多重网格计算报告

polossk

June 5, 2022

Contents

1 问题说明

对于方程

$$-u''(x)+\gamma u(x)u(x)'=f(x), 0<x<1 \tag{1.1}$$

$$u(0)=u(1)=0 \tag{1.2}$$

选用 FAS V-Cycle 方法，针对不同的 $f(x)$，在 $n=512$ 的网格上对其进行求解，并且观察其收敛性。其中

$$f_1(x)=2+\gamma(x-x^2)(1-2x) \tag{1.3}$$

$$f_2(x)=(x^2+3x)\mathrm{e}^x+\gamma(x^4-2x^2+x)\mathrm{e}^{2x} \tag{1.4}$$

2 数值结果

已知方程的解分别为

$$u_1(x)=x(1-x) \tag{2.1}$$

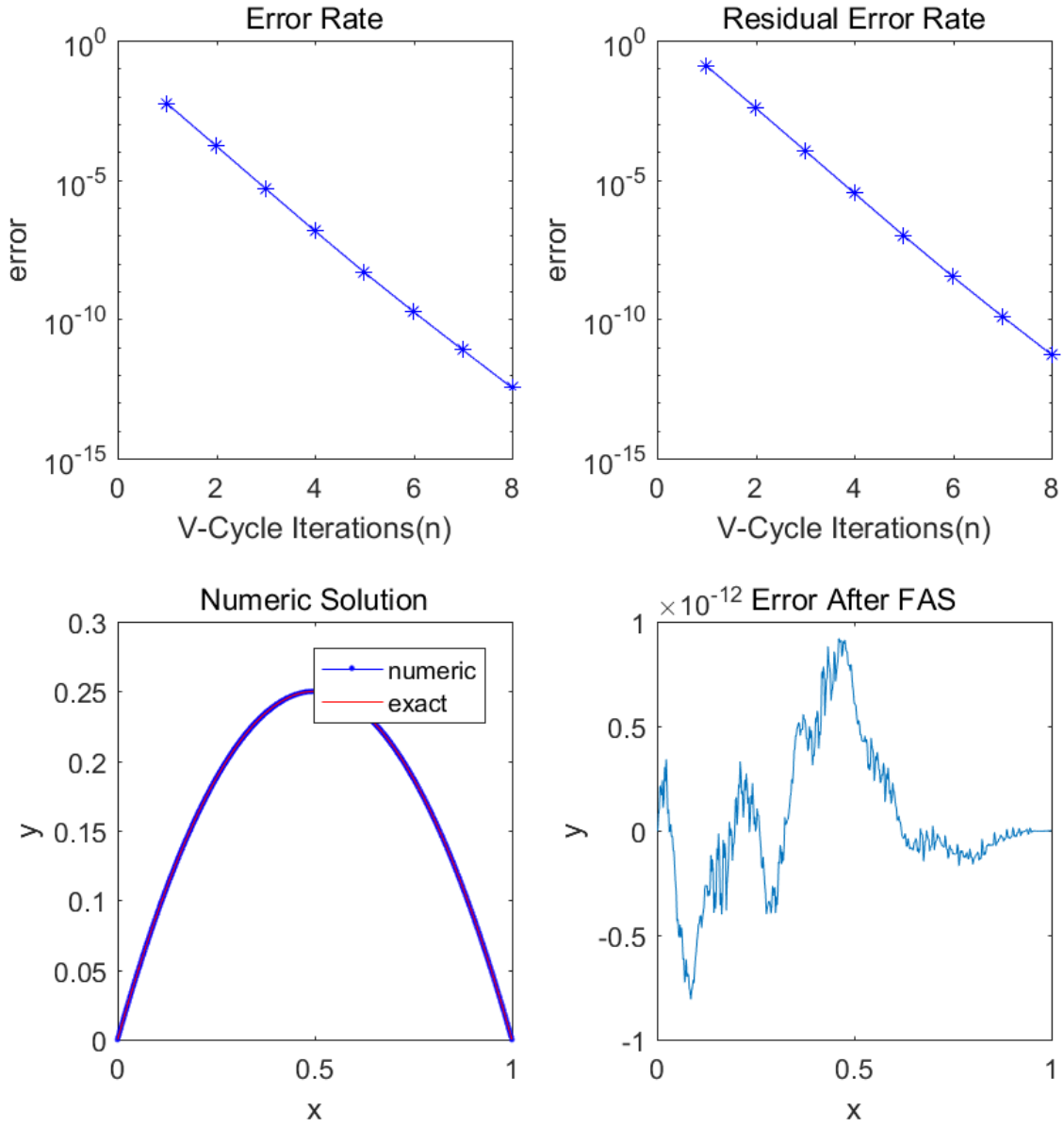
$$u_2(x)=x(1-x)\mathrm{e}^x \tag{2.2}$$

针对不同的非线性项系数 γ 对方程进行求解，计算结果如下

问题 I: $f_1(x)=2+\gamma(x-x^2)(1-2x)$

此时问题的精确解 $u_1(x)=x(1-x)$。计算结果与计算时间如下:

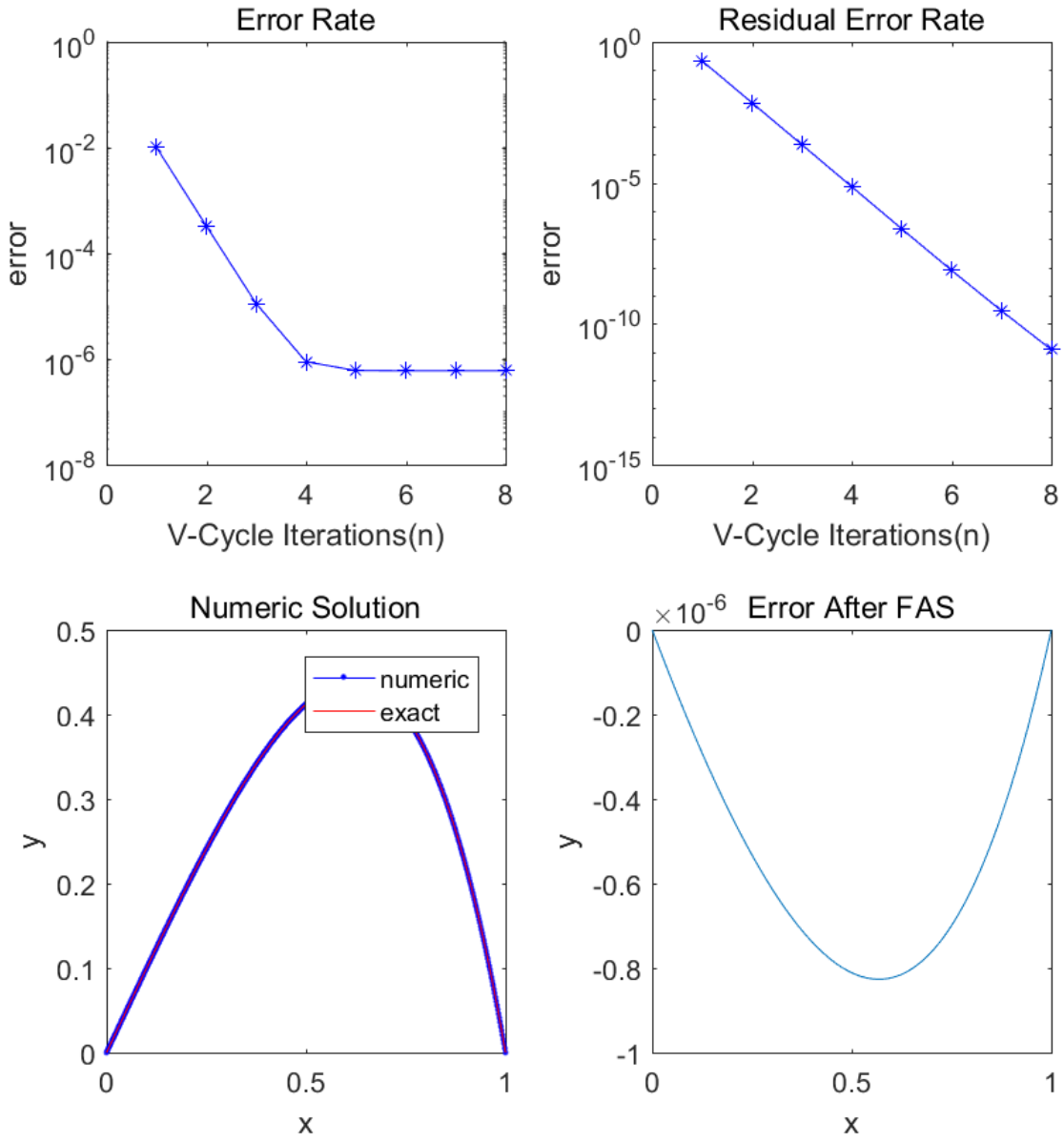
γ	0	1	10	50	70	90
V-cycle 次数 (n)	8	8	7	11	24	60
计算时间 (s)	0.118733	0.104360	0.084539	0.122061	0.247746	0.555296



问题 II: $f_2(x) = (x^2+3x)\mathrm{e}^x + \gamma(x^4-2x^2+x)\mathrm{e}^{2x}$

此时问题的精确解 $u_2(x) = x(1-x)\mathrm{e}^x$。计算结果与计算时间如下:

γ	0	1	10	25	35	39
V-cycle 次数 (n)	8	8	9	12	13	11
计算时间 (s)	0.077732	0.078832	0.095345	0.119505	0.118348	0.101846



计算环境

CPU	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
RAM	8 GB
OS	Windows 10 Home, x64
Matlab	Matlab 2016b, Version 9.1.0.441655

3 多重网格方法

多重网格方法最早由 R. P. Fedorenko[2] 于上世纪 60 年代提出并逐渐发展成型。一开始只是用于求解一般的椭圆方程，随后逐渐发展为一个通用的求解偏微分方程组的方法。从求解的方程类型来看，不仅可以对一般的线性方程进行求解，同样也可以对非线性的问题进行求解（如算例）。同样，由于最终的问题是求解代数方程，所以多重网格方法不仅可以与一般

的有限差分法结合，也可以与有限元方法结合。简而言之，首先在细网格上利用迭代法对原代数方程进行第一次求解（松弛），然后在粗网格上利用迭代法对残差方程进行第二次求解（松弛），从而将残差解反馈给细网格进行校正，这便是一次简单的多重网格的迭代过程。而通过反复的使用这一迭代过程，从而得到最终的结果。

多重网格方法的计算形式简单，其核心在于用残差方程的解对原始解进行校正，而核心的方程求解过程即便是选用简单的迭代法，也能达到较高的收敛速度。而在第五章中，通过讨论加权平均算子 I_h^{2h} 和插值算子 I_{2h}^h ，得到了一般的双网格校正方法在误差与收敛速度上的表现的合理性解释：在一次无松弛的双网格计算之后（对应书中的算子 TG p82-83），误差中的光滑模（低频模段）表现为光滑模和振荡模，但是增幅 $s_k = \sin^2(\frac{\pi k}{2n}) \approx O(\frac{k^2}{n^2}), k \ll n$，因此双网格矫正法在处理误差中的光滑项表现出色。而在针对振荡模（高频模段）同样会得到光滑模和振荡模，而增幅 $c_k = \cos^2(\frac{\pi k}{2n}) \approx O(1 - \frac{k^2}{n^2}), k \ll n$，所以无松弛的双网格校正方法并不能有效地对高频误差进行约简。

而在加入 ν 次松弛之后，不妨设 λ_k 是松弛方法的矩阵 $\mathbf{R}$ 对应的的第 k 个特征值，那么对于误差的第 k 模 $\boldsymbol{w}_k$ 有

$$TG\boldsymbol{w}_k = \lambda_k^\nu s_k \boldsymbol{w}_k + \lambda_k^\nu s_k \boldsymbol{w}_{k'} \tag{3.1}$$

$$TG\boldsymbol{w}_k = \lambda_k^\nu c_k \boldsymbol{w}_k + \lambda_k^\nu c_k \boldsymbol{w}_{k'} \tag{3.2}$$

$$k' = n - k, 1 \le k \le \frac{n}{2}$$

从而通过所选用的松弛法的收敛能力对双网格校正方法进行辅助，从而改善原有方法在高频段的收敛能力不足的缺点。同样得，其他的多重网格方法也可以用类似的方式对其收敛性与收敛阶数进行证明。

虽然多重网格方法在低频上的收敛能力很强，但是实际上很多简单的迭代法同样在低频上具有较好的收敛能力（第二章的部分例题），所以低频的强收敛能力并非多重网格方法的优势。同时注意到，实际上左右多重网格的收敛能力的方法是其核心所选择的迭代法自身的收敛能力。也就是说，选择恰当的迭代法，才是在相同大小的网格上提高其计算精度的唯一方法。另外值得注意的是，由于多重网格方法的核心在于用残差方程的解对原方程进行校正，所以其最大的优势在于计算时间的开销很少。具体而言，如 $FMG(\nu_1, \nu_2)$ 方法，选用 Gauss-Seidel 迭代法，其时间复杂度为 $O(k(\nu_1 + \nu_2)V)$，其中 k 表示 FMG 实施的次数，V 表示空间网格数。由于一般 ν_1, ν_2 均取较小的常数值，而一般 k 满足 $k \ll V$，所以实际上其时间消耗几乎与网格数成线性关系。而一般的迭代法却不能总保证迭代次数 $iter$ 满足 $iter \ll V$ ，所以往往一般的迭代法的计算速度较慢。

另外值得注意的是，在计算空间的存储上，多重网格方法并不占优势。即便是朴素的迭代法，通过一些简单的技巧，可以在求解线性方程组时，将其空间复杂度约简至 $O(V)$，其中 V 表示空间网格数。而注意到，不论是 V-Cycle ，还是 FMG ，其空间复杂度均为 $O(\beta V)$ ，其中 $\beta = 2(1 - 2^{-d})^{-1}, (V - Cycle)$，$\beta = 2(1 - 2^{-d})^{-2}, (FMG)$ ，d 表示解空间的维数。之所以将常数 β 单独列出，是因为其与解空间维数相关，而且维数越大其空间复杂度越小。另外对于维数为 1 的时候，两者分别是 $\Theta(4V)$ 和 $\Theta(8V)$ ，其空间开销几乎对应了额外加密 4 倍的朴素迭代法的消耗。但是考虑到其计算速度的优势，这些额外的空间开销完全可以忍受。

最后一点，多重网格方法可以对非线性的方程进行求解，并且能保证一定的求解精度与收敛速度。实质上，和前文类似，如果选用非线性的迭代法，如牛顿法，利用多重网格的校正格式，便可以对非线性方程进行求解，之前的数值算例便是一个很好的例子。

References

[1] Shen S. Math symbols in LaTeX[CP/OL]. Zenodo, 2017. https://doi.org/10.5281/zenodo.4120375.

[2] Fedorenko R P. A relaxation method for solving elliptic difference equations [J]. Ussr Computational Mathematics and Mathematical Physics, 1962, 1(4): 1092-1096.

[3] Briggs W L, Henson V E, Mccormick S F. A multigrid tutorial (2nd ed.)[M]. Society for Industrial and Applied Mathematics„ 2000.

[4] Oh S, Milstein A B, Bouman C A, et al. A general framework for nonlinear multigrid inversion [J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2005, 14(1): 125-40.

[5] Brandt A, Ron D. Multigrid solvers and multilevel optimization strategies[J]. Combinatorial Optimization, 2003, 14: 1-69.