doi: 10.3969/j. issn. 1007-2861. 2011. 04. 010

线性系统单输入的 2 种简单的极点配置算法

蒋尔雄

(上海大学 理学院,上海 200444)

摘要: 对线性系统的单输入情况,提出2种简单的极点配置算法. 2种方法都将未知量归结为一个线性代数方程组的解,而这个线性代数方程组系数矩阵的每一行均为系数矩阵是三角形的线性代数方程组的解. 该算法计算简单,计算量少. 第一种方法还同时求出配置后矩阵的特征向量,为系统设计提供参考;第二种方法的计算量更少. 对第一种方法进行误差分析,证明只要计算精度充分高,都能达到对任意给定的大于0的极点配置误差要求.

关键词: 单输入线性系统;极点配置;反馈增益矩阵;矩阵特征值;Hessenberg 矩阵

中图分类号: O 151.2 **文献标志码**: A **文章编号**: 1007-2861(2011)04-0429-09

Two Algorithms for Pole Assignment of Single-Input Linear Systems

JIANG Er-xiong

(College of Sciences, Shanghai University, Shanghai 200444, China)

Abstract: Two algorithms for the pole assignment of single-input linear systems are presented. The algorithm reduces an unknown $\boldsymbol{y}$ to a solution of a system of linear algebraic equations. Each row of the coefficient matrix $\boldsymbol{G}$ is a solution of a linear algebraic equation system with a triangle coefficient matrix. In the first algorithm, $\boldsymbol{G}$ is the eigenvector matrix of $\hat{\boldsymbol{H}}$ which is orthogonal similar to state feedback matrix $(\boldsymbol{A}+\boldsymbol{BF})$. Low cost as compared with many existing algorithms are required.

Key words: single-input linear system; pole assignment; feedback gain matrix; matrix eigenvalue; Hessenberg matrix

目前已有很多关于线性系统极点配置问题的研究[1-7]. 对线性系统的单输入情况,本研究提出了2种简单的极点配置算法,与一些著名的算法比较,具有计算简单,运算量少的特点. 第一种算法的运算量为$5n^3/6+n^2/2+n/6$,第二种算法的运算量为$n^3/6+n^2/2+n/3$,其中 n 为状态向量的维数. 如果问题是要寻找增益向量 $\boldsymbol{F}$,我们把它归结为寻找未知向量 $\boldsymbol{y}$,其中 $\boldsymbol{y}$ 与 $\boldsymbol{F}$ 的关系为

$$\boldsymbol{F}=\frac{(\boldsymbol{y}-\boldsymbol{h})}{\beta}\boldsymbol{Q}^{\mathrm{T}},$$

式中,正交矩阵 $\boldsymbol{Q}$、向量 $\boldsymbol{h}$、正常数 β 都是已知的. 我们列出一个以 $\boldsymbol{y}$ 为解的线性代数方程组. 该线性代数方程组的系数矩阵的每一行均为系数矩阵是三角形的线性代数方程组的解,并容易求得. 第一种算法还求出了配置后矩阵的特征向量,有助于了解问题的条件数.

从20世纪60年代开始,研究者提出了很多极点配置算法,但对算法的数值计算作误差分析的却很少见到. 因此,本研究对第一种算法进行了误差分析,证明 $\boldsymbol{y}$ 计算是向后误差稳定的;同时,证明了只

收稿日期:2011-06-09

基金项目:上海市重点学科建设资助项目(S30104)

通信作者:蒋尔雄(1934~),男,教授,博士生导师,研究方向为数值线性代数. E-mail:ejiang@fudan.edu.cn

要计算精度充分高,都能达到对任意给定的大于0的极点配置误差要求.

考虑如下不变线性系统:

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t),\\ \boldsymbol{x}(0) &= \boldsymbol{x}_0,\end{aligned} \tag{1}$$

式中,$\boldsymbol{x}\in\mathbf{R}^n$,$\boldsymbol{u}\in\mathbf{R}^p$ 分别为状态向量和输入向量,常数矩阵 $\boldsymbol{A}\in\mathbf{R}^{n\times n}$,$\boldsymbol{B}\in\mathbf{R}^{n\times p}$ 是给定的. 假定系统是可控的,即矩阵$[\boldsymbol{A}-\lambda\boldsymbol{I},\boldsymbol{B}]$的秩为 n,对任意 $\lambda\in\mathbf{C}$ 成立. 令 $\boldsymbol{u}(t)=\boldsymbol{F}\boldsymbol{x}(t)+\boldsymbol{v}(t)$,系统(1)转化为

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= (\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F})\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{v}(t),\\ \boldsymbol{x}(0) &= \boldsymbol{x}_0.\end{aligned}$$

极点配置问题就是寻找一个矩阵 $\boldsymbol{F}$,使得 $n\times n$ 矩阵 $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F}$ 的 n 个特征值恰为 n 个给定的极点 $\lambda_1,\lambda_2,\cdots,\lambda_n$. 本研究只考虑 $p=1$,即系统(1)是单输入的特殊情况. 此时,$\boldsymbol{B}$ 为一个 n 维列向量,而$\boldsymbol{F}=(f_1,f_2,\cdots,f_n)$为一个 n 维行向量. 对于矩阵对$(\boldsymbol{A},\boldsymbol{B})$,可以找到一个实正交矩阵 $\boldsymbol{Q}$,使得

$$\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{H} = (h_{i,j}),\quad \boldsymbol{Q}\boldsymbol{B} = \beta\boldsymbol{e}_1, \tag{2}$$

式中,$\boldsymbol{H}$ 为上 Hessenberg 矩阵,$\beta=\|\boldsymbol{B}\|$(本工作采用的范数均为2-norm),$\boldsymbol{e}_1$ 为恒等矩阵的第一列. 我们知道,系统(1)可控的充要条件[2,4-6]为

$$h_{2,1}h_{3,2}\cdots h_{n,n-1}\neq 0.$$

同一个正交矩阵 $\boldsymbol{Q}$ 作用在矩阵 $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F}$ 上,有

$$\hat{\boldsymbol{H}} = \boldsymbol{Q}(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F})\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{H}+\beta\boldsymbol{e}_1\boldsymbol{g}^{\mathrm{T}},$$

式中,

$$\boldsymbol{g}^{\mathrm{T}} = \boldsymbol{F}\boldsymbol{Q}^{\mathrm{T}}.$$

值得注意的是,$\hat{\boldsymbol{H}}$ 为上 Hessenberg 矩阵,相似于 $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F}$,其特征值即为给定的极点. 矩阵 $\hat{\boldsymbol{H}}$ 的 n 行(除第一行外)都跟 $\boldsymbol{H}$ 相同,因此,确定了其第一行,就能确定 $\boldsymbol{g}^{\mathrm{T}}$,从而可以确定

$$\boldsymbol{F} = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{Q}.$$

因此,我们将 $\hat{\boldsymbol{H}}$ 的第一行作为未知量,记 $\hat{\boldsymbol{H}}$ 的第一行为$(y_1,y_2,\cdots,y_n)$,$\boldsymbol{g}^{\mathrm{T}}=(g_1,g_2,\cdots,g_n)$. 令

$$\begin{aligned}\boldsymbol{y} &= (y_1,y_2,\cdots,y_n)^{\mathrm{T}} =\\ &(h_{1,1}+\beta g_1,h_{1,2}+\beta g_2,\cdots,h_{1,n}+\beta g_n)^{\mathrm{T}}.\end{aligned}$$

基于此,寻找 $\boldsymbol{F}$ 归结为寻找 $\boldsymbol{y}$.

1 2种算法

考虑方程

$$(\hat{\boldsymbol{H}}-\lambda\boldsymbol{I})\boldsymbol{x}(\lambda) = \alpha(\lambda)\boldsymbol{e}_1, \tag{3}$$

式中,λ 为参数,$\boldsymbol{x}(\lambda)=(x_1(\lambda),x_2(\lambda),\cdots,x_n(\lambda))^{\mathrm{T}}$为未知向量,$\alpha(\lambda)$为 λ 的函数. 已知

$$\det(\hat{\boldsymbol{H}}-\lambda\boldsymbol{I}) = \prod_{k=1}^{n}(\lambda_k-\lambda),$$

如果 $\lambda\notin\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$,由 Cramer 法则,可得

$$\begin{aligned}x_n(\lambda) &= \det(\boldsymbol{S}_n)/\det(\hat{\boldsymbol{H}}-\lambda\boldsymbol{I}) =\\ &(-1)^{n-1}\alpha(\lambda)\times h_{2,1}h_{3,2}\cdots h_{n,n-1}/\det(\hat{\boldsymbol{H}}-\lambda\boldsymbol{I}),\end{aligned}$$

式中,矩阵 $\boldsymbol{S}_n$ 为将 $\hat{\boldsymbol{H}}$ 的最后一列换成 $\alpha(\lambda)\boldsymbol{e}_1$ 的结果. 若取

$$\begin{aligned}\alpha(\lambda) &= (-1)^{n-1}\det(\hat{\boldsymbol{H}}-\lambda\boldsymbol{I})/h_{2,1}h_{3,2}\cdots h_{n,n-1} =\\ &-\prod_{k=1}^{n}(\lambda-\lambda_k)/h_{2,1}h_{3,2}\cdots h_{n,n-1},\end{aligned}$$

那么,有 $x_n(\lambda)\equiv 1$. 注意,由于 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 给定,因此,上述 $\alpha(\lambda)$ 也是已知的. 因为 $\hat{\boldsymbol{H}}$ 是上 Hessenberg 矩阵,由 $x_n(\lambda)\equiv 1$,可以从方程(3)自下而上地逐步计算出 $x_{n-1}(\lambda),x_{n-2}(\lambda),\cdots,x_1(\lambda)$. 如果记

$$\begin{aligned}\check{\boldsymbol{x}}(\lambda) &= (x_1(\lambda),x_2(\lambda),\cdots,x_{n-1}(\lambda))^{\mathrm{T}},\\ \boldsymbol{b}(\lambda) &= (h_{2,n},h_{3,n},\cdots,h_{n,n}-\lambda)^{\mathrm{T}},\end{aligned}$$

那么,$\check{\boldsymbol{x}}(\lambda)$满足方程

$$\boldsymbol{U}(\lambda)\check{\boldsymbol{x}}(\lambda) = -\boldsymbol{b}(\lambda), \tag{4}$$

式中,

$$\boldsymbol{U}(\lambda) = \begin{pmatrix} h_{2,1} & h_{2,2}-\lambda & \cdots & h_{2,n-1}\\ 0 & & & \vdots\\ \vdots & & & h_{n-1,n-1}-\lambda\\ 0 & \cdots & 0 & h_{n,n-1}\end{pmatrix}$$

为三角形矩阵.

易知,当 $k\in\{1,2,\cdots,n-1\}$时,$x_{n-k}(\lambda)$为 λ 的 k 次多项式,即

$$x_{n-k}(\lambda) = \frac{\lambda^k}{h_{n,n-1}h_{n-1,n-2}\cdots h_{n-k+1,n-k}}+\cdots.$$

尽管上述结果是在 $\lambda\notin\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$时得到的,但易知当 $\lambda\in\{\lambda_1,\lambda_2,\cdots,\lambda_n\}$时,上式也成立. 此时,$\alpha(\lambda)=0$,而 $\boldsymbol{x}(\lambda)$为 $\hat{\boldsymbol{H}}$ 的一个特征向量,其第 n 分量$x_n(\lambda)=1$.

取 $\lambda=\lambda_j$,$x_n(\lambda_j)=1$,由方程(4)求出$x(\lambda_j)$. $\hat{\boldsymbol{H}}$ 的第一行是未知向量 $\boldsymbol{y}^{\mathrm{T}}$,由式(3)中的第一个方程,可得

$$\boldsymbol{x}(\lambda_j)^{\mathrm{T}}\boldsymbol{y} = \lambda_j x_1(\lambda_j).$$

如果 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 都不相同,我们就可得到 $\boldsymbol{y}^{\mathrm{T}}$ 的 n 个方程,有

$$\boldsymbol{G}\boldsymbol{y} = \boldsymbol{c}, \tag{5}$$

式中,

$$\boldsymbol{G} = (g_{i,j}) = (x_j(\lambda_i)),$$
$$\boldsymbol{c} = (\lambda_1 x_1(\lambda_1), \lambda_2 x_1(\lambda_2), \cdots, \lambda_n x_1(\lambda_n))^{\mathrm{T}}.$$

由于 $\boldsymbol{x}(\lambda_1), \boldsymbol{x}(\lambda_2), \cdots, \boldsymbol{x}(\lambda_n)$ 为特征向量,对应不同的特征值 $\lambda_1, \lambda_2, \cdots, \lambda_n$,因此,它们是线性独立的,从而 $\boldsymbol{G}$ 是非奇异的,方程(5)有唯一解 $\boldsymbol{y}$. 这样,我们给出如下的第一种算法.

算法 Ⅰ

步骤 1　按式(2),求出 $\boldsymbol{Q}$ 和 $\boldsymbol{H}$;

步骤 2　按式(4),对 $j=1,2,\cdots,n$,求出 $\boldsymbol{x}(\lambda_1)$, $\boldsymbol{x}(\lambda_2), \cdots, \boldsymbol{x}(\lambda_n)$;

步骤 3　按方程(5),求出 $\boldsymbol{y}$;

步骤 4　计算 $\boldsymbol{g} = (y_1 - h_{1,1}, y_2 - h_{1,2}, \cdots, y_n - h_{1,n})^{\mathrm{T}}/\beta$, $\beta = \|\boldsymbol{B}\|$;

步骤 5　计算 $\boldsymbol{F} = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{Q}$.

如果 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 中有相同部分,那么式(5)中方程的个数就会减少,即若 $\lambda_1 = \lambda_2$,则由式(4)只能求出一特征向量 $\boldsymbol{x}(\lambda_1) = \boldsymbol{x}(\lambda_2)$,这样方程的个数就减少了. 减少的方程能否得到补充,如何补充的问题将在介绍第二种算法之后再来论述. $\boldsymbol{x}(\lambda)$ 和 $\alpha(\lambda)$ 都是 λ 的多项式,是充分光滑的,因此,对方程(1)两边进行微商,可得

$$(\hat{\boldsymbol{H}} - \lambda \boldsymbol{I})\boldsymbol{x}'(\lambda) = \alpha'(\lambda)\boldsymbol{e}_1 + \boldsymbol{x}(\lambda). \tag{6}$$

一般地,进行 k 次微商,可得

$$(\hat{\boldsymbol{H}} - \lambda \boldsymbol{I})\boldsymbol{x}^{(k)}(\lambda) = \alpha^{(k)}(\lambda)\boldsymbol{e}_1 + k\boldsymbol{x}^{(k-1)}(\lambda). \tag{7}$$

选取一个 λ_j,令 $\lambda = \lambda_j$,从式(4)可以求得 $\boldsymbol{x}(\lambda_j)$,因而,有 $\boldsymbol{y}$ 的第一个方程为

$$\boldsymbol{x}(\lambda_j)^{\mathrm{T}}\boldsymbol{y} = \lambda_j x_1(\lambda_j). \tag{8}$$

利用 $x_n'(\lambda_j) = 0$, $x_{n-1}'(\lambda_j) = 1/h_{n,n-1} \neq 0$ 和 $\alpha'(\lambda_j) = -\prod_{k=1,k\neq j}^{n}(\lambda_j - \lambda_k)/h_{2,1}h_{3,2}\cdots h_{n,n-1}$,从式(6)中后面的 $n-1$ 个方程,可以求出 $\boldsymbol{x}'(\lambda_j)$,再从式(6)的第一个方程求得 $\boldsymbol{y}$ 的第二个方程为

$$[\boldsymbol{x}'(\lambda_j)]^{\mathrm{T}}\boldsymbol{y} = \lambda_j x_1'(\lambda_j) + x_1(\lambda_j) + \alpha'(\lambda_j). \tag{9}$$

依此类推,对每个 k 有

$$x_i^{(k)}(\lambda_j) = 0 (i = n, n-1, \cdots, n-k+1),$$
$$x_{n-k}^{(k)}(\lambda_j) = k!/h_{n,n-1}h_{n-1,n-2}\cdots h_{n-k+1,n-k} \neq 0.$$

由此,可以求出 $\boldsymbol{x}^{(k)}(\lambda_j)$. 再从式(6)中的第一个方程,求得 $\boldsymbol{y}$ 的第 $k+1$ 个方程为

$$[\boldsymbol{x}^{(k)}(\lambda_j)]^{\mathrm{T}}\boldsymbol{y} = \lambda_j x_1^{(k)}(\lambda_j) + k x_1^{(k-1)}(\lambda_j) + \alpha^{(k)}(\lambda_j). \tag{10}$$

由 $k = 2,3,\cdots,n-1$,得到 $\boldsymbol{y}$ 的线性代数方程组为

$$\boldsymbol{G}\boldsymbol{y} = \boldsymbol{c}, \tag{11}$$

式中,

$$\boldsymbol{G} = (g_{i,j}) = [\boldsymbol{x}^{(n-1)}(\lambda_j), \cdots, \boldsymbol{x}'(\lambda_j), \boldsymbol{x}(\lambda_j)]^{\mathrm{T}},$$
$$\boldsymbol{c} = [\lambda_j x_1^{(n-1)}(\lambda_j) + (n-1)x_1^{(n-2)}(\lambda_j) + \alpha^{(n-1)}(\lambda_j), \cdots, \lambda_j x_1(\lambda_j)]. \tag{12}$$

由于 $x_i^{(k)}(\lambda_j) = 0$, $i = n, n-1, \cdots, n-k+1$,因此,矩阵 $\boldsymbol{G}$ 为下三角阵. 又因

$$\det(\boldsymbol{G}) = x_n(\lambda_j)x_{n-1}'(\lambda_j)\cdots x_1^{(n-1)}(\lambda_j) \neq 0,$$

知矩阵 $\boldsymbol{G}$ 非奇异,方程(11)唯一确定 $\boldsymbol{y}$. 这样,我们就给出了如下的第二种算法.

算法 Ⅱ

步骤 1　按式(2),求 $\boldsymbol{Q}$ 和 $\boldsymbol{H}$;

步骤 2　计算 $\alpha'(\lambda_j), \alpha''(\lambda_j), \cdots, \alpha^{(n-1)}(\lambda_j)$;

步骤 3　由方程

$$(\hat{\boldsymbol{H}} - \lambda_j \boldsymbol{I})\boldsymbol{x}^{(k)}(\lambda_j) = \alpha^{(k)}(\lambda_j)\boldsymbol{e}_1 + k\boldsymbol{x}^{(k-1)}(\lambda_j),$$

计算 $\boldsymbol{x}^{(k)}(\lambda_j)(k = 0,1,\cdots,n-1)$,组成矩阵 $\boldsymbol{G}$;

步骤 4　由式(12),计算向量 $\boldsymbol{c}$;

步骤 5　由方程组 $\boldsymbol{G}\boldsymbol{y} = \boldsymbol{c}$,求解 $\boldsymbol{y}$;

步骤 6　计算 $\boldsymbol{g} = (y_1 - h_{1,1}, y_2 - h_{1,2}, \cdots, y_n - h_{1,n})^{\mathrm{T}}/\beta$,其中 $\beta = \|\boldsymbol{B}\|$;

步骤 7　计算 $\boldsymbol{F} = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{Q}$.

算法 Ⅱ 与 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 是否相同无关. 当算法 Ⅰ 有重特征值时,可采用如下的处理办法. 若 λ_j 为 m 重特征值,即 $\lambda_1 = \lambda_2 = \cdots = \lambda_m$,于是 $\alpha(\lambda_j) = \cdots = \alpha^{(m-1)}(\lambda_j) = 0$. 对 $k = 0,1,\cdots,m-1$,方程(7)转化为

$$(\hat{\boldsymbol{H}} - \lambda_j \boldsymbol{I})\boldsymbol{x}^{(k)}(\lambda_j) = k\boldsymbol{x}^{(k-1)}(\lambda_j).$$

这说明向量 $\boldsymbol{x}'(\lambda_j), \boldsymbol{x}''(\lambda_j), \cdots, \boldsymbol{x}^{(m-1)}(\lambda_j)$ 为矩阵 $\hat{\boldsymbol{H}}$ 对应特征值 λ_j 的根向量,同时,我们得到 $\boldsymbol{y}$ 的 m 个方程

$$[\boldsymbol{x}^{(k)}(\lambda_j)]^{\mathrm{T}}\boldsymbol{y} = \lambda_j x_1^{(k)}(\lambda_j) + k x_1^{(k-1)}(\lambda_j),$$
$$k = 0,1,\cdots,m-1.$$

2　数值算例

例 1　考虑如何确定上 Hessenberg 矩阵的第一行,

$$\boldsymbol{H} = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \end{pmatrix},$$

给定矩阵 $\boldsymbol{H}$ 的特征值为 $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3, \lambda_4 = 4$.

按算法 Ⅰ 和算法 Ⅱ,求解例 1.

(1) 算法Ⅰ.

按定义,有

$$U_c=\begin{pmatrix}1&2&3\\0&1&2\\0&0&1\end{pmatrix},\quad I_c=\begin{pmatrix}0&1&0\\0&0&1\\0&0&0\end{pmatrix}.$$

我们有

$$U(1)=U_c-I_c=\begin{pmatrix}1&1&3\\0&1&1\\0&0&1\end{pmatrix},$$

$$U(2)=U_c-2I_c=\begin{pmatrix}1&0&3\\0&1&0\\0&0&1\end{pmatrix},$$

$$U(3)=U_c-3I_c=\begin{pmatrix}1&-1&3\\0&1&-1\\0&0&1\end{pmatrix},$$

$$U(4)=U_c-4I_c=\begin{pmatrix}1&-2&3\\0&1&-2\\0&0&1\end{pmatrix}.$$

$$b_c=(4,3,2)^{\mathrm T},\quad e_3=(0,0,1)^{\mathrm T}.$$

$$b(1)=b_c-e_3=(4,3,1)^{\mathrm T},$$

$$b(2)=b_c-2e_3=(4,3,0)^{\mathrm T},$$

$$b(3)=b_c-3e_3=(4,3,-1)^{\mathrm T},$$

$$b(4)=b_c-4e_3=(4,3,-2)^{\mathrm T}.$$

由方程

$$U(j)\check{x}(j)=-b(j),\quad j=1,2,3,4,$$

式中,$\check{x}(j)=(x_1(j),x_2(j),x_3(j))^{\mathrm T}$,得到

$x(1)=(1,-2,-1,1)^{\mathrm T}$, $x(2)=(-4,-3,0,1)^{\mathrm T}$,
$x(3)=(-9,-2,1,1)^{\mathrm T}$, $x(4)=(-8,1,2,1)^{\mathrm T}$.

所以,

$$G=(x(1),x(2),x(3),x(4))^{\mathrm T}=\begin{pmatrix}1&-2&-1&1\\-4&-3&0&1\\-9&-2&1&1\\-8&1&2&1\end{pmatrix}.$$

对向量 c,有

$$c=(x_1(1),2x_1(2),3x_1(3),4x_1(4))^{\mathrm T}=(1,-8,-27,-32)^{\mathrm T}.$$

最后,求得方程

$$Gy=c.$$

通过求解方程,可得

$$y=(y_1,y_2,y_3,y_4)^{\mathrm T}=(4,-5,6,-7)^{\mathrm T}.$$

(2) 算法Ⅱ.

步骤1　由多项式 $\alpha(\lambda)=-(\lambda^4-10\lambda^3+35\lambda^2-50\lambda+24)$,取 $\lambda_j=\lambda_1=1$,计算得 $\alpha(1)=0,\alpha'(1)=6,\alpha''(1)=-22,\alpha'''(1)=36$.

步骤2　通过解方程组

$$\begin{cases}(H-I)x=0,\\(H-I)x'=\alpha'(1)e_1+x,\\(H-I)x''=\alpha''(1)e_1+2x',\\(H-I)x'''=\alpha'''(1)e_1+3x'',\end{cases}$$

得到向量

$$x=x(1)=(1,-2,-1,1),$$

$$x'=x'(1)=(-3,-2,1,0),$$

$$x''=x''(1)=(-6,2,0,0),$$

$$x'''=x'''(1)=(6,0,0,0).$$

步骤3　计算 $c=(24,-34,4,1)^{\mathrm T}$.

步骤4　解方程组

$$\begin{cases}6y_1=24,\\-6y_1+2y_2=-34,\\-3y_1-2y_2+y_3=4,\\y_1-2y_2-y_3+y_4=1.\end{cases}$$

该线性方程组的系数矩阵为下三角阵,求解得

$$y=(y_1,y_2,y_3,y_4)^{\mathrm T}=(4,-5,6,-7)^{\mathrm T}.$$

最后,求得上 Hessenberg 矩阵

$$H=\begin{pmatrix}4&-5&6&-7\\1&2&3&4\\0&1&2&3\\0&0&1&2\end{pmatrix}$$

的特征值为 $\lambda_1=1,\lambda_2=2,\lambda_3=3,\lambda_4=4$.

例2　考虑单输入线性系统 $\dot{x}=Ax+Bu$,其中

$$A=\begin{pmatrix}6&-15&18&-10\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{pmatrix},\quad B=e_1=\begin{pmatrix}1\\0\\0\\0\end{pmatrix}.$$

A 的特征值为 $\mu_1=1+i,\mu_2=1-i,\mu_3=2+i,\mu_4=2-i$,它们均在复平面的右半平面,因而,该单输入线性系统是不稳定的.

令 $u=Fx+v$,确定 $F=(f_1,f_2,f_3,f_4)$,使得反馈矩阵

$$\hat{A}=A+BF=\begin{pmatrix}6+f_1&-15+f_2&18+f_3&-10+f_4\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{pmatrix}=:(\hat{a}_{ij}),$$

该矩阵的特征值 $\lambda_1=-1,\lambda_2=-2,\lambda_3=-3,\lambda_4=-4$. 我们用算法Ⅱ求解该问题. 注意到 $\boldsymbol{A},\boldsymbol{B}$ 已具有式(2)所要求的形式,有

$$\alpha(\lambda)=-\det(\lambda\boldsymbol{I}-\hat{\boldsymbol{A}})/(\hat{a}_{2,1}\hat{a}_{3,2}\hat{a}_{4,3})=-\lambda^4-10\lambda^3-35\lambda^2-50\lambda-24.$$

取 $\lambda=\lambda_1=-1$,计算得

$$\alpha(-1)=0,\alpha'(-1)=-6,$$
$$\alpha''(-1)=-22,\alpha'''(-1)=-36.$$

通过求解方程组

$$\begin{cases}(\boldsymbol{H}+\boldsymbol{I})\boldsymbol{x}=0,\\(\boldsymbol{H}+\boldsymbol{I})\boldsymbol{x}'=\alpha'(-1)\boldsymbol{e}_1+\boldsymbol{x},\\(\boldsymbol{H}+\boldsymbol{I})\boldsymbol{x}''=\alpha''(-1)\boldsymbol{e}_1+2\boldsymbol{x}',\\(\boldsymbol{H}+\boldsymbol{I})\boldsymbol{x}'''=\alpha'''(-1)\boldsymbol{e}_1+3\boldsymbol{x}'',\end{cases}$$

得到 $\boldsymbol{x},\boldsymbol{x}',\boldsymbol{x}'',\boldsymbol{x}'''$ 分别为

$$\boldsymbol{x}=\boldsymbol{x}(-1)=(-1,1,-1,1),$$
$$\boldsymbol{x}'=\boldsymbol{x}'(-1)=(3,-2,1,0),$$
$$\boldsymbol{x}''=\boldsymbol{x}''(-1)=(-6,2,0,0),$$
$$\boldsymbol{x}'''=\boldsymbol{x}'''(-1)=(6,0,0,0).$$

计算可得 $\boldsymbol{c}=(-60,-10,-10,1)^{\mathrm{T}}$,因此,可得 $\boldsymbol{y}$ 的方程组为

$$\begin{cases}6y_1=-60,\\-6y_1+2y_2=-10,\\3y_1-2y_2+y_3=-10,\\-y_1+y_2-y_3+y_4=1.\end{cases}$$

解得

$$\boldsymbol{y}=(y_1,y_2,y_3,y_4)^{\mathrm{T}}=(-10,-35,-50,-24)^{\mathrm{T}}.$$

最后,可得

$$\boldsymbol{F}=(f_1,f_2,f_3,f_4)=(-16,-20,-68,-14).$$

上述 2 个算例都是整数运算,没有舍入误差,并且结果是完全正确的,这说明该方法是很稳定的.

例 3　考虑线性系统 $\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}$,其中

$$\boldsymbol{A}=\begin{pmatrix}2.088\,235\,294\,117\,65 & -2.303\,921\,568\,627\,45 & 8.774\,509\,803\,921\,57 & -8.676\,470\,588\,235\,29\\-2.676\,470\,588\,235\,28 & 2.107\,843\,137\,254\,91 & -12.049\,019\,607\,843\,14 & 10.852\,941\,176\,470\,58\\1.323\,529\,411\,764\,71 & 1.107\,843\,137\,254\,90 & 2.950\,980\,392\,156\,86 & -1.147\,058\,823\,529\,41\\1.323\,529\,411\,764\,70 & 0.441\,176\,470\,588\,23 & 3.617\,647\,058\,823\,53 & -1.147\,058\,823\,529\,41\end{pmatrix},$$

$\boldsymbol{B}=(1,0,0,0)^{\mathrm{T}}$. 因为 $\boldsymbol{A}$ 的特征值为 $\mu_1=1+i,\mu_2=1-i,\mu_3=2+i,\mu_4=2-i$,因此,系统是不稳定的. 令输入 $\boldsymbol{u}=\boldsymbol{F}\boldsymbol{x}+\boldsymbol{v}$,我们要找向量 $\boldsymbol{F}=(f_1,f_2,f_3,f_4)$,使得反馈矩阵

$$\hat{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{B}\boldsymbol{F}$$

有特征值 $\lambda_1=-1,\lambda_2=-2,\lambda_3=-3,\lambda_4=-4$.

我们再次采用算法Ⅱ来求解该问题.

步骤 1　首先按式(2)将 $\boldsymbol{B},\boldsymbol{A}$ 正交变换为 $\beta\boldsymbol{e}_1,\boldsymbol{H}$,其中 $\boldsymbol{H}$ 为上 Hessenberg 矩阵. 因为 $\boldsymbol{B}=\boldsymbol{e}_1$,因而 $\beta=1$. 这样的正交矩阵为 $\boldsymbol{Q}=\boldsymbol{Q}_2\boldsymbol{Q}_1$,其中

$$\boldsymbol{Q}_1=\begin{pmatrix}1 & 0 & 0 & 0\\0 & -0.819\,487\,301\,197\,2 & 0.405\,240\,987\,421\,84 & 0.405\,240\,987\,418\,4\\0 & 0.405\,240\,987\,421\,84 & 0.909\,743\,665\,059\,86 & -0.090\,256\,334\,940\,14\\0 & 0.405\,240\,987\,421\,84 & -0.090\,256\,334\,940\,14 & 0.909\,743\,665\,059\,86\end{pmatrix},$$

$$\boldsymbol{Q}_2=\begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & -0.965\,975\,728\,911\,77 & -0.258\,632\,734\,110\,31\\0 & 0 & -0.258\,632\,734\,110\,31 & 0.965\,975\,728\,911\,77\end{pmatrix}.$$

而

$$\boldsymbol{H}=\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^{\mathrm{T}}=\begin{pmatrix}2.088\,235\,294\,117\,65 & 1.927\,764\,043\,650\,52 & -5.077\,760\,376\,702\,64 & -11.317\,300\,654\,785\,64\\3.266\,030\,467\,907\,64 & 2.000\,294\,968\,364\,45 & -8.723\,191\,007\,309\,83 & -13.219\,838\,593\,270\,96\\0 & 1.154\,299\,330\,068\,13 & -0.848\,250\,901\,492\,46 & -5.351\,939\,309\,898\,75\\0 & 0 & 0.294\,418\,448\,178\,90 & 2.759\,720\,639\,010\,37\end{pmatrix},$$

$$\boldsymbol{e}_1=\boldsymbol{Q}\boldsymbol{B}=\boldsymbol{B}.$$

令 $\hat{\boldsymbol{H}}$ 的第一行为 $\boldsymbol{y}^{\mathrm{T}}=(y_1,y_2,y_3,y_4)$,于是

$$\hat{\boldsymbol{H}} = \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \\ \end{array}$$
$$\begin{pmatrix} 3.266\,030\,467\,907\,64 & 2.000\,294\,968\,364\,45 & -8.723\,191\,007\,309\,83 & -13.219\,838\,593\,270\,96 \\ 0 & 1.154\,299\,330\,068\,13 & -0.848\,250\,901\,492\,46 & -5.351\,939\,309\,898\,75 \\ 0 & 0 & 0.294\,418\,448\,178\,90 & 2.759\,720\,639\,010\,37 \end{pmatrix}.$$

步骤2 计算

$$\alpha(\lambda) = (-\lambda^4 - 10\lambda^3 - 35\lambda^2 - 50\lambda - 24)/\delta,$$

式中,

$$\delta = \hat{H}_{2,1}\hat{H}_{3,2}\hat{H}_{4,3} = 1.109\,950\,713\,558\,37.$$

取 $\lambda = \lambda_1 = -1$,计算得

$\alpha(-1) = 0$, $\alpha'(-1) = -5.405\,645\,427\,953\,01$, $\alpha''(-1) = -19.820\,699\,902\,494\,38$,

$\alpha'''(-1) = -32.433\,872\,567\,718\,07$.

步骤3 计算 $\boldsymbol{x},\boldsymbol{x}',\boldsymbol{x}'',\boldsymbol{x}'''$.

由

$$(\hat{\boldsymbol{H}} - \mu_1\boldsymbol{I})\boldsymbol{x} = 0,$$

可得

$$\boldsymbol{x} = (x_1,x_2,x_3,x_4) = (-35.860\,981\,107\,008\,58, 6.315\,323\,582\,396\,22, -12.769\,989\,999\,831\,24, 1).$$

由

$$(\hat{\boldsymbol{H}} - \mu_1\boldsymbol{I})\boldsymbol{x}' = -5.405\,645\,427\,953\,01\boldsymbol{e}_1 + \boldsymbol{x},$$

可得

$$\boldsymbol{x}' = (x_1',x_2',x_3',x_4') = (21.578\,417\,941\,975\,86, -11.509\,501\,440\,903\,68, 3.396\,526\,291\,696\,12, 0).$$

由

$$(\hat{\boldsymbol{H}} - \mu_1\boldsymbol{I})\boldsymbol{x}'' = -19.820\,699\,902\,494\,83\boldsymbol{e}_1 + 2\boldsymbol{x}',$$

可得

$$\boldsymbol{x}'' = (x_1'',x_2'',x_3'',x_4'') = (-12.454\,183\,093\,813\,32, 5.885\,000\,888\,800\,05, 0, 0).$$

由

$$(\hat{\boldsymbol{H}} - \mu_1\boldsymbol{I})\boldsymbol{x}''' = -32.433\,872\,567\,718\,07\boldsymbol{e}_1 + 3\boldsymbol{x}'',$$

可得

$$\boldsymbol{x}''' = (x_1''',x_2''',x_3''',x_4''') = (5.405\,645\,427\,953\,01, 0, 0, 0).$$

合并上述结果为

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{x}(-1) = (-35.860\,981\,107\,008\,58, 6.315\,323\,582\,396\,22, -12.769\,989\,999\,831\,24, 1), \\ \boldsymbol{x}' &= \boldsymbol{x}'(-1) = (21.578\,417\,941\,975\,86, -11.509\,501\,440\,903\,68, 3.396\,526\,291\,696\,12, 0), \\ \boldsymbol{x}'' &= \boldsymbol{x}''(-1) = (-12.454\,183\,093\,813\,32, 5.885\,000\,888\,800\,05, 0, 0), \\ \boldsymbol{x}''' &= \boldsymbol{x}'''(-1) = (5.405\,645\,427\,953\,01, 0, 0, 0). \end{aligned}$$

步骤4 计算

$$\boldsymbol{c} = (-75.202\,067\,277\,111\,04, 35.790\,319\,075\,270\,67, -62.845\,044\,476\,937\,45, 35.860\,981\,107\,008\,58).$$

步骤5 解下列方程组:

$$\begin{cases} 5.405\,645\,427\,953\,01y_1 = -75.202\,067\,277\,111\,04, \\ -12.454\,183\,093\,813\,32y_1 + 5.885\,000\,888\,800\,05y_2 = 35.790\,319\,075\,270\,67, \\ 21.578\,417\,941\,975\,86y_1 - 11.509\,501\,440\,903\,68y_2 + 3.396\,526\,291\,696\,12y_3 = -62.845\,044\,476\,937\,45, \\ -35.860\,981\,107\,008\,58y_1 + 6.315\,323\,582\,396\,22y_2 - 12.769\,989\,999\,831\,24y_3 + y_4 = 35.860\,981\,107\,008\,58. \end{cases}$$

得到

$$\boldsymbol{y} = (y_1, y_2, y_3, y_4) = (-13.911\,764\,705\,882\,36, -23.359\,273\,571\,472\,35, -9.275\,583\,778\,516\,7, -433.956\,291\,007\,986).$$

步骤 6　由

$$\hat{\boldsymbol{H}} = \boldsymbol{H} + \boldsymbol{e}_1\boldsymbol{g}^{\mathrm{T}},$$

计算可得

$$\boldsymbol{g}^{\mathrm{T}} = \boldsymbol{y} - (h_{1,1}, h_{1,2}, h_{1,3}, h_{1,4}) = (-16.000\,000\,000\,000, -25.287\,037\,615\,123, -4.197\,823\,401\,149, -422.638\,990\,353\,200).$$

步骤 7　计算

$$\boldsymbol{F} = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{Q} = (-16.000\,000\,000\,000, -98.341\,463\,414\,634, 129.634\,146\,341\,464, -390.902\,439\,024\,390).$$

利用算得的 $\boldsymbol{F}$,有

$$\boldsymbol{A} + \boldsymbol{BF} = 10^2\begin{pmatrix} -0.139\,117\,647\,058\,82 & -1.006\,453\,849\,832\,61 & 1.384\,086\,561\,453\,85 & -3.995\,789\,096\,126\,25 \\ -0.026\,764\,705\,882\,35 & 0.021\,078\,431\,372\,55 & -0.120\,490\,196\,078\,43 & 0.108\,529\,411\,764\,71 \\ 0.013\,235\,294\,117\,65 & 0.011\,078\,431\,372\,55 & 0.029\,509\,803\,921\,57 & -0.011\,470\,588\,235\,29 \\ 0.013\,235\,294\,117\,65 & 0.004\,411\,764\,705\,88 & 0.036\,176\,470\,588\,24 & -0.011\,470\,588\,235\,29 \end{pmatrix},$$

可得,该矩阵的特征值为

$$\hat{\lambda}_1 = -0.999\,999\,999\,999\,54,$$

$$\hat{\lambda}_2 = -2.000\,000\,000\,001\,36,$$

$$\hat{\lambda}_3 = -2.999\,999\,999\,998\,89,$$

$$\hat{\lambda}_4 = -4.000\,000\,000\,000\,19.$$

与给定的特征值比较,误差的阶为 $O(\|\boldsymbol{A}+\boldsymbol{BF}\|u)$,此处,$u \approx 1.11 \times 10^{-16}$. 这说明所计算的 $\boldsymbol{F}$ 很精确,算法Ⅱ对本例是非常有效的.

3　算法分析

3.1　算法的运算量分析

如果不考虑将 $\boldsymbol{B}$,$\boldsymbol{A}$ 归化为 $\beta\boldsymbol{e}_1$,$\boldsymbol{H}$ 的运算量,算法Ⅰ的运算量主要花在步骤 1 和步骤 2 上. 在步骤 1 中要解 n 个 $(n-1)$ 阶的系数矩阵为三角形的线性方程组,每个方程组的计算量为 $n(n-1)/2$,因此,总共为 $n^2(n-1)/2 = n^3/2 - n^2/2$;在步骤 2 中要解一个系数矩阵为满矩阵的线性方程组,其计算量为 $n^3/3 + n^2 - n/3$,所以,总的运算量为

$$\frac{5n^3}{6} + \frac{n^2}{2} + \frac{n}{6}.$$

对于算法Ⅱ,假如忽略步骤 1,那么主要花费的运算量在步骤 2 和步骤 4 中. 因为是解系数矩阵为三角形的线性方程组,它们的阶数从 1 阶到 n 阶,因此,总的运算量为

$$\sum_{k=1}^{n} k(k+1)/2 = n^3/6 + n^2/2 + n/3.$$

表 1 所示为已有的几种著名算法的运算量,与之比较,我们给出的算法Ⅰ的运算量居中,而算法Ⅱ的运算量是最少的.

表 1　几种算法的运算

Table 1　Calculation of some algorithm

算法	运算量	稳定性
Miminis 和 Paige[2]	$\frac{5}{6}n^3$	是
Petkov 等[5]	$\frac{5}{3}n^3$	是
Patel 和 Misra[6]	$\frac{5}{6}n^3$	是
Datta[1]	$\frac{1}{6}n^3$	否
Modified Datta[8]	$\frac{5}{6}n^3$	是

4.2　误差分析

我们对算法Ⅰ进行误差分析,首先分析解方程(4)的误差. 令 $\tilde{\boldsymbol{x}}(\lambda_j)$ 为 $\boldsymbol{x}(\lambda_j)$ 的计算结果,计算机精度为 u,有

$$\check{\tilde{\boldsymbol{x}}}(\lambda_j) = (\tilde{x}_1(\lambda_j), \tilde{x}_2(\lambda_j), \cdots, \tilde{x}_{n-1}(\lambda_j))^{\mathrm{T}},$$

则 $\check{\tilde{\boldsymbol{x}}}(\lambda_j)$ 满足

$$\tilde{\boldsymbol{U}}(\lambda_j)\check{\tilde{\boldsymbol{x}}}(\lambda_j) = -\boldsymbol{b}(\lambda_j),$$

式中,

$$\tilde{\boldsymbol U}(\lambda_j)=\boldsymbol U(\lambda_j)+\delta\boldsymbol U(\lambda_j)=\begin{pmatrix} h_{2,1}(1+\delta_{1,2}) & (h_{2,2}-\lambda_j)(1+\delta_{2,2}) & \cdots & h_{2,n-1}(1+\delta_{2,n-1}) \\ \vdots & & & \vdots \\ 0 & \cdots & h_{n-1,n-2}(1+\delta_{n-1,n-2}) & (h_{n-1,n-1}-\lambda_j)(1+\delta_{n-1,n-1}) \\ 0 & \cdots & 0 & h_{n,n-1}(1+\delta_{n,n-1}) \end{pmatrix}.$$

在 $nu<0.01$ 条件下,下式[3]成立:

$$\|\delta\boldsymbol U(\lambda_j)\|_\infty\leqslant g\frac{n(n-1)}{2}1.01u,$$

$$g=\max_{i,k,\lambda_j}|(\boldsymbol U(\lambda_j))_{i,k}|.$$

因此,

$$\|\tilde{\boldsymbol x}(\lambda_j)-\boldsymbol x(\lambda_j)\|_\infty\leqslant\|\boldsymbol U(\lambda_j)^{-1}\|_\infty\|\tilde{\boldsymbol x}(\lambda_j)\|_\infty g\frac{n(n-1)}{2}1.01u.$$

定义

$$\omega=\max_{\lambda_j}\|\boldsymbol U(\lambda_j)^{-1}\|_\infty,\quad \sigma=\max_{\lambda_j}\|\tilde{\boldsymbol x}(\lambda_j)\|_\infty,$$

$$\tau=\omega\sigma g\frac{n(n-1)}{2}1.01u,$$

由此可知,方程(5)的系数矩阵 $\boldsymbol G$ 和右端向量 $\boldsymbol c$ 扰动后为

$$\tilde{\boldsymbol G}=\boldsymbol G+\delta\boldsymbol G,\quad \tilde{\boldsymbol c}=\boldsymbol c+\delta\boldsymbol c,$$

式中,

$$\|\delta\boldsymbol G\|_\infty\leqslant n\tau,\quad \|\delta\boldsymbol c\|_\infty\leqslant\max_j\|\lambda_j\|\tau.$$

假如我们采用部分选主元的高斯消去法解方程(5)求 $\boldsymbol y$,那么,由文献[3]可知,$\boldsymbol y$ 的计算结果 $\tilde{\boldsymbol y}$ 满足方程

$$(\tilde{\boldsymbol G}+\delta\tilde{\boldsymbol G})\tilde{\boldsymbol y}=\tilde{\boldsymbol c},\quad \|\delta\tilde{\boldsymbol G}\|_\infty\leqslant n^2\rho\|\tilde{\boldsymbol G}\|_\infty u,$$

式中,ρ 为增长因子. 由此可知,解 $\tilde{\boldsymbol y}$ 满足方程

$$(\boldsymbol G+\delta\boldsymbol G+\delta\tilde{\boldsymbol G})\tilde{\boldsymbol y}=\tilde{\boldsymbol c},\qquad(13)$$

并且 $\lim\limits_{u\to0}\|\delta\boldsymbol G+\delta\tilde{\boldsymbol G}\|=0$. 这就证明了算法Ⅰ计算 $\boldsymbol y$ 是向后误差稳定的[9].

另一方面,

$$\boldsymbol G(\tilde{\boldsymbol y}-\boldsymbol y)=-\delta\boldsymbol G\tilde{\boldsymbol y}-\delta\tilde{\boldsymbol G}\tilde{\boldsymbol y}+\delta\boldsymbol c\Rightarrow\tilde{\boldsymbol y}-\boldsymbol y=$$

$$\boldsymbol G^{-1}(-\delta\boldsymbol G\tilde{\boldsymbol y}-\delta\tilde{\boldsymbol G}\tilde{\boldsymbol y}+\delta\boldsymbol c)=:(\epsilon_1,\epsilon_2,\cdots,\epsilon_n)^{\mathrm T},$$

$$|\epsilon_i|\leqslant\epsilon:=\|\boldsymbol G^{-1}\|_\infty((n\tau+n^2\rho\|\tilde{\boldsymbol G}\|_\infty u)\|\tilde{\boldsymbol y}\|_\infty+\max_j\|\lambda_j\|\tau).$$

假如 ρ 不是非常大,那么

$$\epsilon_i=O(\kappa(\boldsymbol G)\max_{\lambda_j}\kappa(\boldsymbol U(\lambda_j))n^3u),$$

式中,$\kappa(\boldsymbol S)=\|\boldsymbol S^{-1}\|_\infty\|\boldsymbol S\|_\infty$,记矩阵 $\boldsymbol S$ 的条件数. 令

$$\tilde{\boldsymbol H}=\begin{pmatrix}\tilde y_1 & \tilde y_2 & \cdots & \tilde y_n\\ h_{2,1} & h_{2,2} & \cdots & h_{2,n}\\ \vdots & & & \vdots\\ 0 & \cdots & h_{n,n-1} & h_{n,n}\end{pmatrix}=$$

$$\begin{pmatrix}y_1+\epsilon_1 & y_2+\epsilon_2 & \cdots & y_n+\epsilon_n\\ h_{2,1} & h_{2,2} & \cdots & h_{2,n}\\ \vdots & & & \vdots\\ 0 & \cdots & h_{n,n-1} & h_{n,n}\end{pmatrix},$$

$$\tilde{\boldsymbol H}=\begin{pmatrix}y_1 & y_2 & \cdots & y_n\\ h_{2,1} & h_{2,2} & \cdots & h_{2,n}\\ \vdots & & & \vdots\\ 0 & \cdots & h_{n,n-1} & h_{n,n}\end{pmatrix}+\begin{pmatrix}\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n\\ 0 & 0 & \cdots & 0\\ \vdots & & & \vdots\\ 0 & \cdots & 0 & 0\end{pmatrix}=$$

$$\hat{\boldsymbol H}+\boldsymbol E.$$

下面,我们证明计算结果 $\tilde{\boldsymbol H}$ 的 n 个特征值逼近给定的极点 $\lambda_1,\lambda_2,\cdots,\lambda_n$,即矩阵 $\hat{\boldsymbol H}$ 的 n 个特征值. 记 $\tilde{\boldsymbol H}$ 的 n 个特征值为 $\mu_1,\mu_2,\cdots,\mu_n$,2 个矩阵特征值的配对距离为

$$d=\min_\sigma\max_j|\lambda_j-\mu_{\sigma(j)}|,$$

式中,σ 为 $\{1,2,\cdots,n\}$ 置换全体. 利用如下著名结果:

引理1[10] 对任意 $n\times n$ 矩阵 $\boldsymbol A$ 和 $\boldsymbol E$,如果 $\boldsymbol A$ 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$,$\boldsymbol A+\boldsymbol E$ 的特征值为 $\mu_1,\mu_2,\cdots,\mu_n$,则 2 组特征值之间的配对距离为

$$d\leqslant3.46(2M)^{1-\frac1n}\|\boldsymbol E\|^{\frac1n},$$

式中,

$$M=\max(\|\boldsymbol A\|,\|\boldsymbol A+\boldsymbol E\|).$$

由引理1,因为 $\tilde{\boldsymbol H}=\hat{\boldsymbol H}+\boldsymbol E$,

$$\boldsymbol E=\begin{pmatrix}\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n\\ 0 & 0 & \cdots & 0\\ \vdots & & & \vdots\\ 0 & \cdots & 0 & 0\end{pmatrix},$$

并且 $\lim\limits_{u\to0}\|\boldsymbol E\|^{\frac1n}=0$,所以有如下定理成立.

定理 1　对任意 $\eta>0$,可以找到一个实数 ϵ,当计算机精度 u 满足

$$|u| \leqslant \epsilon$$

时,则有

$$d \leqslant \eta.$$

由定理 1 可知,只要精度 $|u|$ 足够小,计算结果特征值都可以达到给定的正的误差界.

参考文献:

[1] DATTA B N. An algorithm to assign eigenvalues in Hessenberg matrix: single input case [J]. IEEE Trans on Automat Contr, 1987, 32(5):414-417,

[2] MIMINIS G S, PAIGE C C. An algorithm for pole assignment of time invariant linear systems [J]. Int J Contr, 1982, 35:341-354.

[3] FORSYTHE G E, MOLER C B. Computer solution of linear algebraic systems [M]. New York: Prentice-Hall, 1967.

[4] MIMINIS S G S, PAIGE C C. A direct algorithm for pole assignment of time-invariant multi-input linear systems using state feedback [J]. Automatica J IFAC, 1988, 24:343-356.

[5] PETKOV P H, CHRISTOR N D, KONSTANTINOV M M. A computation algorithm for pole assignment of linear single-input systems [J]. IEEE Trans on Automat Contr, 1984, 29:1045-1048.

[6] PATEL R V, MISRA P. Numerical algorithm for eigenvalue assignment by state feedback [J]. Proc IEEE, 1984, 72(12):1755-1764.

[7] VARGA A. A multishift Hessenberg method for pole assignment of single-input system [J]. IEEE Trans on Automat Contr, 1996, 12:1795-1799.

[8] ARNOLD M E. Algorithm and conditioning for eigenvalue assignment [D]. Illinois: Northern Illinois University, 1992.

[9] 唐珍. 舍入误差分析引论[M]. 上海:上海科技出版社,1987.

[10] KRAUSE G M. Bounds for the variation of matrix eigenvalues and polynomial roots [J]. Linear Algebra Appl, 1994, 208(15):73-82.

(编辑:孟庆勋)