

# Anderson 加速方法中的松弛参数选取\*

段晓宇<sup>1</sup>, 安恒斌<sup>2,3</sup>

1. 中国工程物理研究院研究生院, 北京 100088
2. 北京应用物理与计算数学研究所, 北京 100094
3. 中国工程物理研究院高性能数值模拟软件中心, 北京 100088

**摘要:** Anderson 加速方法是提高不动点迭代收敛速度的一类方法。该方法中的松弛参数一般在区间  $(0, 1]$  内, 且大多数选取为 1。考虑扩大 Anderson 加速方法中松弛参数的选取范围, 以求解非线性扩散方程的 Picard 迭代过程为例, 比较了不同松弛参数选取方法的效果。数值结果表明, 合适的松弛参数可以有效提高 Anderson 加速方法的效果。

**关键词:** 迭代方法; Anderson 加速; 非线性扩散方程; 参数选取

**中图分类号:** O241.7 **文献标志码:** A **文章编号:** 2097-0137 (2022) 06-0113-10

## On choice of relaxation parameter for Anderson acceleration method

DUAN Xiaoyu<sup>1</sup>, AN Hengbin<sup>2,3</sup>

1. Graduate School of CAEP, Beijing 100088, China
2. Institute of Applied Physics and Computational Mathematics, Beijing 100094, China
3. CAEP Software Center for High Performance Numerical Simulation, Beijing 100088, China

**Abstract:** Anderson acceleration method is a kind of method to improve the convergence speed of fixed-point iteration. The relaxation parameter is generally in the interval  $(0, 1]$ , and is selected as 1 in most cases. This paper expanded the selection range of the relaxation parameter. Taking the Picard iteration of solving the nonlinear diffusion equation as an example, the effect of the relaxation parameter selection method was compared. The numerical results show that the appropriate relaxation parameter can effectively improve the convergence speed of the Anderson acceleration method.

**Key words:** iterative methods; Anderson acceleration; nonlinear diffusion equations; parameter selection

非线性方程的数值求解在很多科学与工程计算领域具有广泛的应用。求解非线性问题以迭代法为主, 其中最主要的迭代法包括 Newton 迭代法和不动点迭代方法<sup>[1]</sup>。Newton 迭代方法的每步迭代中需要计算 Jacobi 矩阵, 并求解 Newton 线性方程组。Newton 迭代法的显著优点是该方法具有局部二阶收敛速度。不动点迭代法的算法实施相对简洁, 但该方法的收敛速度较慢。在很多实际应

用中, 当采用不动点迭代方法求解非线性问题时, 常常会出现迭代序列收敛缓慢甚至发散的现象。因此设计并采用有效的加速方法十分关键<sup>[2-4]</sup>。

在已有研究中, 关于向量序列的加速方法可大致分为向量外推法<sup>[5-8]</sup>和 Anderson 加速方法两类<sup>[9]</sup>。然而, 直到近期(约 2009 年之后), Anderson 加速方法才吸引更多的关注<sup>[10-12]</sup>。

Anderson 加速方法是针对不动点迭代进行加

\* 收稿日期: 2021-11-19

录用日期: 2021-12-21

网络首发日期: 2022-05-07

基金项目: 国家重点研发计划(2017YFA0603903); 国家自然科学基金(12171045, 11671051)

作者简介: 段晓宇(1997年生), 女; 研究方向: 数值并行算法; E-mail: duanxiaoyu19@gcaep.ac.cn

通信作者: 安恒斌(1974年生), 男; 研究方向: 数值并行算法及应用; E-mail: an\_hengbin@iapcm.ac.cn

速的一类算法<sup>[13-15]</sup>。在数学原理上与向量外推法不同, Anderson 加速方法不仅用到了迭代序列的信息, 也用到了迭代算子的信息。一般的 Anderson 加速迭代公式中包括函数值的线性组合和迭代值的线性组合两项, 这两项的系数分别为  $p$  和  $1-p$ , 其中要求  $0 < p \leq 1$ 。目前对松弛参数的研究较少, 并且在 Anderson 加速方法的诸多应用中往往默认为  $p = 1$ <sup>[10, 14-15]</sup>。Fang 和 Saad 考虑了松弛参数为  $p = 1$  之外的情形<sup>[16]</sup>, 但没有进一步分析参数取值对该方法的收敛速度的影响。结合具体应用, 近期部分研究者初步探讨了在区间  $(0, 1]$  内不同松弛参数  $p$  的取值对迭代方法的影响<sup>[17-18]</sup>。

以上所有的研究工作均要求松弛参数在区间  $(0, 1]$  内。本文重点关注 Anderson 加速方法中松弛参数的选取, 并且考虑适当扩大松弛参数  $p$  的选取范围。以求解非线性扩散方程的 Picard 迭代为例, 本文分析了松弛参数对 Anderson 加速方法迭代收敛速度的影响。数值实验结果表明, Anderson 加速方法中松弛参数的取值对迭代次数有较大的影响。对于不同的问题, 最小迭代次数对应的参数取值是不同的。默认的参数取值  $p = 1$  往往对应着较少的迭代次数。但在一些问题上, 可以找到使迭代次数更少的参数取值  $p$ ; 并且参数可以不受  $0 < p \leq 1$  的限制。

本文第 1 节介绍 Anderson 加速方法的主要思想及算法实现<sup>[10, 13]</sup>; 第 2 节介绍 Anderson 加速方法中松弛参数的选取及具体实施; 第 3 节给出非线性扩散方程的离散以及 Picard 迭代法的实现方式; 第 4 节通过数值实验, 分析了松弛参数的选取对 Anderson 加速方法的影响; 第 5 节给出本文的结论。

## 1 Anderson 加速方法

Anderson 加速算法是加速不动点迭代收敛的一类方法。在科学和工程计算领域, 许多问题都可以归结为不动点问题。不动点问题的一般形式如下:

$$u = G(u), \quad G: \mathbf{R}^n \rightarrow \mathbf{R}^n. \quad (1)$$

基于上述不动点方程, 可以构造如下的迭代公式。给定初始点  $u^{(0)}$ , 令

$$u^{(k+1)} = G(u^{(k)}), \quad k = 0, 1, \dots, \quad (2)$$

这种迭代方法叫做不动点迭代,  $G$  称为迭代函数。如果  $G$  是一个压缩映射, 则由不动点迭代法得到的序列可以收敛到不动点问题的精确解。

在不动点迭代构造的压缩映射  $G(u)$  的基础上, Anderson 加速方法的迭代公式可以表示为<sup>[10]</sup>

$$u^{(k+1)} = \sum_{i=0}^{m_k} G(u^{(k-m_k+i)}), \quad (3)$$

其中  $m_k = \min\{m, k\}$ , 称为 Anderson 深度。 $m_k$  决定了在迭代中利用的历史信息的数量,  $m$  的选择依赖于具体的应用。

Anderson 加速方法的思想是将第  $k+1$  步迭代点定义为第  $k-m_k, \dots, k-1, k$  步函数值  $G$  的线性组合; 线性组合的系数  $\alpha$  是通过极小化残差向量的仿射组合的范数确定的, 即求解约束最小二乘问题<sup>[13]</sup>

$$\min_{\alpha = (\alpha_0, \dots, \alpha_{m_k})^T} \|F_k \alpha\|_2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1, \quad (4)$$

其中  $F_k = (f_{k-m_k}, \dots, f_k)$ , 残差  $f_k = G(u^{(k)}) - u^{(k)}$ 。由于不动点问题的精确解  $x^*$  满足  $f(x^*) = 0$ , 所以通过极小化残差序列的组合向量范数定义组合系数。

为了方便求解上面的约束优化问题, 将式(4)转化为无约束最小二乘问题<sup>[10-11, 13]</sup>

$$\min_{\gamma = (\gamma_0, \dots, \gamma_{m_k-1})^T} \|f_k - \mathcal{F}_k \gamma\|_2, \quad (5)$$

其中  $\mathcal{F}_k = (\Delta f_{k-m_k}, \dots, \Delta f_{k-1})$ ,  $\Delta f_i = f_{i+1} - f_i$ ,  $i = k-m_k, \dots, k-1$ , 即采用残差向量的增量定义最小二乘问题。系数  $\gamma$  和  $\alpha$  满足

$$\alpha_i = \begin{cases} \gamma_0, & i = 0, \\ \gamma_i - \gamma_{i-1}, & 1 \leq i \leq m_k - 1, \\ 1 - \gamma_{m_k-1}, & i = m_k. \end{cases} \quad (6)$$

基于无约束最小二乘问题, 可得到 Anderson 加速方法的另一种表达形式

$$u^{(k+1)} = G(u^{(k)}) - g_k \gamma^{(k)}, \quad (7)$$

其中  $g_k = (\Delta G_{k-m_k}, \dots, \Delta G_{k-1})$ ,  $\Delta G_i = G(u^{(i+1)}) - G(u^{(i)})$ ,  $i = k-m_k, \dots, k-1$ 。公式(3)中采用函数值的线性组合定义迭代点, 经过转换之后, 公式(7)中新的迭代点由相应的函数值的增量定义。

在 Anderson 加速方法中, 函数值增量矩阵  $\mathcal{F}$  与残差增量矩阵  $g$  的列向量个数最多不超过  $m$  个。在第  $k$  步迭代, 如果  $k < m$ , 那么在上一步迭代矩阵的右端需要添加最新的向量; 如果  $k \geq m$ , 那么在添加新向量的同时, 还需要删除矩阵中最左端的向量<sup>[13]</sup>。在 Anderson 加速方法中, 采用 QR 分解求解无约束最小二乘问题。Anderson 加速不动点迭代的算法描述如表 1。

表1 Anderson加速不动点迭代的算法-1 ( $p=1$ )  
Table 1 Anderson accelerated fixed point iteration algorithm ( $p=1$ )

---

Input: 迭代初值  $u^{(0)}$  和 Anderson 加速深度的参数  $m > 0$   
Output: 非线性问题的近似解  $u$

1. 令  $u^{(1)} = G(u^{(0)})$
2. for  $k = 1, 2, \dots$ , do
3.      $m_k = \min\{m, k\}$
4.     计算  $G(u^{(k)})$ ,  $f_k = G(u^{(k)}) - u^{(k)}$ ,  $\Delta f_{k-1} = f_k - f_{k-1}$ ,  $\Delta G_{k-1} = G(u^{(k)}) - G(u^{(k-1)})$
5.     if  $m_k \leq m$  then
6.          $g_k = [g_{k-1}, \Delta G_{k-1}]$ ,  $\mathcal{F}_k = [\mathcal{F}_{k-1}, \Delta f_{k-1}] = Q_k R_k$
7.     else
8.          $g_k = [g_{k-1}(:, 2:m), \Delta G_{k-1}]$ ,  $\mathcal{F}_k = [\mathcal{F}_{k-1}(:, 2:m), \Delta f_{k-1}] = Q_k R_k$
9.     end if
10.     求解最小二乘问题  $\min_{\gamma = (\gamma_0, \dots, \gamma_{m_k-1})^T} \|f_k - \mathcal{F}_k \gamma\|_2$ , 确定极小化问题  $\gamma^{(k)} = (\gamma_0^{(k)}, \dots, \gamma_{m_k-1}^{(k)})^T$
11.     更新  $u^{(k+1)} = G(u^{(k)}) - g_k \gamma^{(k)}$
12. end for
13. return  $u^{(k+1)}$

---

在迭代公式(3)的基础上, 考虑更加一般的 Anderson 加速形式, 即<sup>[10]</sup>

$$u^{(k+1)} = (1-p) \sum_{i=0}^{m_k} \alpha_i^{(k)} u^{(k-m_k+i)} + \sum_{i=0}^{m_k} \alpha_i^{(k)} G(u^{(k-m_k+i)}), \quad (8)$$

其中  $p$  是松弛系数, 且  $0 < p \leq 1$ 。迭代公式由迭代点  $u$  的线性组合和函数值  $G(u)$  的线性组合两部分组成, 也就是在迭代公式(3)的基础上加入了历史迭代点的信息。显然, 当  $p=1$  时算法流程如表1所示。目前, 很多应用中均选取  $p=1$ <sup>[11, 19-23]</sup>。

## 2 Anderson 加速方法中松弛参数的选取

在 Anderson 加速迭代一般形式(8)的基础上, 探讨 Anderson 加速方法中松弛参数对迭代方法的影响, 并且适当扩大参数  $p$  的取值范围, 使其不受  $0 < p \leq 1$  的限制。

为了方便讨论, 首先将公式(8)中迭代点的加权平均  $\bar{u}^{(k)}$  和函数值的加权平均  $\bar{G}^{(k)}$  记为

$$\begin{aligned} \bar{u}^{(k)} &= \sum_{i=0}^{m_k} \alpha_i^{(k)} u^{(k-m_k+i)}, \\ \bar{G}^{(k)} &= \sum_{i=0}^{m_k} \alpha_i^{(k)} G(u^{(k-m_k+i)}). \end{aligned} \quad (9)$$

那么迭代公式可写为

$$u^{(k+1)} = (1-p)\bar{u}^{(k)} + p\bar{G}^{(k)}. \quad (10)$$

通过调整  $p$  的取值, 分析参数取值对 Anderson 加速方法的影响。由式(9)可知,  $\bar{G}^{(k)}$  实际上是 Anderson 加速算法1(参数  $p=1$ ) 计算得到的迭代值, 可以记  $\bar{G}^{(k)} = u_{AA1}^{(k+1)}$ 。于是, 迭代公式进一步写为

$$u^{(k+1)} = (1-p)\bar{u}^{(k)} + pu_{AA1}^{(k+1)}. \quad (11)$$

参考 Anderson 加速算法中的思想, 新的迭代

点  $u^{(k+1)} = \sum_{i=0}^{m_k} \alpha_i^{(k)} G(u^{(k-m_k+i)})$  可以转化为增量的形式  $u^{(k+1)} = G(u^{(k)}) - g_k \gamma^{(k)}$  进行计算; 将  $\bar{u}^{(k)}$  与  $u^{(k+1)}$  类比, 由于表达式中参数相同, 可将  $u^{(k+1)}$  表示为

$$\bar{u}^{(k)} = u^{(k)} - \mathcal{M}_k \gamma^{(k)}, \quad (12)$$

其中  $\mathcal{M}_k = (\Delta u^{(k-m_k)}, \dots, \Delta u^{(k-1)})$ ,  $\Delta u^{(i)} = u^{(i+1)} - u^{(i)}$  表示迭代点的增量。又因为  $u^{(k)} = G(u^{(k)}) - f_k$ ,  $\mathcal{M}_k = g_k - \mathcal{F}_k$ , 代入式(12)可得

$$\begin{aligned} \bar{u}^{(k)} &= G(u^{(k)}) - f_k - (g_k - \mathcal{F}_k) \gamma^{(k)} \\ &= u_{AA1}^{(k+1)} - (f_k - \mathcal{F}_k) \gamma^{(k)}. \end{aligned} \quad (13)$$

将式(13)代入式(11), 得到迭代公式的最终形式为

$$u^{(k+1)} = u_{AA1}^{(k+1)} - (1-p)(f_k - \mathcal{F}_k) \gamma^{(k)}. \quad (14)$$

于是, 含松弛参数的 Anderson 加速算法可描述如表2。

表 2 Anderson 加速不动点迭代的算法-2

Table 2 Anderson accelerated fixed point iteration algorithm

---

Input: 迭代初值  $u^{(0)}$  和 Anderson 加速深度的参数  $m > 0$

Output: 非线性问题的近似解  $u$

1. 令  $u^{(1)} = G(u^{(0)})$
2. for  $k = 1, 2, \dots$  do
3.      $m_k = \min\{m, k\}$
4.     计算  $G(u^{(k)})$ ,  $f_k = G(u^{(k)}) - u^{(k)}$ ,  $\Delta f_{k-1} = f_k - f_{k-1}$ ,  $\Delta G_{k-1} = G(u^{(k)}) - G(u^{(k-1)})$
5.     计算  $g_k$  和 QR 分解  $\mathcal{F}_k = Q_k R_k$
6.     求解最小二乘问题得  $\gamma^{(k)}$
7.     令  $u^{(k+1)} = G(u^{(k)}) - g_k \gamma^{(k)}$
8.     更新  $u^{(k+1)} = u^{(k+1)} - (1-p)(f_k - \mathcal{F}_k \gamma^{(k)})$
9. end for
10. return  $u^{(k+1)}$

---

### 3 非线性扩散方程的离散及求解

考虑二维区域  $\Omega = (0, 1) \times (0, 1)$  上的扩散方程

$$-\nabla \cdot (\kappa(u) \nabla u) = f(x, y), \quad (x, y) \in \Omega, \quad (15)$$

其中  $\kappa(u)$  为扩散系数,  $f(x, y)$  为源项。如果  $\kappa$  与  $u$  有关, 偏微分方程(15)为非线性方程

$$-\frac{\partial}{\partial x} \left( \kappa(u) \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( \kappa(u) \frac{\partial u}{\partial y} \right) = f(x, y).$$

将区域  $\Omega$  划分为  $N \times N$  的均匀网格, 对于方程(15), 扩散算子采用五点有限差分离散。令  $u_{i,j}$  表示  $u(x_i, y_j)$  的近似, 则  $(x_i, y_j)$  处的离散形式为

$$\begin{aligned} & -\frac{1}{h^2} \left[ \kappa(u_{i+\frac{1}{2},j})(u_{i+1,j} - u_{i,j}) \right. \\ & \quad \left. - \kappa(u_{i-\frac{1}{2},j})(u_{i,j} - u_{i-1,j}) \right] \\ & - \frac{1}{h^2} \left[ \kappa(u_{i,j+\frac{1}{2}})(u_{i,j+1} - u_{i,j}) \right. \\ & \quad \left. - \kappa(u_{i,j-\frac{1}{2}})(u_{i,j} - u_{i,j-1}) \right] = f_{i,j}, \end{aligned} \quad (16)$$

对于式(16)中扩散系数在半点处的值, 我们用两端点处的平均值代替半点处的函数值, 即

$$\begin{cases} \kappa(u_{i+\frac{1}{2},j}) = \kappa\left(\frac{u_{i+1,j} + u_{i,j}}{2}\right), \\ \kappa(u_{i,j+\frac{1}{2}}) = \kappa\left(\frac{u_{i,j+1} + u_{i,j}}{2}\right). \end{cases} \quad (17)$$

将未知量按自然序排列, 将式(16)写成

$F(u) = 0$  的形式,  $F$  为非线性方程的残差向量。

采用 Picard 迭代法求解非线性扩散方程<sup>[24]</sup>。Picard 迭代的关键是构造一个压缩映射  $G$ , 把非线性扩散方程的求解问题转化为映射  $G$  的不动点问题。用  $u^{(k+1)}$  表示下一个迭代点, 构造 Picard 迭代

$$u^{(k+1)} = -[\nabla \cdot (\kappa(u^{(k)}) \nabla)]^{-1} f \equiv G(u^{(k)}), \quad k = 0, 1, 2, \dots \quad (18)$$

为了表示方便, 把函数  $G(u)$  定义为

$$G(u) = -[\nabla \cdot (\kappa(u) \nabla)]^{-1} f. \quad (19)$$

第  $k+1$  步非线性迭代对应的 Picard 迭代线性化方程组为

$$-\nabla \cdot (\kappa(u^{(k)}) \nabla u^{(k+1)}) = f. \quad (20)$$

计算第  $k+1$  步迭代点时, 扩散系数用第  $k$  步的结果表示。同样, 使用五点有限差分法离散扩散算子。对于网格中的任意节点  $i, j = 1, 2, \dots, N$ , 离散格式为

$$\begin{aligned} & \frac{1}{h^2} \left[ \kappa\left(\frac{u_{i+1,j}^{(k)} + u_{i,j}^{(k)}}{2}\right)(u_{i+1,j}^{(k+1)} - u_{i,j}^{(k+1)}) \right. \\ & \quad \left. - \kappa\left(\frac{u_{i,j}^{(k)} + u_{i-1,j}^{(k)}}{2}\right)(u_{i,j}^{(k+1)} - u_{i-1,j}^{(k+1)}) \right] \\ & + \frac{1}{h^2} \left[ \kappa\left(\frac{u_{i,j+1}^{(k)} + u_{i,j}^{(k)}}{2}\right)(u_{i,j+1}^{(k+1)} - u_{i,j}^{(k+1)}) \right. \\ & \quad \left. - \kappa\left(\frac{u_{i,j}^{(k)} + u_{i,j-1}^{(k)}}{2}\right)(u_{i,j}^{(k+1)} + u_{i,j-1}^{(k+1)}) \right] = -f_{i,j}. \end{aligned} \quad (21)$$

即构成一个  $N^2$  阶线性方程组, 方程组的解就



收敛; 当  $p$  在 1 附近变化时, 迭代次数相对较少。分别选取模型 1 和模型 2 的两类扩散系数的数值结果进行展示。其中, 模型 1 选取  $\beta = 1$  和 5 的情况,

迭代次数随参数变化的结果如图 1 所示; 模型 2 选取  $\gamma = 0.75$  和 1.00 的情况, 结果如图 2 所示。

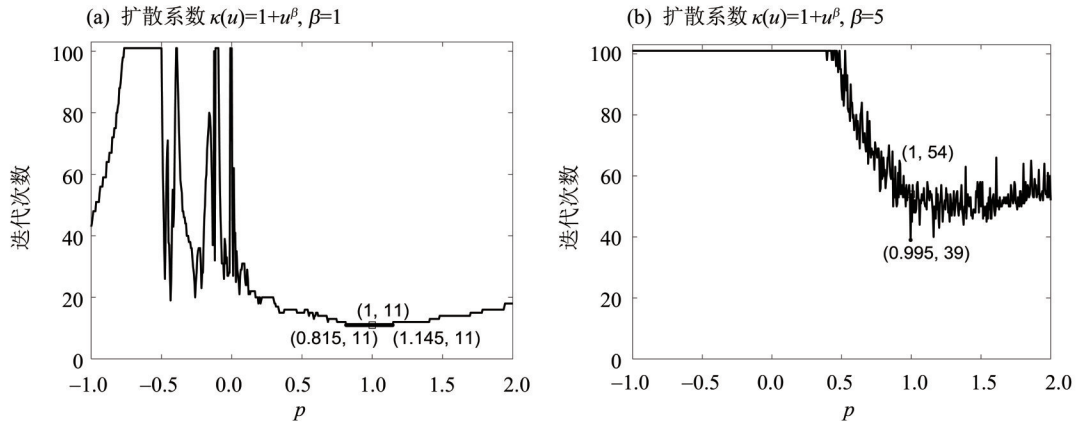


图 1 Anderson 加速中松弛参数选取对迭代次数的影响(模型 1)

Fig. 1 Influence of relaxation parameter choice on iteration number of Anderson acceleration method (model 1)

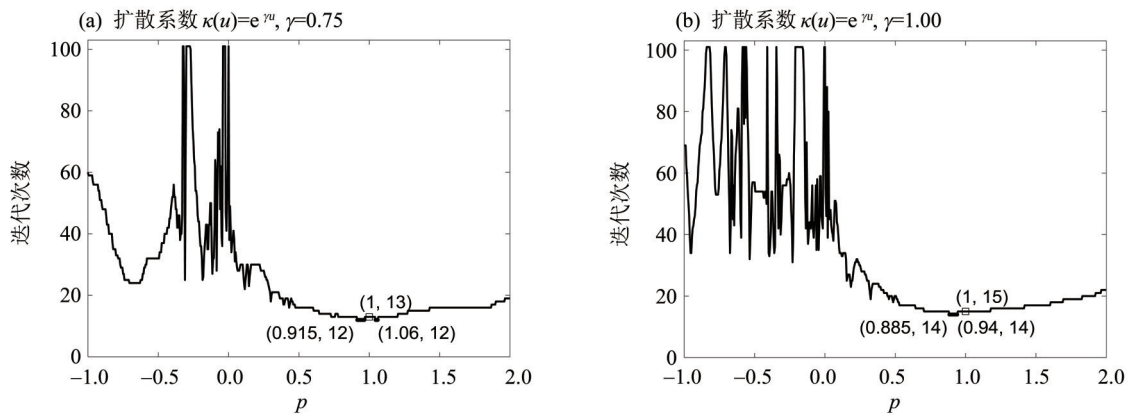


图 2 Anderson 加速中松弛参数选取对迭代次数的影响(模型 2)

Fig. 2 Influence of relaxation parameter choice on iteration number of Anderson acceleration method (model 2)

图 1 给出了在网格规模为  $64 \times 64$  下, 求解非线性扩散方程模型 1 时  $p$  对迭代次数的影响。图 1 (a) 说明, 当  $\beta = 1$  时, Anderson 加速-1 需要迭代 11 次; 扩大  $p$  的取值范围后, 当  $p \in [0.815, 1.145]$  时, 迭代次数仍为 11。其中  $p > 1$  时, 意味着迭代值这一项的系数  $1 - p$  为负数。此外, 当  $p$  为负数时, 迭代次数较多甚至不能收敛。图 1(b) 展示了  $\beta = 5$  时的求解情况, 曲线前半段说明迭代不能收敛, 曲线后半段说明了迭代次数随参数  $p$  的变化震荡, Anderson 加速-1 的迭代次数为 54, 有一些取值能够使迭代次数减少; 当  $p = 0.995$  时, 迭代次数最少为 39, 相对 Anderson 加速-1 的迭代次数而言, 减少了 1/4 以上。

收敛; 当  $p \in [0.5, 2.0]$  时, 迭代次数相对较少, 并且均能找到使迭代次数更少的参数  $p$  的取值。图 2(a) 展示了  $\gamma = 0.75$  的情况, Anderson 加速-1 的迭代次数为 13,  $p$  在 1 左右一小段区间能使迭代次数减少 1 次。图 2(b) 是  $\gamma = 1.0$  的情况, Anderson 加速-1 的迭代次数为 15; 而当  $p \in [0.885, 0.94]$  时, 迭代次数为 14。

图 2 给出了求解第二类模型的数值结果。从图中可以看出, 当  $p$  为负数时, 迭代次数较多甚至不

收敛。综上所述,  $p$  的选取对收敛效果的影响较大。但参数  $p$  的影响不是单调的, 对于非线性强的问题, 迭代次数的变化十分震荡。Anderson 加速-1 中, 参数  $p = 1$  往往对应着较少的迭代次数。在一些情形下, 还可以找更好的  $p$  取值, 使迭代次数更少。并且, 参数  $p$  可以不受  $0 < p < 1$  的限制; 对于一些问题, 可以在  $p > 1$  的情况找到使迭代次数最

少的参数取值。

在小规模情况下对 Anderson 加速方法进行不同松弛参数选取测试, 得到了参数选取对方法效果有一定影响的结论。进一步, 选定一些具有代表性的参数值, 应用到较大规模的非线性扩散方程求解中。选择3种典型取值, 记为  $p_1$ ,  $p_2$  和  $p_3$ , 参数  $p$  的选取准则是能够使迭代次数较少或  $p$  值较为远离 1。两类模型的典型参数取值及相应的迭代次数 IT 见表 4, 表 4 中给出了规模为  $64 \times 64$  的情况下, Anderson 加速中松弛参数的典型取值。

#### 4.2 典型松弛参数的 Anderson 加速方法在较大规模问题的上应用

根据 4.1 节的分析, 选出较优的松弛参数, 并比较 Anderson 加速方法和 Picard 迭代在求解更大规模问题上的应用效果。本节中, 网格规模为  $128 \times 128$ ,  $256 \times 256$  和  $512 \times 512$ 。不同算例中, 固定典型参数取值用于测试, 统计选取固定典型参数求解模型问题的迭代次数和求解时间。求解模型 1 时, 扩散系数中的参数  $\beta$  分别取 1, 3 和 5, 对应的数值结果见表 5~7。为方便表述,  $N = 128$  表示网格规模为  $128 \times 128$ 。

表 4 两类模型的典型参数取值及相应的迭代次数

Table 4 Typical parameter values of the two types of models and the corresponding number of iterations

模型	$\beta/\gamma$	AA (1)		AA (2)					
		$p$	IT	$p_1$	IT	$p_2$	IT	$p_3$	IT
模型 1	1	1	11	0.82	11	1.10	11	1.14	11
	3	1	17	0.84	18	0.97	17	1.05	17
	5	1	54	0.995	39	1.16	47	1.32	45
模型 2	0.75	1	13	0.80	13	0.92	12	1.06	12
	1.00	1	15	0.89	14	0.93	14	1.10	15
	1.25	1	16	0.81	16	1.10	16	0.90	16

表 5 不同方法求解模型 1 ( $\kappa = 1 + u^\beta, \beta = 1$ ) 的迭代次数和时间

Table 5 Iteration numbers and solution time for model 1 ( $\kappa = 1 + u^\beta, \beta = 1$ ) by different methods

项目	$N = 128$		$N = 256$		$N = 512$	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	18	1.99	21	39.33	23	415.05
AA(1) ( $p = 1.00$ )	11	2.04	11	35.46	12	331.17
AA(2) ( $p = 0.82$ )	12	1.80	12	34.96	13	355.06
AA(2) ( $p = 1.10$ )	12	1.66	12	31.25	12	336.35
AA(2) (1.14)	12	1.67	12	29.93	13	370.61

表 6 不同方法求解模型 1 ( $\kappa = 1 + u^\beta, \beta = 3$ ) 的迭代次数和时间

Table 6 Iteration numbers and solution time for model 1 ( $\kappa = 1 + u^\beta, \beta = 3$ ) by different methods

项目	$N = 128$		$N = 256$		$N = 512$	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	41	5.80	49	145.90	52	1454.43
AA(1) ( $p = 1.00$ )	18	3.48	18	58.47	19	526.01
AA(2) ( $p = 0.84$ )	19	3.20	20	66.76	20	556.22
AA(2) ( $p = 0.97$ )	18	3.25	19	63.54	19	539.06
AA(2) ( $p = 1.05$ )	18	3.11	18	60.62	19	540.87

表 7 不同方法求解模型 1( $\kappa = 1 + u^\beta, \beta = 5$ )的迭代次数和时间Table 7 Iteration numbers and solution time for model 1 ( $\kappa = 1 + u^\beta, \beta = 5$ ) by different methods

项目	N = 128		N = 256		N = 512	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	110	18.99	127	413.84	130	3 163.88
AA(1) ( $p = 1.00$ )	56	9.93	52	175.16	56	1 335.15
AA(2) ( $p = 0.995$ )	51	9.25	52	189.03	56	1 386.53
AA(2) ( $p = 1.16$ )	54	10.42	46	164.07	48	1 381.87
AA(2) ( $p = 1.32$ )	47	7.01	47	107.08	52	682.18

表 5~7 均列出了 3 种网格规模下, 分别用 Picard 迭代、3 种松弛参数取值的 Anderson 加速求解模型 1 的迭代次数和时间, 3 个表分别对应 3 种扩散系数的情况。从表中可以看出, 对于同一个问题, 网格规模越大、计算量越大, 计算时间显著增加。对比 Picard 迭代, 不同网格规模下 Anderson 加速方法均可减少约一半甚至一半以上的迭代次数。从求解时间来看, Anderson 加速方法求解更快。随着  $\beta$  的增大, 即问题的非线性增强, Anderson 加速和 Picard 迭代的迭代次数均增加。

在小规模问题中使迭代次数最少的参数取值  $p$ , 用于大规模问题的求解时, 迭代次数不一定最

少。由表 5~6 可以看出, 当  $\beta = 1$  和 3 时, 迭代次数和 Anderson 加速-1 的迭代次数相差不超过 1; 由表 7 可知, 当  $\beta = 5$  时,  $p$  的 3 种取值均能减少 Anderson 加速-1 的迭代次数。

表 8~10 展示了不同方法求解模型 2 的结果, 三个表分别对应扩散系数为  $\gamma = 0.75, 1.00$  和 1.25 的情况。从表中可以看出, 同求解模型 1 时基本类似, 当模型问题的非线性越强, 迭代次数越多; 对于同一问题, 规模越大, 迭代次数增加或不变。由表 8~9 可知, 当扩散项参数为  $\gamma = 0.75$  和 1.00 时, Anderson 加速的迭代次数减少了一半以上; 由表 10 可知, 当扩散项参数为 1.25 时, 加速后的

表 8 不同方法求解模型 2( $\kappa = e^{\gamma u}, \gamma = 0.75$ )的迭代次数和时间Table 8 Iteration numbers and solution time for model 2 ( $\kappa = e^{\gamma u}, \gamma = 0.75$ ) by different methods

项目	N = 128		N = 256		N = 512	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	24	2.35	28	46.00	30	508.73
AA(1) ( $p = 1.00$ )	13	2.32	13	43.46	14	368.58
AA(2) ( $p = 0.80$ )	13	2.27	14	46.72	14	398.36
AA(2) ( $p = 0.92$ )	13	2.44	13	44.55	13	368.44
AA(2) ( $p = 1.06$ )	13	2.65	14	46.83	14	392.39

表 9 不同方法求解模型 2( $\kappa = e^{\gamma u}, \gamma = 1.0$ )的迭代次数和时间Table 9 Iteration numbers and solution time for model 2 ( $\kappa = e^{\gamma u}, \gamma = 1.0$ ) by different methods

项目	N = 128		N = 256		N = 512	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	31	4.39	36	96.16	39	1103.37
AA(1) ( $p = 1.00$ )	15	2.10	16	39.37	16	348.42
AA(2) ( $p = 0.89$ )	15	2.35	15	49.56	16	449.79
AA(2) ( $p = 0.93$ )	15	2.52	15	49.39	16	439.94
AA(2) ( $p = 1.10$ )	15	2.43	15	49.86	16	454.75

迭代次数大约是 Picard 迭代次数的 1/3。从求解时间上看, Anderson 加速方法求解更快。松弛参数  $p$  的取值使迭代次数与 Anderson 加速-1 的迭代次数

相差不超过 1; 且  $p$  距离 1 较远、 $1 - p$  为负数的情况, 也能使迭代收敛。

表 10 不同方法求解模型 2 ( $\kappa = e^{\gamma n}, \gamma = 1.25$ ) 的迭代次数和时间

Table 10 Iteration numbers and solution time for model 2 ( $\kappa = e^{\gamma n}, \gamma = 1.25$ ) by different methods

项目	$N = 128$		$N = 256$		$N = 512$	
	IT	CPU/s	IT	CPU/s	IT	CPU/s
Picard	40	6.32	47	156.06	50	1401.75
AA(1) ( $p = 1.00$ )	17	2.31	17	42.00	17	378.18
AA(2) ( $p = 0.81$ )	17	2.78	17	50.59	18	492.20
AA(2) ( $p = 0.90$ )	17	2.15	17	42.15	18	373.10
AA(2) ( $p = 1.10$ )	17	2.70	17	54.49	18	499.71

## 5 结 论

本文探讨了 Anderson 松弛参数的选取, 并适当扩大参数  $p$  的范围, 将其应用到二维非线性扩散方程的求解中。首先, 求解网格规模为  $64 \times 64$  的问题, 分析数值结果、选取典型参数取值; 然后, 将其应用到网格规模为  $128 \times 128$ ,  $256 \times 256$  和  $512 \times 512$  的问题中。通过数值实验, 得出如下主要结论:

1)  $p$  的取值对 Anderson 迭代次数有影响。当  $p$  接近于 0 或为负数时, 迭代次数很多甚至不收敛。当  $p \in [0.5, 2.0]$  时, 迭代次数相对少一些。当  $p$  在 1 附近时, 往往对应着最少的迭代次数; 但对于非线性强的问题, 迭代次数较为震荡。

2) Anderson 加速中默认  $p = 1$ , 这一取值往往对应着少的迭代次数。

3) 合适的松弛参数可以有效提高 Anderson 加速方法的效果。在很多情况下, 可以找到使迭代次数比 Anderson 加速-1 更少的参数  $p$ , 最优的松弛参数选取依赖于具体的问题; 对于非线性强的问题, 不同的参数取值对应的迭代次数相差较大, 更有可能找到使迭代次数大幅度减少的松弛参数  $p$ 。

4) 参数  $p$  的选取可以不受传统  $0 < p \leq 1$  的限制。在  $1 < p \leq 2$  的范围内, 迭代也均能收敛。对于一些问题, 甚至可以在  $p > 1$  的情况找到迭代次数最少的参数取值。

## 参考文献:

- [1] KNOLL D A, KEYES D E. Jacobian-free Newton-Krylov methods: A survey of approaches and applications [J]. Journal of Computational Physics, 2004, 193(2): 357-397.
- [2] PUGACHEV B P. Acceleration of the convergence of iterative processes and a method of solving systems of non-linear equations [J]. USSR Computational Mathematics and Mathematical Physics, 1977, 17(5): 199-207.
- [3] WYNN P. Acceleration techniques for iterated vector and matrix problems [J]. Mathematics of Computation, 1962, 16(79): 301-322.
- [4] BREZINSKI C, REDIVO-ZAGLIA M, SAAD Y. Shanks sequence transformations and Anderson acceleration [J]. SIAM Review, 2018, 60(3): 646-669.
- [5] JBILOU K, SADOK H. Vector extrapolation methods. Applications and numerical comparison [J]. Journal of Computational and Applied Mathematics, 2000, 122(1/2): 149-165.
- [6] BREZINSKI C. Recursive interpolation, extrapolation and projection [J]. Journal of computational and applied mathematics, 1983, 9(4): 369-376.
- [7] JBILOU K, SADOK H. Some results about vector extrapolation methods and related fixed-point iterations [J]. Journal of Computational and Applied Mathematics, 1991, 36(3): 385-398.
- [8] BREZINSKI C, REDIVO-ZAGLIA M. Extrapolation methods for the numerical solution of nonlinear Fred-

- holm integral equations [J]. *Journal of Integral Equations and Applications*, 2019, 31(1): 29–57.
- [9] ANDERSON D G. Iterative procedures for nonlinear integral equations [J]. *Journal of the ACM*, 1965, 12(4): 547–560.
- [10] WALKER H F, NI P. Anderson acceleration for fixed–point iterations [J]. *SIAM Journal on Numerical Analysis*, 2011, 49(4): 1715–1735.
- [11] AN H, JIA X, WALKER H F. Anderson acceleration and application to the three–temperature energy equations [J]. *Journal of Computational Physics*, 2017, 347: 1–19.
- [12] SHI W, SONG S, WU H, et al. Regularized Anderson Acceleration for off–policy deep reinforcement learning [J]. *Advances in Neural Information Processing Systems*, 2019, 32: 10231–10241.
- [13] WALKER H F. Anderson acceleration: Algorithms and implementations [R]. MS–6–15–50, Worcester Polytechnic Institute Mathematical Sciences Department, 2011.
- [14] POTRA F A, ENGLER H. A characterization of the behavior of the Anderson acceleration on linear problems [J]. *Linear Algebra and its Applications*, 2013, 438(3): 1002–1011.
- [15] EVANS C, POLLOCK S, REBHOLZ L G, et al. A proof that Anderson acceleration improves the convergence rate in linearly converging fixed–point methods (but not in those converging quadratically) [J]. *SIAM Journal on Numerical Analysis*, 2020, 58(1): 788–810.
- [16] FANG H, SAAD Y. Two classes of multisection methods for nonlinear acceleration [J]. *Numerical Linear Algebra with Applications*, 2009, 16(3): 197–221.
- [17] POLLOCK S, REBHOLZ L G. Anderson acceleration for contractive and noncontractive operators [J]. *IMA Journal of Numerical Analysis*, 2021, 41(4): 2841–2872.
- [18] AKSENOV V, CHERTOV M, SINKOV K. Application of accelerated fixed–point algorithms to hydrodynamic well–fracture coupling [J]. *Computers and Geotechnics*, 2021, 129: 103783.
- [19] YANG Y. Anderson acceleration for seismic inversion [J]. *Geophysics*, 2021, 86(1): R99–R108.
- [20] BIAN W, CHEN X, KELLEY C T. Anderson acceleration for a class of nonsmooth fixed–point problems [J]. *SIAM Journal on Scientific Computing*, 2021, 43(5): S1–S20.
- [21] FILIPPINI M, ALOTTO P, GIUST A. Anderson acceleration for electromagnetic nonlinear problems [J]. *COMPEL—the International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 2019, 38(5): 1493–1506.
- [22] PENG Y, DENG B, ZHANG J, et al. Anderson acceleration for geometry optimization and physics simulation [J]. *ACM Transactions on Graphics*, 2018, 37(4): 1–14.
- [23] LOTT P A, WALKER H F, WOODWARD C S, et al. An accelerated Picard method for nonlinear systems related to variably saturated flow [J]. *Advances in Water Resources*, 2012, 38: 92–101.
- [24] 秦子康, 安恒斌, 王新玉. 两类向量序列加速收敛方法比较 [J]. *数值计算与计算机应用*, 2021, 42(4): 379–394.

(责任编辑 王海蓉)

(终校 本刊编辑部)