1,...,N-2 (or 1,...,N-1 in MATLAB), together with D1uc(1)
 7  % and uc(1)(written in MATLAB notation). Note that D2uc(N)=D2uc(N+1) are
 8  % known to be null. So this checks: again, we have an (N+1)-column-vector
 9  % to solve and at the same time we have N+1 equations. Finally, we retrieve
10  % D1uc and uc from integration recusion relationship.
11  %
12  % Input: (N+1)-column-vector fc
13  % Call: qtrid(Top,Sub,Diag,Sup,b) to solve quasi-tridiag linear system
14  % Output: (N+1)-column-vectors uc, D1uc, D2uc
15  %   condD2uc:condition number of the involved linear system solving for D2uc
16  ____________________________________________________________________________
17  global a b lamda N%'disturb' claimed later
18  if(mod(N,2)==0)% Determine the largest even number and odd number below N
19      Neven=N;Nodd=N-1;
20  else    Neven=N-1;Nodd=N;
21  end
22  ____________________________________________________________________________
23  %% Part I: Solve for x=[uc(1),D2uc(1),D2uc(3),...,D2uc(Neven-2+1)]'.
24  %----------- Construct the quasi-tridiag matrix
25  leng=Neven/2;% length of Sub, Diag
26  size=leng+1;% size of the quasi-tridiag system
27  Sub=zeros(leng,1);Diag=zeros(leng,1);Sup=zeros(leng-1,1);Top=zeros(size,1);
28  B=zeros(leng,1);% Keep in mind that we'll later insert a boundary condition as the first row
29  %-- k=0 ----------------%
30  Sub(1)=lamda;Diag(1)=-1;Sup(1)=0;B(1)=fc(1);
31  %-- k=2:2:Neven-4 ------%
32  for k=2:2:Neven-4 % k=2 to be later modified
33      tmp=[1/4/k/(k-1);-1/2/(k^2-1);1/4/k/(k+1)];
34      Sub(k/2+1)=tmp(1)*lamda;
35      Diag(k/2+1)=tmp(2)*lamda-1;
```

```
    Sup(k/2+1)=tmp(3)*lamda;
    ki=k+1;
    B(k/2+1)=fc(ki);
end
%-- k=Neven-2 ----------%
k=Neven-2;
tmp=[1/4/k/(k-1);-1/2/(k^2-1)];
Sub(k/2+1)=tmp(1)*lamda;
Diag(k/2+1)=tmp(2)*lamda-1;
ki=k+1;
B(k/2+1)=fc(ki);
%-- k=2: modify Sub(2) due to c0--%
Sub(2)=Sub(2)*2;
%----------- Fill in Top:boundary --%
Top(1)=1;Top(2)=1/4;Top(3)=-7/48;
j=4;
for k=4:2:Neven-2
    Top(j)=3/(k-2)/(k-1)/(k+1)/(k+2);
    j=j+1;
end
%----------- Complete B with boundary condition-%
B=[a/2+b/2;B];
%----------- Solve the quasi-tridiag linear system
x1=qtrid(Top,Sub,Diag,Sup,B);% Call qtrid.m
%----------- Record the condition number
A=gallery('tridiag',Sub,[1;Diag],[1;Sup]);% Built-in
A(1,:)=Top';condD2uc1=condest(A);
_____________________________________________________________________________
%% Part II: Solve for x=[D1uc(1),D2uc(2),D2uc(4),...,D2uc(Nodd-2+1)]'.
%----------- Construct the quasi-tridiag matrix
leng=(Nodd-1)/2;% length of Sub, Diag
size=leng+1;% size of the quasi-tridiag system
Sub=zeros(leng,1);Diag=zeros(leng,1);Sup=zeros(leng-1,1);Top=zeros(size,1);
B=zeros(leng,1);% Keep in mind that we'll later insert a boundary condition as the first row
%-- k=1 ----------------%
Sub(1)=lamda;Diag(1)=-1-lamda/8;Sup(1)=lamda/8;B(1)=fc(2);
%-- k=3:2:Nodd-4 -------%
for k=3:2:Nodd-4
    tmp=[1/4/k/(k-1);-1/2/(k^2-1);1/4/k/(k+1)];
    Sub((k+1)/2)=tmp(1)*lamda;
    Diag((k+1)/2)=tmp(2)*lamda-1;
    Sup((k+1)/2)=tmp(3)*lamda;
    ki=k+1;
    B((k+1)/2)=fc(ki);
end
%-- k=Nodd-2 -----------%
k=Nodd-2;
tmp=[1/4/k/(k-1);-1/2/(k^2-1)];
Sub((k+1)/2)=tmp(1)*lamda;
Diag((k+1)/2)=tmp(2)*lamda-1;
ki=k+1;
B((k+1)/2)=fc(ki);
%----------- Fill in Top:boundary --%
Top(1)=1;Top(2)=-1/12;
j=3;
for k=3:2:Nodd-2
    Top(j)=3/(k-2)/(k-1)/(k+1)/(k+2);
    j=j+1;
end
%----------- Complete B with boundary condition-%
B=[b/2-a/2;B];
%----------- Solve the quasi-tridiag linear system
x2=qtrid(Top,Sub,Diag,Sup,B);% Call qtrid.m
%----------- Record the condition number
A=gallery('tridiag',Sub,[1;Diag],[1;Sup]);% Built-in
A(1,:)=Top';condD2uc2=condest(A);
condD2uc=max(condD2uc1,condD2uc2); % output
_____________________________________________________________________________
```

```
%% Part III:Catenate even-index uc(i.e.x1) and odd-index uc(i.e.x2)
uc0=x1(1);
D1uc0=x2(1);
D2uc=zeros(N+1,1);
D2uc(0+1:2:Neven-2+1)=x1(2:end);
D2uc(1+1:2:Nodd-2+1)=x2(2:end);
D2uc([end-1 end])=[0;0];
clear x1 x2 Top Sub Diag Sup B tmp leng size k j;
%% Part IV:Obtain D1uc from D2uc by spectral integration
%D1uc is the (N+1)-column-vector that stores 1-order spectral coeff.
global disturb;
D2uc=D2uc+disturb.*norm(D2uc);% This is a slight disturb to test stability.
D1uc=zeros(N+1,1);
D1uc(1)=D1uc0;% k=0
D1uc(2)=1/2*(2*D2uc(1)-D2uc(3));% k=1
for k=2:N-1
    ki=k+1;
    D1uc(ki)=(D2uc(ki-1)-D2uc(ki+1))/2/k;
end
D1uc(N+1)=0;
clear k ki;
%% Part V:Obtain uc from D1uc by spectral integration
uc=zeros(N+1,1);
uc(1)=uc0;% k=0
uc(2)=1/2*(2*D1uc(1)-D1uc(3));% k=1
for k=2:N-1
    ki=k+1;
    uc(ki)=(D1uc(ki-1)-D1uc(ki+1))/2/k;
end
uc(N+1)=D1uc(N)/2/N;
```

程序 6. 测试函数及误差比较的绘制函数 plotError.m

```
function plotError(n,error1,error2,f,g)
% Plot our function and the error of numerical solution in uc, D1uc and D2uc
% Input:
% 1)n: a sample of N
% 2)error1:times*3 matrix, with each column storing (relative) errors in uc
%   D1uc and D2uc respectively, generated in Method I (spectral differentiation)
% 3)error2:times*3 matrix, with each column storing (relative) errors in uc
%   D1uc and D2uc respectively, generated in Method II (spectral integration)
% 4)f: function_handle RHS function f.
% 5)g: function_handle, exact solution
%% Plot the relative errors in uc, D1uc and D2uc. Compare.
subplot(2,3,1);
loglog(n,error1(:,1),'b','Linewidth',2);grid on;hold on;
loglog(n,error2(:,1),'r','Linewidth',1.6);ylim([1e-15 1e0]);
legend('Method I','Method II');xlabel('\it{N}');ylabel('Relative Error E_r');
title('Max E_r in Chebyshev Expansion Coef. of \it{u(x)}');

subplot(2,3,2);
loglog(n,error1(:,2),'b','Linewidth',2);grid on;hold on;
loglog(n,error2(:,2),'r','Linewidth',1.6);ylim([1e-15 1e0]);
legend('Method I','Method II');xlabel('\it{N}');ylabel('Relative Error E_r');
title('Max E_r in Chebyshev Expansion Coef. of \it{u''(x)}');

subplot(2,3,3);
loglog(n,error1(:,3),'b','Linewidth',2);grid on;hold on;
loglog(n,error2(:,3),'r','Linewidth',1.6);ylim([1e-15 1e0]);
legend('Method I','Method II');xlabel('\it{N}');ylabel('Relative Error E_r');
title('Max E_r in Chebyshev Expansion Coef. of \it{u"(x)}');
%% Plot the test function
% Perform inverse Chebyshev transformation---use uc to produce un(x)
subplot(2,3,4:6);
```

```
% Analytical solution plot
t=linspace(-1,1,1000)';
plot(t,g(t),'r','Linewidth',3);grid on;hold on;
xlim([-1.3 1.3]); % axis for x
% Numerical solution plot
global N disturb%%%%%%%
N=50;disturb=0;
fc=ChebyTransform(f,N);
uc=Method1(fc);
xgrid=cos(pi/N.*(0:N))';% Generate N+1 Chebyshev points from 1 to -1
y=zeros(length(xgrid),1);
for i=0:N
    y=y+uc(i+1)*mfun('T',i,xgrid); % Use Matlab Built-in Library 'myfun',
end                        % parameter 'T' represents Chebyshev polynomials.
title('Illustration of the Test Function');
xlabel('\it{x}');ylabel('\it{u}');
pause;
plot(xgrid,y,'b-*');
legend('Exact Solution of \it{u(x)}','Numerical Solution when \it{N}=50');
```

程序 7. 条件数对比绘制函数 plotCond.m

```
function plotCond(n,cond1,cond2)
% Plot Condition Number
figure;
subplot(2,1,1)
semilogy(n,cond1,'b','Linewidth',2);
ylabel('Condition Number');
legend('Method I');grid on;
set(gca,'XTickLabel',[]);
subplot(2,1,2)
semilogy(n,cond2,'r','Linewidth',2);
ylabel('Condition Number');
xlabel('\it{N}');
legend('Method II');grid on;
```

程序 8. 测试函数集 Test_functions.m，可拷贝代替 TwiceTestError.m 的测试函数代码

```
%% Senario #1
lamda=400;
f=@(x)(-400*(cos(pi*x)).^2-2*pi^2*cos(2*pi*x));
g=@(x)(exp(-20)/(1+exp(-20))*exp(20*x)+exp(-20*x)/(1+exp(-20))-cos(pi*x).^2);
a=g(-1);b=0;
D1g=@(x)(20*exp(-20)/(1+exp(-20))*exp(20*x)-20*exp(-20*x)/(1+exp(-20))+pi*sin(2*pi*x));
D2g=@(x)(lamda*g(x)+4*pi*pi*cos(2*pi*x));
%--------------------------------------------------------------------------
%% Senario #2 Dual tipping
lamda=1e5;slamda=sqrt(lamda);
f=@(x)(0);
a=1;b=2;
Coef=[exp(-slamda) exp(slamda);exp(slamda) exp(-slamda)]\[a;b];
g=@(x)(Coef(1)*exp(slamda*x)+Coef(2)*exp(-slamda*x));
D1g=@(x)(Coef(1)*slamda*exp(slamda*x)-Coef(2)*slamda*exp(-slamda*x));
D2g=@(x)(lamda*g(x));
clear slamda;
%--------------------------------------------------------------------------
%% Senario #3 Osilation
lamda=25/4-2500;
f=@(x)(250.*cos(50*x+50).*exp(-5/2*(x+1)));
g=@(x)(sin(50*x+50).*exp(-5/2*(x+1)));
D1g=@(x)(50*cos(50*x+50).*exp(-5/2*(x+1))-5/2*sin(50*x+50).*exp(-5/2*(x+1)));
D2g=@(x)(lamda*g(x)-f(x));
a=0;b=g(1);
```

毕 业 论 文（设计） 考 核

一、 指导教师对毕业论文（设计）的评语：

指导教师(签名）_____

年 月 日

二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：

成绩比例	文献综述 占（10%）	开题报告 占（20%）	外文翻译 占（10%）	毕业论文（设计）质量及答辩 占（60%）	总评成绩
分 值					

答辩小组负责人（签名）_____

年 月 日