

基于LU分解的阻尼谱修正迭代法在病态线性方程组中的应用

莫春鹏, 覃柏英*

(广西科技大学 理学院, 广西 柳州 545006)

摘要: 基于阻尼谱修正迭代法, 结合矩阵LU分解和新数值迭代方式, 提出了基于矩阵LU分解的阻尼谱修正迭代法, 将其应用于病态线性方程组的求解. 采用经典算例, 探讨矩阵LU分解和新数值迭代方式对阻尼谱修正迭代法求解病态线性方程组的性能影响. 结果表明, 矩阵LU分解和新数值迭代方式都可提高阻尼谱修正迭代法求解病态线性方程组的精度, 且提出的算法可提高高维病态线性方程组求解的精度.

关键词: LU分解; 谱修正迭代法; 病态矩阵; 线性方程组

中图分类号: O151.2; O241.6 **DOI:** 10.16375/j.cnki.cn45-1395/t.2021.03.019



0 引言

线性方程组是一类常见的数学模型, 科学技术领域中的许多实际问题都可将其转化为该模型^[1-2]. 若系数矩阵的条件数较大, 则该矩阵是病态的, 对应的线性方程组常称病态线性方程组. 对于病态线性方程组的求解, 传统算法常难以保证所求数值解的精度^[3-5]. 因此, 研究病态线性方程组的求解, 以提高求解的准确性和数值计算的效率, 具有重要的意义和价值.

目前, 研究者提出了多种数值算法求解病态线性方程组^[6-15]. 王新洲等^[14]提出了谱修正迭代法. 该算法借助系数矩阵病态性的改善, 使病态线性方程组求解的准确性得以提高, 但该算法的数值计算效率较低. 由此, 邓兴升等^[16]提出了阻尼谱修正迭代法. 该算法虽可提高数值计算的效率, 但能否收敛至病态线性方程组的真解, 与逆矩阵求解和数值迭代等方式有极大关系.

因此, 本文改进阻尼谱修正迭代法的数值迭代方式, 并采用LU分解避免直接求解矩阵的逆矩阵, 提出基于矩阵LU分解的新阻尼谱修正迭代法.

采用4个经典算例, 探讨该算法求解病态线性方程组的准确性和效率.

1 阻尼谱修正迭代法

将病态线性方程组表示为:

$$AX = b \quad (1)$$

已知 $m \times n$ 实系数矩阵 A 和 $m \times 1$ 实常数项 b . 需求解该方程组的 $n \times 1$ 解向量 X .

为了将系数矩阵统一为实对称矩阵, 可将上述方程组转化为如下线性方程组:

$$A^T AX = A^T b \quad (2)$$

设 $B = A^T A$ 和 $H = A^T b$, 则 B 为 $n \times n$ 的实对称矩阵, H 为 $n \times 1$ 的实列向量, 从而

$$BX = H \quad (3)$$

若采用最小二乘法(LSM)求解可获得上述线性方程组的最小二乘解 $\hat{X} = B^{-1}H$.

为改善线性系数矩阵 B 的病态性, 适当选择一个阻尼因子 $\alpha > 0$, 通过添加 αX , 可将线性方程组(3)转化为:

$$(B + \alpha I)X = H + \alpha X \quad (4)$$

其中 I 为 n 阶单位矩阵.

收稿日期: 2020-12-18

基金项目: 广西自然科学基金项目(2018GXNSFAA294122); 广西高校中青年教师基础能力提升项目(2017KY0339)资助.

作者简介: 莫春鹏, 硕士, 讲师, 研究方向: 计算数学.

*通信作者: 覃柏英, 硕士, 副教授, 研究方向: 计算流体力学, E-mail: 392644147@qq.com.

该线性方程组可采用如下迭代法数值求解而获得线性方程组(1)和式(3)的解向量 \mathbf{X} :

$$(\mathbf{B} + \alpha\mathbf{I})\hat{\mathbf{X}}^{(k)} = \mathbf{H} + \alpha\hat{\mathbf{X}}^{(k-1)} \quad (5)$$

在实际应用中, 式(5)的迭代法常需转化为如下的迭代算法求解:

$$\hat{\mathbf{X}}^{(k)} = (\mathbf{B} + \alpha\mathbf{I})^{-1}[\mathbf{H} + \alpha\hat{\mathbf{X}}^{(k-1)}] \quad (6)$$

该算法称为阻尼谱修正迭代法(DCCV).

对式(6)也可构造如下改进迭代算法数值求解而获得线性方程组(1)和式(3)的解向量 \mathbf{X} :

$$\begin{cases} \mathbf{R}^{(k)} = \mathbf{H} - \mathbf{B}\hat{\mathbf{X}}^{(k)} \\ \mathbf{D}^{(k)} = (\mathbf{B} + \alpha\mathbf{I})^{-1}\mathbf{R}^{(k)} \\ \hat{\mathbf{X}}^{(k+1)} = \hat{\mathbf{X}}^{(k)} + \mathbf{D}^{(k)} \end{cases} \quad (7)$$

其中: \mathbf{R} 为误差矩阵, \mathbf{D} 为对角矩阵. 该算法称为改进阻尼谱修正迭代法(IDCCV).

2 基于LU分解的阻尼谱修正迭代法

由于数值计算存在的舍入误差, 矩阵的求逆常给病态线性方程组的准确求解带来较大影响. 因此, 为了避免式(6)和式(7)中对矩阵 $\mathbf{B} + \alpha\mathbf{I}$ 的直接求逆, 可对其进行LU分解产生一个下三角形矩阵 \mathbf{L} 和一个上三角形矩阵 \mathbf{U} 以及一个置换矩阵 \mathbf{P} , 使之满足 $\mathbf{LU} = \mathbf{P}(\mathbf{B} + \alpha\mathbf{I})$.

对于DCCV, 由式(5), 设 $\mathbf{Y} = \mathbf{U}\hat{\mathbf{X}}^{(k)}$. 有

$$\begin{aligned} \mathbf{LU}\hat{\mathbf{X}}^{(k)} &= \mathbf{PH} + \alpha\mathbf{P}\hat{\mathbf{X}}^{(k-1)} \\ \mathbf{LY} &= \mathbf{PH} + \alpha\mathbf{P}\hat{\mathbf{X}}^{(k-1)} = \mathbf{W} \end{aligned}$$

其中 \mathbf{W} 为结果矩阵.

由于 \mathbf{L} 为下三角形矩阵, 即 \mathbf{L} 可逆, 有 $\mathbf{Y} = \mathbf{L}^{-1}\mathbf{W}$. 利用中间解 \mathbf{Y} , 由 $\mathbf{U}\hat{\mathbf{X}}^{(k)} = \mathbf{Y}$, 且 \mathbf{U} 为上三角

形矩阵, 即 \mathbf{U} 可逆, 有 $\hat{\mathbf{X}}^{(k)} = \mathbf{U}^{-1}\mathbf{Y}$.

因 $L_{i,i} = 1$, $Y_1 = W_1$ 以及 $\hat{X}_n^{(k)} = Y_n/U_{n,n}$, 因此, 中间解 \mathbf{Y} 和解 $\hat{\mathbf{X}}^{(k)}$ 可根据如下递推式分别求出 $Y_i (i = 2, \dots, n)$ 和 $\hat{X}_i^{(k)} (i = n - 1, \dots, 1)$:

$$Y_i = W_i - \sum_{j=1}^{i-1} L_{i,j} Y_j$$

$$\hat{X}_i^{(k)} = Y_i/U_{i,i} - \sum_{j=i+1}^n U_{i,j} \hat{X}_j^{(k)}/U_{i,i}$$

该求解线性方程组(1)和式(3)的算法, 称为基于LU分解的阻尼谱修正迭代法(LUDCCV).

对于IDCCV, 由式(7), 设 $\mathbf{Z} = \mathbf{UD}^{(k)}$, 有

$$\mathbf{LUD}^{(k)} = \mathbf{PR}^{(k)}, \mathbf{LZ} = \mathbf{PR}^{(k)} = \mathbf{V}$$

由于 \mathbf{L} 为下三角形矩阵, 即 \mathbf{L} 可逆. 从而 $\mathbf{Z} = \mathbf{L}^{-1}\mathbf{V}$, \mathbf{V} 为列向量. 利用中间解 \mathbf{Z} , 由 $\mathbf{UD}^{(k)} = \mathbf{Z}$, 且 \mathbf{U} 为上三角形矩阵, 即 \mathbf{U} 可逆, 从而 $\mathbf{D}^{(k)} = \mathbf{U}^{-1}\mathbf{Z}$.

因 $L_{i,i} = 1$, $Z_1 = V_1$ 以及 $D_n^{(k)} = Z_n/U_{n,n}$, 因此, 中间解 \mathbf{Z} 和解 $\mathbf{D}^{(k)}$ 可根据如下递推式分别求出 $Z_i (i = 2, \dots, n)$ 和 $D_i^{(k)} (i = n - 1, \dots, 1)$:

$$Z_i = V_i - \sum_{j=1}^{i-1} L_{i,j} Z_j$$

$$D_i^{(k)} = Z_i/U_{i,i} - \sum_{j=i+1}^n U_{i,j} D_j^{(k)}/U_{i,i}$$

该求解线性方程组(1)和式(3)的算法, 称为基于LU分解的改进阻尼谱修正迭代法(LUIDCCV).

3 经典算例

算例1 给定线性方程组(1)如下的系数矩阵 $\mathbf{A}_{19 \times 4}$ 和常数项 $\mathbf{b}_{19 \times 1}$:

$$\mathbf{A}_{19 \times 4} = \begin{bmatrix} 2.0000 & -500.0000 & -499.6263 & 1.0000 \\ -2.0000 & 400.0000 & 399.5920 & 3.0000 \\ -2.0000 & 100.0000 & 99.5902 & -4.0000 \\ -1.0000 & 250.0000 & 249.7653 & 1.0000 \\ -1.0000 & 320.0000 & 319.8174 & 5.0000 \\ 1.0000 & 100.0000 & 100.2072 & 3.0000 \\ 3.0000 & 700.0000 & 700.5895 & -3.0000 \\ 5.0000 & -100.0000 & -98.9732 & 1.0000 \\ 4.0000 & 200.0000 & 200.8473 & 2.5000 \\ 4.0000 & 300.0000 & 300.7668 & 4.0000 \\ 1.5000 & 200.0000 & 200.2992 & 2.0000 \\ 3.0000 & 500.0000 & 500.5667 & 6.0000 \\ 4.0000 & 600.0000 & 600.8225 & 5.0000 \\ 5.0000 & 700.0000 & 701.0150 & 7.0000 \\ -5.0000 & 300.0000 & 298.9845 & -3.0000 \\ 9.0000 & 300.0000 & 301.7832 & 1.0000 \\ 3.0000 & -900.0000 & -899.4226 & -9.0000 \\ 2.0000 & 700.0000 & 700.4278 & 2.0000 \\ 7.0000 & 200.0000 & 201.3925 & 6.0000 \end{bmatrix}, \mathbf{b}_{19 \times 1} = \begin{bmatrix} -1551.8021 \\ 1230.5472 \\ 320.1443 \\ 771.6245 \\ 977.5078 \\ 302.1315 \\ 2179.9432 \\ -310.1571 \\ 615.1557 \\ 920.8269 \\ 615.1787 \\ 1534.7067 \\ 1848.1160 \\ 2153.0240 \\ 935.7752 \\ 931.8531 \\ -2763.2762 \\ 2165.4845 \\ 606.8280 \end{bmatrix}$$

此时 $X = [0.2, 1.5, 1.6, -2.8]^T$ 为相应线性方程组(1)的真解. 由于矩阵 A 非对称, 将其转化, 取 $B = A^T A$, 则其条件数为 $1.610 \text{ 1E}9$. 因此, 相应线性方程组(3)为病态线性方程组.

$$A_{18 \times 7} = \begin{bmatrix} 2.0000 & -5.0000 & 1.0000 & 1.0000 & 2.0000 & 5.0293 & 1.2596 \\ 2.0000 & 4.0000 & 1.0000 & 3.0000 & 4.0000 & -3.0126 & 2.3340 \\ -2.0000 & 1.0000 & 1.0000 & -4.0000 & 1.0000 & -2.9760 & 0.6818 \\ -1.0000 & 2.5000 & 4.0000 & 1.0000 & 3.0000 & 2.0334 & 2.7495 \\ -1.0000 & 3.2000 & 4.0000 & 5.0000 & 3.5000 & 1.9961 & 2.9609 \\ 1.0000 & 1.0000 & -3.0000 & 3.0000 & 2.0000 & -0.9555 & 0.1131 \\ 3.0000 & 7.0000 & -3.0000 & -3.0000 & 1.0000 & 3.1155 & -0.3393 \\ 5.0000 & -1.0000 & -2.0000 & 1.0000 & 3.0000 & 8.0262 & 0.8863 \\ 4.0000 & 2.0000 & -2.0000 & 2.5000 & -1.0000 & 5.9994 & -1.1067 \\ 4.0000 & 3.0000 & -2.0000 & 4.0000 & 6.0000 & 6.0457 & 2.3365 \\ 1.5000 & 2.0000 & 4.0000 & 2.0000 & 3.0000 & 7.0028 & 2.6168 \\ 3.0000 & 5.0000 & 7.0000 & 6.0000 & 2.0000 & 12.9446 & 3.0648 \\ 1.0000 & 1.0000 & 2.3000 & 2.0000 & -4.0000 & 4.3233 & -1.2729 \\ 5.0000 & 4.0000 & 2.0000 & 2.0000 & 1.0000 & 11.9862 & 1.0333 \\ 1.2000 & -2.1000 & 4.0000 & -3.0000 & 3.0000 & 6.4638 & 2.7536 \\ -3.0000 & 2.0000 & 4.0000 & -3.0000 & 1.5000 & -1.9068 & 1.9144 \\ 1.0000 & 2.0000 & 2.0000 & 1.0000 & 5.0000 & 3.9739 & 3.0994 \\ 2.0000 & 1.0000 & 2.0000 & 1.0000 & 2.0000 & 6.0052 & 1.6000 \end{bmatrix}, b_{18 \times 1} = \begin{bmatrix} 14.35446 \\ 8.35576 \\ 2.74982 \\ 24.73961 \\ 21.56885 \\ -0.98621 \\ 31.00523 \\ 34.22785 \\ 15.72203 \\ 41.84873 \\ 39.81424 \\ 58.67556 \\ 0.62231 \\ 52.18315 \\ 37.43436 \\ 10.89664 \\ 36.60252 \\ 30.45768 \end{bmatrix}$$

转化而取 $B = A^T A$, 则其条件数为 $2.996 \text{ 7E}5$. 因此, 相应线性方程组(3)为病态线性方程组.

算例3 给定系数矩阵 A 和常数项 b :

$$A = (a_{ij})_{n \times n}, a_{ij} = \begin{cases} 1, & i \neq j, \\ 1 + p^2, & i = j. \end{cases}$$

$$b = [b_1, b_2, \dots, b_n]^T, b_i = \sum_{j=1}^n a_{ij}$$

其中: $i, j = 1, 2, \dots, n$, 此时 $X = [1, 1, \dots, 1]^T$ 为相应线性方程组(1)的真解. 由于 A 对称, 不需转化而直接取 $B = A$. 当 $n = 10$ 和 $p = 5.0 \times 10^{-3}$ 时, 矩阵 A 的条件数为 $4.000 \text{ 0E}7$, 因此, 相应线性方程组(1)为病态线性方程组.

算例4 给定系数矩阵 A 和常数项 b :

$$A = (a_{ij})_{n \times n}, a_{ij} = 1/(i + j - 1)$$

$$b = [b_1, b_2, \dots, b_n]^T, b_i = \sum_{j=1}^n j a_{ij}$$

其中: $i, j = 1, 2, \dots, n$, 此时 $X = [1, 2, \dots, n]^T$ 为相应线性方程组(1)的真解. 由于 A 对称, 不需转化而直接取 $B = A$. 当 $n = 8$ 时, 矩阵 A 的条件数为 $1.525 \text{ 8E}10$, 因此, 相应线性方程组(1)为病态线性方程组.

算例2 给定如下的系数矩阵 $A_{18 \times 7}$ 和常数项 $b_{18 \times 1}$.

此时 $X = [0.2, 2.0, 1.5, -1.6, 4.8, 3.4, -2.1]^T$ 为相应线性方程组(1)的真解. 由于矩阵 A 非对称, 将其

4 算法的性能分析

利用算法 LSM、DCCV、IDCCV、LUDCCV 和 LUIDCCV 求解病态线性方程组(1)的数值解为 \hat{X} , 可计算 $E_b = A\hat{X}$ 和 $E_x = \hat{X} - X$, 从而可定义为绝对误差的残差 $E_b = \|E_b\|_2$ 以评价数值解 \hat{X} 对方程组(1)的满足程度, 以及可定义为均方误差的残差 $E_x = \|E_x\|_2^2/n$ 和相对误差的残差 $E_\infty = \|E_x\|_\infty / \|X\|_\infty$ 以评价数值解 \hat{X} 偏离真实解 X 的程度. 因此, 利用残差 E_b 、 E_x 、 E_∞ 可评价上述5种算法数值求解病态线性方程组的准确性, 以及比较与分析上述5种算法数值求解病态线性方程组的性能.

现采用5种算法 LSM、DCCV、IDCCV、LUDCCV 和 LUIDCCV 分别数值求解上述4个算例, 结果分别如表1—表4所示. 其中 α 的值在4个算例中分别为 0.280 、 0.089 、 4.000×10^{-14} 和 5.000×10^{-12} , 算法的迭代次数用 F 表示.

由表1—表4可知, 5种算法都可实现上述4个算例的病态线性方程组较高精度求解, 且 LUIDCCV、IDCCV、LUDCCV 和 DCCV 都优于 LSM. 同时, LUDCCV 和 LUIDCCV 分别优于 DCCV 和

IDCCV.这说明当线性方程组病态时,采用LU分解避免系数矩阵直接求逆的LUDCCV和LUIDCCV优于系数矩阵直接求逆的DCCV和IDCCV,因此,LU分解可减弱矩阵求逆数值计算过程中舍入误差存在的影响,从而提高算法求解病态线性方程组数值解的精度.LIDCCV和LUIDCCV分别优于DCCV和LUDCCV,这也说明采用新数值迭代方式,也可提高算法求解病态线性方程组数值解的精度.因此,

表1 算例1的求解结果

算法	LSM	DCCV	LUDCCV	IDCCV	LUIDCCV
\hat{x}_1	0.200 000 017 693	0.200 000 027 099	0.200 000 024 558	0.200 000 006 815	0.200 000 000 570
\hat{x}_2	1.500 000 296 632	1.500 000 132 915	1.500 000 122 947	1.500 000 034 119	1.500 000 002 852
\hat{x}_3	1.600 000 146 771	1.599 999 864 763	1.599 999 877 054	1.599 999 965 881	1.599 999 997 148
\hat{x}_4	-2.800 000 000 062	-2.799 999 999 871	-2.799 999 999 881	-2.799 999 999 966	-2.799 999 999 997
E_b	8.741 971E-04	4.577 676E-06	1.220 185E-08	3.386 315E-09	2.832 256E-10
E_x	1.374 857E-07	1.103 553E-07	1.011 313E-07	2.806 531E-08	2.345 898E-09
E_∞	1.059 399E-07	4.829 907E-08	4.390 958E-08	1.218 552E-08	1.018 552E-09
F	1	735	735	735	735

表2 算例2的求解结果

算法	LSM	DCCV	LUDCCV	IDCCV	LUIDCCV
\hat{x}_1	0.199 999 999 964	0.200 000 000 042	0.200 000 000 010	0.200 000 000 016	0.200 000 000 004
\hat{x}_2	2.000 000 000 000	2.000 000 000 000	2.000 000 000 000	2.000 000 000 000	2.000 000 000 000
\hat{x}_3	1.499 999 999 940	1.500 000 000 019	1.500 000 000 005	1.500 000 000 008	1.500 000 000 002
\hat{x}_4	-1.599 999 999 999	-1.600 000 000 000	-1.600 000 000 000	-1.600 000 000 000	-1.600 000 000 000
\hat{x}_5	4.800 000 000 007	4.799 999 999 997	4.800 000 000 000	4.799 999 999 999	4.800 000 000 001
\hat{x}_6	3.400 000 000 043	3.399 999 999 978	3.399 999 999 995	3.399 999 999 992	3.399 999 999 998
\hat{x}_7	-2.100 000 000 013	-2.099 999 999 995	-2.100 000 000 001	-2.099 999 999 998	-2.100 000 000 001
E_b	6.725 144E-10	2.390 012E-11	8.231 028E-13	1.143 962E-12	4.389 336E-13
E_x	3.296 769E-11	1.984 407E-11	5.052 097E-12	7.733 900E-12	2.028 170E-12
E_∞	1.260 029E-11	8.678 093E-12	2.182 895E-12	3.421 158E-12	8.081 614E-13
F	1	644	644	644	644

表3 算例3的求解结果

算法	LSM	DCCV	LUDCCV	IDCCV	LUIDCCV
\hat{x}_1	0.999 999 955 297	0.999 999 959 022	1.000 000 010 751	0.999 999 993 868	0.999 999 999 439
\hat{x}_2	2.000 000 029 802	1.999 999 996 275	1.999 999 976 112	2.000 000 005 591	1.999 999 993 221
\hat{x}_3	2.999 999 988 824	2.999 999 962 747	2.999 999 985 105	3.000 000 012 209	2.999 999 988 003
\hat{x}_4	3.999 999 985 099	4.000 000 007 451	3.999 999 988 386	3.999 999 990 405	3.999 999 991 285
\hat{x}_5	5.000 000 007 451	4.999 999 966 472	5.000 000 018 307	5.000 000 000 747	4.999 999 992 784
\hat{x}_6	5.999 999 981 374	5.999 999 970 198	6.000 000 028 170	5.999 999 982 668	6.000 000 002 647
\hat{x}_7	6.999 999 985 099	6.999 999 970 198	6.999 999 972 656	6.999 999 978 110	6.999 999 999 240
\hat{x}_8	8.000 000 011 176	8.000 000 029 802	7.999 999 998 735	8.000 000 010 805	8.000 000 009 077
\hat{x}_9	9.000 000 037 253	9.000 000 000 000	9.000 000 025 991	9.000 000 007 627	9.000 000 011 465
\hat{x}_{10}	9.999 999 970 198	10.000 000 000 000	9.999 999 995 786	10.000 000 017 970	10.000 000 012 839
E_b	1.531 452E-07	4.358 749E-07	3.552 714E-14	7.105 427E-15	7.105 427E-15
E_x	2.587 222E-08	2.362 942E-08	2.003 963E-08	1.328 287E-08	8.847 241E-09
E_∞	4.470 348E-09	4.097 819E-09	2.817 026E-09	2.188 994E-09	1.283 895E-09
F	1	8	8	8	8

表4 算例4的求解结果

算法	LSM	DCCV	LUDCCV	IDCCV	LUIDCCV
\hat{x}_1	1.000 000 000 058	1.000 000 000 058	1.000 000 000 095	1.000 000 000 036	0.999 999 999 993
\hat{x}_2	1.999 999 996 275	2.000 000 011 176	1.999 999 994 899	1.999 999 998 232	2.000 000 000 342
\hat{x}_3	2.999 999 985 099	2.999 999 821 186	3.000 000 066 654	3.000 000 021 439	2.999 999 995 758
\hat{x}_4	4.000 000 357 628	4.000 000 000 000	3.999 999 639 241	3.999 999 890 843	4.000 000 022 233
\hat{x}_5	4.999 999 761 581	5.000 000 238 419	5.000 000 971 107	5.000 000 279 234	4.999 999 941 326
\hat{x}_6	5.999 999 523 163	5.999 999 046 326	5.999 998 626 323	5.999 999 621 720	6.000 000 081 997
\hat{x}_7	6.999 998 331 070	6.999 998 092 651	7.000 000 977 221	7.000 000 259 283	6.999 999 942 091
\hat{x}_8	7.999 999 880 791	7.999 999 940 395	7.999 999 724 392	7.999 999 929 202	8.000 000 016 265
E_b	6.151 248E-07	8.818 229E-07	1.248 199E-14	6.280 370E-16	2.129 777E-15
E_x	6.125 294E-07	7.196 037E-07	7.555 107E-07	2.089 722E-07	4.519 253E-08
E_∞	2.086 163E-07	2.384 186E-07	1.717 096E-07	4.728 504E-08	1.024 962E-08
F	1	143	143	143	143

迭代算法 LUIDCCV 相比他 4 种迭代算法最优，可明显提高阻尼谱修正迭代法求解病态线性方程组数值解的精度，使所求数值解 \hat{X} 更高程度满足线性方程组(1)和接近线性方程组(1)的真实解 X 。

5 高维问题中的应用

为了更充分说明 LUDCCV 求解病态线性方程

组的性能，现将其应用于高维问题，即将 LUDCCV 应用于算例3和算例4的高维病态线性方程组的求解。

对于算例3的病态线性方程组，分别取其维数 $n = 100, 200, 500, 1000, 2000, 3000, 4000$ ，参数 p 取 5.0×10^{-6} 时，矩阵 A 的条件数 κ 分别如表5所示，阻尼因子 α 取 1.000. 采用 LUIDCCV 分别求解，残差 E_b, E_x, E_∞ 的值分别如表5所示。

表5 高维情况下算例3的求解结果

n	100	200	500	1 000	2 000	3 000	4 000
κ	4.051 1E+12	8.540 0E+12	2.262 1E+13	8.172 4E+13	3.757 1E+15	1.210 2E+14	1.650 7E+14
E_b	2.339 4E-13	4.092 6E-12	1.523 8E-11	2.502 8E-11	1.021 4E-11	3.733 2E-10	2.009 4E-10
E_x	6.557 4E-14	1.799 5E-13	6.703 5E-13	2.368 7E-12	2.490 4E-11	3.518 5E-11	5.410 8E-11
E_∞	1.826 3E-13	4.677 4E-13	1.684 0E-12	5.267 3E-12	4.995 2E-11	8.141 8E-11	1.849 2E-10
F	8	8	8	8	8	8	8

对于算例4的病态线性方程组，分别取其维数 $n = 100, 200, 500, 1000, 2000, 3000, 4000$ ，

矩阵 A 的条件数 κ 分别如表6所示，阻尼因子 α 取 5.000×10^{-12} ，残差 E_b, E_x, E_∞ 的值分别如表6所示。

表6 高维情况下算例4的求解结果

n	100	200	500	1 000	2 000	3 000	4 000
κ	3.339 7E+19	6.075 7E+19	3.223 3E+20	6.147 4E+20	4.876 7E+21	2.771 7E+21	8.089 6E+21
E_b	2.726 8E-12	2.289 9E-11	1.366 9E-09	5.461 5E-09	6.671 3E-09	2.932 1E-09	1.314 3E-07
E_x	4.985 4E-02	1.383 9E-01	5.985 7E-01	1.970 8E+00	9.471 7E+00	1.916 1E+01	3.224 0E+01
E_∞	1.274 6E-03	2.035 3E-03	5.545 0E-03	8.232 1E-03	3.626 1E-02	5.155 0E-02	7.975 6E-02
F	20	20	20	20	20	20	20

由表5可知,对于高维弱病态的线性方程组,LUDCCV可获得其较高精度的数值解,该数值解较高程度满足线性方程组(1),同时也比较接近线性方程组(1)的真实解.但由表6可知,将LUDCCV应用于高维强病态的线性方程组时,所求的数值解其精度有待提高.

为了提高LUIDCCV求解高维强病态线性方程组的精度,对于线性方程组(3)的常数项,若其各元素之间的数值相差较大,可对常数项 H 先归一化,再求解线性方程组(3),从而提高LUDCCV求解高维强病态线性方程组的精度,使其数值解更满足线

性方程组(1)和接近其真实解.设 $H = [h_1, h_2, \dots, h_n]^T$.由此,构造 $C = \text{diag}(1/h_1, 1/h_2, \dots, 1/h_n)$,可将线性方程组(3)转化为:

$$CBX = E_{n \times 1}$$

其中, $E_{n \times 1} = CH = [1, 1, \dots, 1]^T$,即实现常数项的归一化.由此,再采用下式进行数值迭代求解:

$$(CB + \alpha I)\hat{X}^{(k)} = E_{n \times 1} + \alpha \hat{X}^{(k-1)}$$

根据上述方法,对算例4的强病态线性方程组进行归一化处理后,再采用LUIDCCV数值迭代求解强病态线性方程组,结果如表7所示.

表7 高维算例4归一化后的求解结果

n	100	200	500	1 000	2 000	3 000	4 000
κ	3.339 7E+19	6.075 7E+19	3.223 3E+20	6.147 4E+20	4.876 7E+21	2.771 7E+21	8.089 6E+21
E_b	2.101 2E-13	7.069 1E-13	5.603 5E-12	2.291 9E-11	1.362 3E-10	4.207 4E-10	9.512 8E-10
E_x	5.903 3E-04	9.913 8E-04	2.732 8E-03	5.938 5E-03	1.539 1E-02	2.725 5E-02	4.110 6E-02
E_{∞}	1.546 4E-05	1.413 0E-05	2.108 6E-05	2.613 4E-05	3.700 2E-05	4.354 0E-05	5.033 7E-05
F	20	20	20	20	20	20	20

比较表6与表7的结果可知,对常数项 H 先归一化后再采用LUIDCCV求解线性方程组(3),高维强病态线性方程组求解的精度提高明显,从而LUIDCCV求解病态线性方程组时可使其数值解更满足线性方程组(1),接近其真实解 X .

6 结语

为提高谱修正迭代法的计算效率,使阻尼谱修正迭代法收敛至病态线性方程组的真解并提高其收敛速度,实现病态线性方程组高精度和高效率的求解,本文改进了阻尼谱修正迭代法的数值迭代方式,并采用矩阵的LU分解避免直接求解系数矩阵的逆矩阵,由此提出基于矩阵LU分解的新阻尼谱修正迭代法.采用4个经典算例,将基于LU分解的改进阻尼谱修正迭代法应用于高维病态线性方程组,探讨了矩阵的LU分解和新的迭代方式对阻尼谱修正迭代法求解病态线性方程组的性能影响.可得到如下结论:

1)结合矩阵的LU分解或新的迭代方式都可提高阻尼谱修正迭代法求解病态线性方程组的精度,但新的迭代方式对解精度的提高效果更明显,且同时结合矩阵的LU分解和新迭代方式的阻尼谱修正迭代法精度最高;

2)相比于其他4种算法,结合矩阵LU分解和

新迭代方式的阻尼谱修正迭代法求解病态线性方程组的精度可明显提高,且将其应用于高维弱病态线性方程组求解,其数值解的精度也较高;

3)若常数项 H 的各元素间的数值相差较大且非零,采用归一化法对常数项 H 进行处理后再求解,可提高基于矩阵LU分解的新阻尼谱修正迭代法求解病态线性方程组的精度.

参考文献

- [1] 熊新,吴洪涛,于学华,等.工程车辆油气悬架参数化建模与幅频特性分析[J].振动.测试与诊断,2014,34(5):926-931.981.
- [2] 钱进,吴时国,崔若飞,等.新驿矿区奥陶系灰岩孔隙度预测方法[J].桂林理工大学学报,2012,32(1):43-47.
- [3] DENG X SH, YIN L B, PENG S CH, et al. An iterative algorithm for solving ill-conditioned linear least squares problems[J]. Geodesy and Geodynamics, 2015, 6(6): 453-459.
- [4] REICHEL L, RODRIGUEZ G. Old and new parameter choice rules for discrete ill-posed problems[J]. Numerical Algorithms, 2013, 63(1): 65-87.
- [5] BREZINSKI C, NOVATI P, REDIVO-ZAGLIA M. A rational Arnoldi approach for ill-conditioned linear sys-

- tems[J]. Journal of Computational and Applied Mathematics, 2012, 236(8): 2063-2077.
- [6] 覃有堂, 林贤坤, 覃柏英. 精细积分阻尼谱修正迭代法在病态线性方程组中的应用[J]. 广西科技大学学报, 2017, 28(3): 1-8.
- [7] SMOKTUNOWICZ Alicja, SMOKTUNOWICZ Agata. Iterative refinement techniques for solving block linear systems of equations[J]. Applied Numerical Mathematics, 2013, 67: 220-229.
- [8] NEUMAN A, REICHEL L, SADOK H. Implementations of range restricted iterative methods for linear discrete ill-posed problems[J]. Linear Algebra and Its Applications, 2012, 436(10): 3974-3990.
- [9] VILOCHE BAZÁN F S, CUNHA M C C, BORGES L S. Extension of GKB-FP algorithm to large-scale general-form Tikhonov regularization[J]. Numerical Linear Algebra with Applications, 2014, 21(3): 316-339.
- [10] WU X Y. An effective predictor-corrector process for large scale linear system of equations[J]. Applied Mathematics and Computation, 2006, 180(1): 160-166.
- [11] WU X Y. Note on predictor-corrector process for ill-conditioned linear system of equations[J]. Applied Mathematics and Computation, 2006, 183(1): 596-602.
- [12] WU X Y, FANG Y L. Wilkinson's iterative refinement of solution with automatic step-size control for linear system of equations[J]. Applied Mathematics and Computation, 2007, 193(2): 506-513.
- [13] RILEY J D. Solving systems of linear equations with a positive definite, symmetric, but possibly ill-conditioned matrix[J]. Mathematical Tables and other Aids to Computation, 1955, 9(51): 96-101.
- [14] 王新洲, 刘丁酉, 张前勇, 等. 谱修正迭代法及其在测量数据处理中的应用[J]. 黑龙江工程学院学报, 2001(2): 3-6.
- [15] 刘丽华. 解二维 Poisson 方程离散化线性方程组的新型二次 PE_k 方法[J]. 广西科技大学学报, 2016, 27(2): 100-103.
- [16] 邓兴升, 孙虹虹. 自适应谱修正 LU 分解法解算高病态法方程[J]. 大地测量与地球动力学, 2014, 34(6): 135-139.

Iteration method by correcting characteristic value with damping factor based on LU decomposition for ill-conditioned system of linear equations

MO Chunpeng, QIN Boying*

(College of Science, Guangxi University of Science and Technology, Liuzhou 545006, China)

Abstract: Based on the iteration method by correcting characteristic value with damping factor, combined with LU matrix decomposition and a new numerical iteration method, this paper proposes an improved iteration method by correcting characteristic value with damping factor based on LU matrix decomposition, which is applied to solve ill-conditioned system of linear equations. Some classical examples are used to investigate the influence of LU matrix decomposition and the new numerical iteration method on the performance of the algorithm for solving ill-conditioned system of linear equations. The results show that both the LU matrix decomposition and the new numerical iteration method can improve the accuracy of the algorithm for solving ill-conditioned system of linear equations, and the proposed algorithm can also improve the accuracy of solving high-dimensional ill-conditioned system of linear equations.

Key words: LU decomposition; iteration method by correcting characteristic value; ill-conditioned matrix; system of linear equations

(责任编辑: 罗小芬、黎 娅)