On ill-conditioned eigenvalues, multiple roots of polynomials, and their accurate computations

Zhonggang Zeng*

September 11, 1998

Abstract

Algebraic eigenvalues with associated left and right eigenvectors (nearly) orthogonal, and polynomial roots that are multiple, have been known to be sensitive to perturbations in numerical computation and thereby ill-conditioned. By constructing an extended polynomial system, it is proved that, under certain conditions, traditionally defined ill-conditioned eigenvalues and polynomial roots can be remarkably stable and insensitive to perturbations, if they are computed as numerical multiple eigenvalues/roots. Furthermore, using the extended polynomial system which is easy to solve, those numerical multiple eigenvalues can be computed with high forward accuracy, while backward errors are estimated as by-products through auxiliary variables of the extended system.

Keywords: eigenvalue, ill-condition, polynomial, multiple root

AMS subject classifications: 65F15, 65F35, 65H05, 65H17

1 Introduction

It is known in classical numerical linear algebra theories that an algebraic eigenvalue, whose associated left and right eigenvectors are nearly orthogonal, is sensitive to perturbations and thereby ill-conditioned [5, 23]. For ill-conditioned eigenvalues, existing

*Department of Mathematics, Northeastern Illinois University, Chicago, IL 60625-4699, email: zzeng@neiu.edu. Research at MSRI, Berkeley is supported in part by Northeastern Illinois University COR Grant 1-30906.

numerical methods generally give answers with large (forward) errors. Similar condition theories and computational difficulties exist for multiple roots of polynomials [3, 22]. In fact, if a matrix A has ill-conditioned eigenvalues, it is a close neighbor of a matrix with multiple eigenvalues [15, 24]. Namely,

$$\det(A - \lambda I) = (\lambda - \lambda_*)^m p(\lambda) + \varepsilon(\lambda), \quad m > 1,$$

with small perturbation $\varepsilon(\lambda)$ from the polynomial $(\lambda - \lambda_*)^m p(\lambda)$, which has an m-fold eigenvalue λ_*. The behavior of ill-conditioned eigenvalues can be illustrated by the matrix

$$A = (a_{ij}) = \begin{pmatrix} 1 & & & \text{\Large 0} \\ 1 & 1 & & \\ & \ddots & \ddots & \\ \text{\Large 0} & & 1 & 1 \end{pmatrix}_{100\times 100}, \tag{1}$$

A tiny perturbation $\varepsilon = 10^{-10}$ at the entry $a_{1,100}$ results in a perturbed characteristic polynomial $(\lambda - 1)^{100} - 10^{-10}$ and an (forward) error $10^{-0.1} \approx 0.79$ from the 100-fold eigenvalue $\lambda = 1$. Although the computational result 1.794328234724 can hardly be considered an approximation to the target eigenvalue 1.0, it is an exact solution to a nearby problem. In other words, for a given matrix A, one can only expect a numerical algorithm to calculate *exact* eigenvalues of another matrix $A + E$ with an inevitable tiny noise matrix E from inherent round-off errors. If the perturbation E is small enough, the computational results and the algorithm are of high *backward accuracy*. The large *forward error*, or the actual distance between the exact eigenvalue and the calculated one, is attributed to the sensitivity of the problem instead of the algorithm. "If the answer is highly sensitive to perturbations, you have probably asked the wrong question" [19], meaning that the eigenproblem is from a poor model with a wrong matrix to begin with.

Polynomial root finding is one of the oldest mathematical problems with a huge volume of literature. More than 3000 articles and books on this topic were listed in [12] and a related website[1]. Many of them are dealing with multiple roots, such as [9, 13, 14, 17, 18, 25]. There has been an "attainable accuracy" barrier for numerical multiple-root solvers [9, 25]: if the polynomial $f(x) = (x - x_*)^m p(x)$ and k is the number of correct digits attainable in evaluating $f(x)$, then the multiple root x_* can only be calculated to the accuracy of k/m correct digits. With IEEE standard double precision, one can evaluate function values up to 16 digits, making multiple roots of

[1] As this paper in writing, the website is
`http://www.elsevier.nl/homepage/sac/cam/mcnamee/index.html`

16 plus multiplicities not worth trying since not even a single digit from the result can be trusted anyway.

Base on these condition theories, every multiple root of a polynomial, and every eigenvalue of a matrix corresponding to a non-linear elementary divisor are automatically ill-conditioned. It seems that we are giving up too many problems for being "wrong". In this paper, we shall argue that, $\lambda = 1$ is an ill-conditioned eigenvalue of matrix A in (1) only because it is regarded and computed as a *simple* eigenvalue. It is a remarkably well-conditioned *numerical multiple eigenvalue* that can be, and should be, computed to high accuracy.

In later sections, we shall prove that ill-conditioned eigenvalues or polynomial multiple roots are *not* sensitive to perturbations under certain conditions. Furthermore, they can be accurately calculated via solving an extended polynomial system, with auxiliary variables estimating backward errors of the eigenvalues. Our proposed method is tested with defective and derogatory algebraic eigenvalues, i.e. multiple polynomial roots, with multiplicity in hundreds and accurately locates multiple roots to 12 digits and beyond.

The extended polynomial system is also applicable in determining the Jordan canonical structure of a matrix. One of the most difficult issues in numerical computation of Jordan canonical form is to decide whether a cluster of computed eigenvalues is actually a multiple eigenvalue [1, 6, 10]. We provide a practical way for this purpose.

2 Multiple roots of a polynomial

From common definitions, a point $x_* \in \boldsymbol{C}$ is an m-fold root (or zero) of the polynomial f if

$$f(x_*) = f'(x_*) = \cdots = f^{(m-1)}(x_*) = 0, \quad \text{and} \quad f^{(m)}(x_*) \neq 0$$

x_* is a simple root if $m = 1$ and multiple root if $m \geq 2$. The integer m is the multiplicity of x_*. Namely, if x_* is an m-fold root of f, we can write $f(x) = (x - x_*)^m g(x - x_*)$, where g is a lower ordered polynomial with nonzero constant term $g(0)$. Theoretically [20], when $m \geq 2$, Newton's iteration

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad k = 0, 1, \cdots \tag{2}$$

converges linearly and the modified Newton's iteration

$$x_{k+1} = x_k - m\frac{f(x_k)}{f'(x_k)} \quad k = 0, 1, \cdots \tag{3}$$

converges quadratically to the m-fold root x_* of f. In numerical computation, it is almost a certainty that neither iteration (2) nor (3) converges near x_* unless the multiplicity m is small or $f(x)$ is explicitly evaluated in the factored form $(x-x_*)^m(b_0+b_1x+b_2x^2+\cdots)$. QR iteration [4, 5, 23] on the companion matrix would also show similar devastating forward errors. To see why Newton's iterations above fail, let's consider the floating point evaluations of $f(x)=(x-x_*)^m g(x-x_*)$ and its derivative at x near x_*

$$\begin{aligned} fl[f(x)] &= (x-x_*)^m\tilde{g}(x-x_*)+(x-x_*)^{m-1}\varepsilon_{m-1}(x)+\cdots+(x-x_*)\varepsilon_1(x)+\varepsilon_0(x) \\ fl[f'(x)] &= (x-x_*)^{m-1}\hat{g}(x-x_*)+(x-x_*)^{m-2}\sigma_{m-2}(x)+\cdots+(x-x_*)\sigma_1(x)+\sigma_0(x) \end{aligned}$$

where $\varepsilon_i(x)$, $\sigma_i(x)$, $i=0,1,\cdots$, are unpredictable tiny numbers corresponding to x, as results of rounding errors in evaluations. When x is close to x_*, the computed Newton correction

$$\frac{fl[f(x)]}{fl[f'(x)]}\approx\frac{\varepsilon_0(x)}{\sigma_0(x)}$$

behaves like a random number generator and has little meaningful relationship to original data.

There are many other numerical methods intended for multiple roots of polynomials [9, 13, 14, 17, 18, 20, 25]. They all suffer similar limitations and apparently can only applied to finding polynomial roots of low multiplicity.

Newton's iterations (2) and (3) and other multiple-root finders fail in numerical computation of multiple roots because they are essentially applied to compute *simple* roots, where numerically nonzero f' near x_* is expected in the denominator. To calculate multiple roots of a polynomial, the fact that $f^{(j)}(x_*)=0$, $j=0,1,\cdots,m-1$ must be taken into consideration. Based on this observation, we construct an extended polynomial operator $F:\boldsymbol{C}^m\longrightarrow\boldsymbol{C}^m$ associated with the polynomial $f:\boldsymbol{C}\longrightarrow\boldsymbol{C}$, such that an m-fold root of f is a simple zero of F. Here, from common definitions, a point $z^*=(z_1^*,\cdots,z_n^*)$ is a simple zero of $F:\boldsymbol{C}^n\longrightarrow\boldsymbol{C}^n$ if $F(z^*)=0$ and the Jacobian $DF(z^*)\equiv\left(\frac{\partial f_i}{\partial z_j}(z^*)\right)$ is a nonsingular $n\times n$ matrix.

Theorem 1 *A complex number $x_*\in\boldsymbol{C}$ is an m-fold root of a polynomial f if and only if the point $(0,0,\cdots,0,x_*)^\top\in\boldsymbol{C}^m$ is a simple zero of the extended polynomial operator $F:\boldsymbol{C}^m\longrightarrow\boldsymbol{C}^m$ associated with f with auxiliary variables $t_1,\cdots,t_{m-1}$,* *defined as*

$$F\begin{pmatrix} t_1 \\ \vdots \\ t_{m-1} \\ x \end{pmatrix} = \begin{pmatrix} \frac{f(x)}{0!} & - & t_1\frac{f^{(m)}(x)}{m!} \\ \frac{f'(x)}{1!} & - & t_2\frac{f^{(m)}(x)}{m!} \\ & \vdots & \\ \frac{f^{(m-2)}(x)}{(m-2)!} & - & t_{m-1}\frac{f^{(m)}(x)}{m!} \\ f^{(m-1)}(x) & & \end{pmatrix}. \tag{4}$$

Proof: Since $F\left((0,\cdots,0,x)^\top\right) = \left(\frac{f(x)}{0!}, \frac{f'(x)}{1!}, \cdots, \frac{f^{(m-2)}(x)}{(m-2)!}, f^{(m-1}(x)\right)^\top$, the point $(0,0,\cdots,0,x_*)^\top$ is a zero of F if and only if

$$f(x_*) = f'(x_*) = \cdots = f^{(m-1)}(x_*) = 0.$$

Moreover, the Jacobian

$$DF\left((0,\cdots,0,x_*)^\top\right) = diag\left(\frac{f^{(m)}(x_*)}{m!}, \cdots, \frac{f^{(m)}(x_*)}{m!}, f^{(m)}(x_*)\right)$$

is nonsingular if and only if $f^{(m)}(x_*) \neq 0$, where $diag(\cdots)$ denotes the diagonal matrix with listed entries. QED.

Corollary 1 *Let x_* be an m-fold root of $f \in C^{m+1}(\Omega)$. Then there is a neighborhood N of x_* in Ω such that Newton's iteration*

$$z_{k+1} = z_k - [DF(z_k)]^{-1}F(z_k), \quad k = 0, 1, \cdots,$$

converges to $(0,\cdots,0,x_*)^\top$ from every $z_0 = (0,0,\cdots,0,x_0)^\top \in \boldsymbol{C}^{m-1} \times N$.

Proof Standard theory of Newton's iteration. QED.

3 Conditioning of polynomial roots

As mentioned above, a polynomial with an m-fold root x_* can be written as $(x - x_*)^m g(x - x_*)$ with $g(0) \neq 0$. In numerical computation, we expect the backward accuracy. That is, whenever we calculate an approximate root of f, we are evaluating an exact root of $\hat{f}(x) = f(x) + h(x)$ with polynomial $h(x)$ "small" in some sense. If f has an m-fold root x_*, we can further assume that

$$h(x) = (x - x_*)^{m-1}\varepsilon_{m-1} + \cdots + (x - x_*)\varepsilon_1 + \varepsilon_0$$

without loss of generality because higher order terms of $(x - x_*)$ can be absorbed into $(x - x_*)^m g(x - x_*)$ while maintaining the multiplicity of x_* as a root of f. In this section, we consider the polynomial

$$f(x) = (x - x_*)^m g(x - x_*) + (x - x_*)^{m-1}\varepsilon_{m-1} + \cdots + (x - x_*)\varepsilon_1 + \varepsilon_0 \tag{5}$$

where g is a polynomial with nonzero constant term $g(0)$. We are interested in cases where

$$|g(0)| \gg \max_{0 \le i \le m-1} \{|\varepsilon_i|\}$$

which implies that $f(x)$ is "near" the polynomial $(x - x_*)^m g(x - x_*)$ that has an m-fold root x_*.

Lemma 1 *Let $\{a_1, a_2, a_3, \cdots, a_n\} \subset \boldsymbol{C}$ with $a_1 \neq 0$. Then there is a $\delta > 0$ such that for every θ with $|\theta| < \delta |a_1|$, the polynomial*

$$f(x) = \theta + a_1 x + a_2 x^2 + a_3 x^2 + \cdots + a_n x^n$$

has a root $\sigma = -\dfrac{\theta}{a_1} + O(|\theta|^2)$

Proof Let $\hat{a}_k = \max_{|x| \le 1} \left\{ \dfrac{\left|f^{(k)}(x)\right|}{k!} \right\}$, $k = 2, 3, \cdots, n$. Notice that $\hat{a}_k$'s are independent of θ. Let $\delta = \min \left\{ \dfrac{1}{2}, \left(\dfrac{4}{|a_1|} \sum_{k=2}^{n} k \hat{a}_k \right)^{-1} \right\}$, and $\sigma_0 = -\dfrac{\theta}{a_1}$ where $|\theta| < \delta |a_1|$. From the Taylor expansion of $f(x)$ at σ_0, we have

$$f(x) = a_{00} + a_{01}(x - \sigma_0) + \cdots + a_{0n}(x - \sigma_0)^n \quad \text{where} \quad a_{0k} = \frac{f^{(k)}(\sigma_0)}{k!}, \quad k = 0, 1, \cdots, n$$

and thus $|a_{0k}| < \hat{a}_k$, $k = 2, 3, \cdots, n$, Particularly, because $\sigma_0 \in (-\delta, \delta)$, we have inequalities $|\sigma_0| \sum_{k=2}^{n} \hat{a}_k \le |\sigma_0| \sum_{k=2}^{n} k \hat{a}_k < \dfrac{1}{4}|a_1|$, implying

$$\begin{aligned}
|a_{00}| &= \left| (\theta + a_1\sigma_0) + \sum_{j=2}^{n} a_j \sigma_0^j \right| \le \left|\sigma_0^2\right| \left| \sum_{j=2}^{n} a_j \sigma_0^{j-2} \right| \le |\sigma_0| \left(|\sigma_0| \sum_{k=2}^{n} \hat{a}_k \right) < \frac{1}{4} |\sigma_0| |a_1| \\
|a_{01}| &= \left| a_1 + \sum_{j=2}^{n} j a_j \sigma_0^{j-1} \right| \ge |a_1| - |\sigma_0| \sum_{k=2}^{n} k \hat{a}_k > \frac{3}{4} |a_1|.
\end{aligned}$$

Let $\sigma_1 = -\dfrac{a_{00}}{a_{01}}$, we have $|\sigma_1| < \dfrac{\frac{1}{4}|\sigma_0||a_1|}{\frac{3}{4}|a_1|} = \dfrac{1}{3}|\sigma_0| < \delta$. Generally, with mathematical induction, we can write

$$f(x) = a_{k0} + a_{k1}\left(x - \sum_{j=0}^{k}\sigma_j\right) + \cdots + a_{kn}\left(x - \sum_{j=0}^{k}\sigma_j\right)^n \quad \text{with} \quad \sigma_k = -\frac{a_{k-1,0}}{a_{k-1,1}}$$

where $a_{kj} = \frac{1}{j!}f^{(j)}\left(\sum_{l=0}^{k}\sigma_l\right)$, $j = 0, 1, \cdots, n$. Particularly,

$$\begin{aligned} |a_{k0}| &= \left| a_{k-1,0} + a_{k-1,1}\sigma_k + \sum_{j=2}^{n} a_{k-1,j}\sigma_k^j \right| \le |\sigma_k|\left(|\sigma_k|\sum_{j=2}^{n}\hat{a}_j\right) < \frac{1}{4}\left(\frac{1}{3}\right)^k |\sigma_k||a_1| \\ |a_{k1}| &= \left| a_{k-1,1} + \sum_{j=2}^{n} j a_{k-1,j}\sigma_k^{j-1} \right| \ge |a_{k-1,1}| - |\sigma_k|\sum_{j=2}^{n} j\hat{a}_j > |a_{k-1,1}| - \frac{1}{4}\left(\frac{1}{3}\right)^k |a_1| \\ &> \left[1 - \frac{1}{4}\sum_{j=0}^{k}\left(\frac{1}{3}\right)^j\right]|a_1| > \frac{5}{8}|a_1| \end{aligned}$$

and

$$\sigma_{k+1} = -\frac{a_{k0}}{a_{k1}}, \quad \text{with} \quad |\sigma_{k+1}| < \frac{\frac{1}{4}\left(\frac{1}{3}\right)^k |\sigma_k||a_1|}{\frac{5}{8}|a_1|} < \frac{1}{3}|\sigma_k|, \qquad k = 1, 2, \cdots.$$

Thus $\sigma = \sum_{j=0}^{\infty}\sigma_j$ converges and the inequality $|a_{k0}| < \frac{1}{4}\left(\frac{1}{3}\right)^k |\sigma_k||a_1|$ yields

$$f(\sigma) = \lim_{k\to\infty} f\left(\sum_{j=0}^{k}\sigma_j\right) = \lim_{k\to\infty} a_{k0} = 0$$

Also, $\sigma = \sigma_0 + \sum_{j=1}^{\infty}\sigma_j = -\dfrac{\theta}{a_1} + \sum_{j=1}^{\infty}\sigma_j$, while

$$\left|\sum_{j=1}^{\infty}\sigma_j\right| < \frac{3}{2}|\sigma_1| = \frac{3}{2}\left|\frac{a_{00}}{a_{01}}\right| < \frac{3}{2}\frac{|\sigma_0^2|\sum_{j=2}^{n} j\hat{a}_j}{\frac{3}{4}|a_1|} = O(|\theta|^2).$$

QED.

The following theorem establishes the sensitivity of a multiple root of a polynomial as a simple zero of the extended polynomial operator (4).

Theorem 2 *Let $p(x)$ be a polynomial with an $m-fold$ root x_*, i.e, $p(x) = (x - x_*)^m g(x - x_*)$ with $g(0) \neq 0$. Let*

$$h(x) = \varepsilon_0 + (x - x_*)\varepsilon_1 + (x - x_*)^2\varepsilon_2 + \cdots$$

be a polynomial with $\varepsilon_j \in \boldsymbol{C}$, $j = 0, 1, 2, \cdots$. Then there is a $\delta > 0$ such that if $|\varepsilon_j| < \delta$ for $j = 0, 1, 2, \cdots$, the extended polynomial operator (4) associated with the perturbed polynomial $f(x) = p(x) + h(x)$ has a simple zero

$$\begin{pmatrix} t_1^* \\ t_2^* \\ \vdots \\ t_{m-1}^* \\ x^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ x_* \end{pmatrix} + \frac{1}{g(0)} \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_{m-2} \\ \frac{\varepsilon_{m-1}}{m} \end{pmatrix} + O(\varepsilon^2)$$

where $\varepsilon = \max_{0 \le j \le m-1} \{|\varepsilon_j|\}$.

Proof Consider

$$f^{(m-1)}(x) = \frac{(m-1)!}{0!}\varepsilon_{m-1} + \frac{m!}{1!}\frac{g(0)}{0!}(x - x_*) + \frac{(m+1)!}{2!}\frac{g'(0)}{1!}(x - x_*)^2 + \cdots.$$

From Lemma 1, $f^{(m-1)}(x)$ has a root at

$$x - x_* = -\frac{(m-1)!\varepsilon_{m-1}}{m!g(0)} + O(\varepsilon^2)$$

i.e., $x^* = x_* - \dfrac{\varepsilon_{m-1}}{mg(0)} + O(\varepsilon^2)$. From (4),

$$t_j^* = \frac{m!f^{(j-1)}(x^*)}{(j-1)!f^{(m)}(x^*)} = \frac{\varepsilon_{j-1} + O(\varepsilon^2)}{g(0) + O(\varepsilon)} = \frac{\varepsilon_{j-1}}{g(0)} + O(\varepsilon^2), \quad j = 1, 2, \cdots, m-1.$$

The point $(t_1^*, \cdots, t_{m-1}^*, x^*)^\top$ is a simple zero of the operator F in (4) since the Jacobian

$$DF \begin{pmatrix} t_1^* \\ t_2^* \\ \vdots \\ t_{m-1}^* \\ x^* \end{pmatrix} = \begin{pmatrix} \frac{f^{(m)}(x^*)}{m!} & & & & \frac{f'(x^*)}{0!} - t_1^*\frac{f^{(m+1)}(x^*)}{m!} \\ & \frac{f^{(m)}(x^*)}{m!} & & & \frac{f''(x^*)}{1!} - t_2^*\frac{f^{(m+1)}(x^*)}{m!} \\ & & \ddots & & \vdots \\ & & & \frac{f^{(m)}(x^*)}{m!} & \frac{f^{(m-1)}(x^*)}{(m-2)!} - t_1^*\frac{f^{(m+1)}(x^*)}{m!} \\ & & & & f^{(m)}(x^*) \end{pmatrix}$$

with $f^{(m)}(x^*) = m![g(0) + \varepsilon_m] + O(\varepsilon^2) \neq 0$ for small enough δ. QED.

Let $f(x)$ be a polynomial and x_* one of its roots. The condition of x_* as a root of $f(x)$ has traditionally been determined [23] by $\frac{1}{f'(x_*)}$. If $f(x)$ is subject to perturbation $h(x) = \varepsilon_0 + (x - x_*)\varepsilon_1 + (x - x_*)^2\varepsilon_2 + \cdots$, the corresponding perturbed root $x_* + \Delta x_*$ of

$$f(x) + h(x) = \varepsilon_0 + (x - x_*)\left(f'(x_*) + \varepsilon_1\right) + (x - x_*)^2 \left(\frac{f''(x_*)}{2!} + \varepsilon_2\right) + \cdots$$

satisfies $|\Delta x_*| = \frac{1}{f'(x_*)}|\varepsilon_0| + O(\varepsilon^2)$, with $\varepsilon = max_i\{|\varepsilon_i|\}$, according to Lemma 1. Therefore, if x_* is a multiple root of f, it is automatically ill-conditioned. Moreover, the (exact) *simple* roots x_j^*, $j = 1, 2, \cdots, m$, of $f(x)+h(x)$ corresponding to the m-fold root x_* of f are [22, pp39-40]

$$x_j^* = x_* + \left[\left(-\frac{\varepsilon_0}{g(0)}\right)^{\frac{1}{m}}\right]_j + O\left(\varepsilon^{\frac{2}{m}}\right), \quad j = 1, 2, \cdots, m$$

where $[(\cdot)^{1/m}]_j$ denotes the $j - th$ interpretation of the $(\cdot)^{1/m}$. This error estimate further confirm that computing *simple* roots of a polynomial around a multiple root is an ill-conditioned problem.

Theorem 2 shows that, however, if we consider an m-fold root of the polynomial, along with $m - 1$ auxiliary variables $t_1 = t_2 = \cdots = t_{m-1} = 0$, as a simple root of the extended polynomial operator (4), a multiple root of a polynomial may not be sensitive to perturbation at all. Actually, from Theorem 2, a small perturbation $h(x)$ upon the the polynomial $(x - x_*)^m g(x - x_*)$ results in a *forward* error proportional to the magnitude of the perturbation. Therefore the m-fold root x_* is *well-conditioned.* Moreover, not only x_* can be computed within an error proportional to the magnitude of the perturbation, the actual perturbation, or the backward error, is well measured by the auxiliary variables $t_1, \cdots, t_{m-1}$, via calculating the simple zero of the extended polynomial system (4).

More specifically, if $g(0) = \frac{f^{(m)}(x_*)}{m!}$ is not small, the m-fold root x_* is numerically stable and the forward error of the computed root x^* is proportional to $\frac{\varepsilon_{m-1}}{m} = \frac{h^{(m-1)}(x_*)}{m!}$, each auxiliary variable t_i has a deviation from 0 proportional to ε_i, $i = 1, \cdots, m - 1$, all with ratio $\frac{1}{g(0)} = \frac{m!}{f^{(m)}(x_*)}$. This ratio effectively describes the sensitivity of x_* as an m-fold root of $f(x)$. Therefore we can consider x_* *well-conditioned* if

$$\left|\frac{f^{(m)}(x_*)}{m!}\right| \gg \left|\frac{f^{(k)}(x_*)}{k!}\right|, \quad k=1,2,\cdots,m-1.$$

Definition 1 *A complex number x_* is a well conditioned numerical m-fold root of a polynomial f if*

$$\left|\frac{f^{(m)}(x_*)}{m!}\right| \gg \left\|\left(\frac{f(x_*)}{0!},\frac{f'(x_*)}{1!},\ldots,\frac{f^{(m-1)}(x_*)}{(m-1)!}\right)^\top\right\|_\infty .$$

Otherwise, it is an ill-conditioned numerical m-fold root of f.

A traditionally defined ill-conditioned root is an ill-conditioned *simple* root by Definition 1. Every numerical multiple root is automatically sensitive to perturbations if it is subject to numerical computation as a simple one, while not necessarily ill-conditioned in the sense of Theorem 2 and Definition 1.

Theorem 2 establishes the forward error of the approximate solution x^* compared to the numerical multiple root x_* in (5) and the meaning of the auxiliary variables t_j's. We shall claim further that, if $(t_1^*,\cdots,t_{m-1}^*,x^*)$ is a simple zero of the extended polynomial operator F in (4) with tiny t_j^*'s, then x^* is an *exact* m-fold root of a polynomial close to f, and we know exactly how far it is from f toward that polynomial. In other words, the computed solution x^* is itself a numerical multiple root of f and is *backward accurate.*

Theorem 3 *Let $f(x)$ be a polynomial with associated polynomial operator F defined as in (4). Let $(t_1^*,\cdots,t_{m-1}^*,x^*)^\top \in \boldsymbol{C}^m$ be a simple zero of F. then there is a polynomial $\hat{g}(x-x_*)$ with $\hat{g}(0)\neq 0$ such that*

$$\begin{aligned} f(x) &= (x-x^*)^m\hat{g}(x-x^*) \\ &\quad+\left[\hat{g}(0)t_{m-1}^*\right](x-x^*)^{m-2}+\cdots+\left[\hat{g}(0)t_2^*\right](x-x^*)+\hat{g}(0)t_1^*. \end{aligned} \tag{6}$$

Proof From Taylor's expansion of $f(x)$ at x^*,

$$\begin{aligned} f(x)=\sum_{k=0}^{n}c_k(x-x^*)^k &= (x-x^*)^m\left(c_m+\sum_{j=1}^{n-m}c_{m+j}(x-x^*)^j\right) \\ &\quad+c_{m-1}(x-x^*)^{m-1}+\cdots+c_1(x-x^*)+c_0, \end{aligned}$$

$$\text{with}\quad c_k=\frac{f^{(k)}(x^*)}{k!}, \qquad k=0,1,\cdots,n$$

Since $(t_1^*, \cdots, t_{m-1}^*, x^*)$ is a simple zero of F in (4), we have

$$\hat{g}(x - x_*) = c_m + \sum_{j=1}^{n-m} c_{m+j}(x - x^*)^j \quad \text{with} \quad \hat{g}(0) = c_m = \frac{f^{(m)}(x^*)}{m!} \neq 0$$

$$c_{m-1} = \frac{f^{(m-1)}(x^*)}{(m-1)!} = 0,$$

$$c_j = \frac{f^{(j)}(x^*)}{j!} = t_j^* \frac{f^{(m)}(x^*)}{m!} = \hat{g}(0)t_j^*, \quad j = 0, 1, \cdots, m-2$$

QED.

The polynomial f in (5) with "small" h is, in its written form, near the polynomial $(x - x_*)^m g(x - x_*)$ and thereby has a numerical m-fold root x_*. In fact, f is near a family of polynomials $(x - \hat{x})^m g_{\hat{x}}(x - \hat{x})$ since we can always write

$$f(x) = (x - \hat{x})^m g_{\hat{x}}(x - \hat{x}) + \sigma_{m-1}(\hat{x})(x - \hat{x})^{m-1} + \cdots + \sigma_1(\hat{x})(x - \hat{x}) + \sigma_0(\hat{x})$$

for every $\hat{x} \in \boldsymbol{C}$, with $\sigma_j(\hat{x}) = \frac{f^{(j)}(\hat{x})}{j!}$. Finding a simple zero of F in (4) is picking a particular $\hat{x} = x^*$ such that $\sigma_{m-1}(\hat{x}) = 0$. Theorem 2 guarantees that this $\hat{x} = x^*$ is indeed close to x_* we are looking for, while Theorem 3 ensures that the backward error

$$f(x) - (x - x^*)\hat{g}(x - x^*) = \left[\hat{g}(0)t_{m-1}^*\right](x - x^*)^{m-2} + \cdots + \left[\hat{g}(0)t_2^*\right](x - x^*) + \hat{g}(0)t_1^*$$

is *known* to be tiny as by-product of solving $F(z) = 0$.

4 Conditioning of algebraic eigenvalues

Let A be an $n \times n$ matrix. Its eigenvalues are roots of the characteristic polynomial $\det(A - \lambda I)$ of A. For an eigenvalue λ, its associated (right) eigenvector x and the left eigenvector y are vectors in $\boldsymbol{C}^n$ such that

$$(A - \lambda I)x = 0, \quad \text{and} \quad y^H(A - \lambda I) = 0$$

respectively, where y^H is the conjugate transpose of y. Assuming further that x and y are normalized, the condition number of the eigenvalue λ is defined as [5, 23]

$$cond(\lambda, A) = \frac{1}{|y^H x|}, \quad \|x\|_2 = \|y\|_2 = 1.$$

When $cond(\lambda, A) \gg 1$, the eigenvalue λ has been called ill-conditioned, and A is close to a matrix $\hat{A}$ with λ as an exact multiple eigenvalue [15, 24]. To be more specific, Wilkinson [24] proved that for $\varepsilon = \left|y^H x\right|$, there is a matrix E such that

$$\|E\|_2 \leq \frac{\varepsilon \|A\|_2}{1-\varepsilon^2}$$

and λ is an exact multiple eigenvalue of $A+E$. On the other hand, if A has an multiple eigenvalue with ascent (i.e., the size of the largest Jordan block associated with the multiple eigenvalue) l, then for $\|\Delta A\|_2$ small enough, given any eigenvalue λ' of $A + \Delta A$, there is an eigenvalue λ such that [2, 3]

$$|\lambda' - \lambda| \leq 2(\|V\|_2 \|V^{-1}\|_2 \|X_*\|_2)^{\frac{1}{l}} \|\Delta A\|_2^{\frac{1}{l}}$$

where V is a Jordan basis of A and X_* is a basis of the left invariant subspace. For very ill-conditioned eigenvalues, traditional algorithms like Francis' QR [4, 23] generally give answers that make no sense in terms of forward accuracy. We have been satisfied with the explanation that the calculated eigenvalues, no matter how ridiculously inaccurate, are all *simple* ones of a neighboring matrix. However, based on our analysis, those "ill-conditioned" eigenvalues may be very much well conditioned in a deeper sense. Given the fact that we have to evaluate the eigenvalues of a neighboring matrix, it is apparently more meaningful to evaluate the eigenvalues of *that* neighboring matrix with exact and *well-conditioned* multiple eigenvalues.

Through an orthogonal transformation P, the matrix A is similar to an upper-Hessenberg matrix

$$P^H A P = B = (b_{jk}) = \begin{pmatrix} * & * & * & \cdots & \cdots & * \\ b_{21} & * & * & \cdots & \cdots & * \\ & b_{32} & * & \cdots & \cdots & * \\ & & \ddots & \ddots & \cdots & \vdots \\ & 0 & & b_{n-1,n-2} & * & * \\ & & & & b_{n,n-1} & * \end{pmatrix}, \quad P^H P = I.$$

Without loss of generality, we may assume that B is irreducible, namely, the subdiagonal entries $b_{j,j-1} \neq 0$, $j = 2, 3, \cdots, n$. In fact, if anyone of those subdiagonal entries is zero or tiny, the eigenproblem of A can be reduced to those of the two submatrices. For irreducible upper-Hessenberg matrices, Hyman's method [7, 23], which evaluates the characteristic polynomial of A and its derivatives, can be described as follows

[11]: for every $\lambda \in \boldsymbol{C}$, let $\frac{f^{(0)}(\lambda)}{0!} \equiv f(\lambda) \in \boldsymbol{C}$ and $x_0 = (x_{01}, \cdots, x_{0n})^\top \in \boldsymbol{C}^n$ satisfy

$$\begin{pmatrix} -1 & & & & \\ 0 & & & & \\ \vdots & & \boxed{B - \lambda I} & & \\ 0 & & & & \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{f^{(0)}(\lambda)}{0!} \\ x_{01} \\ \vdots \\ x_{0n} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{7}$$

Then

$$f(\lambda) \equiv \frac{\det[B - \lambda I]}{\prod_{k=1}^{n-1}(-b_{k+1,k})} = \frac{\det[A - \lambda I]}{\prod_{k=1}^{n-1}(-b_{k+1,k})}, \qquad \text{for} \qquad \lambda \in \boldsymbol{C},$$

Differentiating both sides of (7) with respect to λ, we have that $\frac{f'(\lambda)}{1!}$ and $x_1 = (x_{11}, \cdots, x_{1n})^\top$ satisfy

$$\begin{pmatrix} -1 & & & & \\ 0 & & & & \\ \vdots & & \boxed{B - \lambda I} & & \\ 0 & & & & \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{f'(\lambda)}{1!} \\ x_{11} \\ \vdots \\ x_{1n} \end{pmatrix} = \begin{pmatrix} x_{01} \\ \vdots \\ x_{0n} \\ 0 \end{pmatrix}. \tag{8}$$

Recursively, after $\dfrac{f^{(k-1)}(\lambda)}{(k-1)!}$ and $x_{k-1} = (x_{k-1,1}, \cdots, x_{k-1,n})^\top$ are calculated, the next $\dfrac{f^{(k)}(\lambda)}{k!}$ and $x_k = (x_{k1}, \cdots, x_{kn})^\top$ satisfy

$$\begin{pmatrix} -1 & & & & \\ 0 & & & & \\ \vdots & & \boxed{B - \lambda I} & & \\ 0 & & & & \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{f^{(k)}(\lambda)}{k!} \\ x_{k1} \\ \vdots \\ x_{kn} \end{pmatrix} = \begin{pmatrix} x_{k-1,1} \\ \vdots \\ x_{k-1,n} \\ 0 \end{pmatrix}. \tag{9}$$

The equation (7) can be written as

$$(B - \lambda I)x_0 = f(\lambda)e_1, \quad x_{0n} = 1$$

with $e_1 = (1, 0, \cdots, 0)^\top$. If λ is an eigenvalue of B (or A) and y the associated left eigenvector, then

$$0 = y^H(B - \lambda I)x_0 = \left(y^H e_1\right) f(\lambda) = \overline{y}_1 f(\lambda), \quad \text{where} \;\; y_1 = e_1^H y$$

since B is irreducible upper-Hessenberg, it is easy to verify that $y_1 \neq 0$ and therefore $f(\lambda) = 0$. Thus x_0 is an eigenvector of B associated with λ. Similarly, equation (8) can be written as

$$(B - \lambda I)x_1 = x_0 + f'(\lambda)e_1, \quad x_{1n} = 0,$$

that yields

$$0 = y^H(B - \lambda I)x_1 = y^H x_0 + \overline{y}_1 f'(\lambda).$$

Therefore the condition number of λ

$$\frac{\left|y^H x_0\right|}{\|y\|_2\|x_0\|_2} = \frac{|\overline{y}_1|}{\|y\|_2\|x_0\|_2}\frac{|f'(\lambda)|}{1!}$$

is a multiple of the first derivative of the characteristic polynomial $\det(A - \lambda I)$.

Generally, (9) can be written as

$$(B - \lambda I)x_k = x_{k-1} + \frac{f^{(k)}(\lambda)}{k!}e_1$$

and thus

$$\frac{\left|y^H x_{k-1}\right|}{\|y\|_2\|x_0\|_2} = \frac{|\overline{y}_1|}{\|y\|_2\|x_0\|_2}\frac{\left|f^{(k)}(\lambda)\right|}{k!} \quad k = 1, 2, \cdots, n$$

Based on our analysis in §3, A small $\left|y^H x_0\right|$ does not necessarily mean the eigenvalue is sensitive to perturbations. The numerical m-fold eigenvalue λ of A is well conditioned if

$$\left|y^H x_{m-1}\right| \gg \left|y^H x_{k-1}\right|, \quad k = 1, 2, \cdots, m-1.$$

5 Numerical experiments

In this section, let $J_k(\lambda)$ denote the $k \times k$ Jordan block with eigenvalue λ,

$$J_k(\lambda) = \begin{pmatrix} \lambda & 0 & \cdots & \cdots & 0 \\ 1 & \lambda & 0 & & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & \lambda \end{pmatrix}_{k \times k}$$

and R_m the $m \times m$ random matrix with entries uniformly distributed on $[-1, 1]$. All computations in this section are conducted with Matlab. The condition numbers of eigenvalues are calculated with the Matlab tool `eigsens.m` in [8].

5.1 Example 1: A small matrix

We first test the Newton's iteration on the extended operator (4) with a 10×10 matrix $A = P * diag\,(J_8(2), R_2)\, P^\top$, where P is an orthogonal matrix. The eigenvalues calculated by QR and their condition numbers are

eigenvalues by QR	Condition numbers
0.10174440783530 + 0.33490385222369i	1.89
0.10174440783530 - 0.33490385222369i	1.89
2.01130354558921 + 0.00468662615444i	3045676685097.
2.01130354558921 - 0.00468662615444i	3046164799003.
2.00467135675756 + 0.01130329121656i	3055064826680.
2.00467135675756 - 0.01130329121656i	3055164380992.
1.99531399688166 + 0.01128863222830i	3067021306586.
1.99531399688166 - 0.01128863222830i	3067131909543.
1.98871110077157 + 0.00467199242653i	3074701046616.
1.98871110077157 - 0.00467199242653i	3074634066591.

The bottom eight eigenvalues have condition numbers about 3×10^{12}. They "exploded" from the 8-fold eigenvalue $\lambda = 2$ when calculated as simple ones. The errors are as big as 0.011. However, the Newton's iteration on the extended polynomial operator (4) has no trouble locating the multiple eigenvalue accurately up to 14 digits in 5 iterations. The progress of the iteration is listed below with meaningful digits in λ_i's and the auxiliary variables

$$\begin{array}{rcl} \lambda_0 & = & 2.1 \\ \lambda_1 & = & 2.03 \\ \lambda_2 & = & 2.003 \\ \lambda_3 & = & 2.00003 \\ \lambda_4 & = & 2.000000005 \\ \lambda^* = \lambda_5 & = & 2.00000000000000 \end{array} \qquad \begin{pmatrix} t_1^* \\ t_2^* \\ t_3^* \\ t_4^* \\ t_5^* \\ t_6^* \\ t_7^* \end{pmatrix} = 10^{-15} \begin{pmatrix} 0.3831232126338 \\ 0.5108309501784 \\ 0.5746848189507 \\ -0.3831232126338 \\ -0.1277077375446 \\ -0.7343194908815 \\ -3.1926934386151 \end{pmatrix}.$$

The next test is to perturb matrix A to

$$B = A + 10^{-6} E$$

where E is a random matrix with entries uniformly distributed on $[-1, 1]$. Then "approximate" eigenvalues of B calculated by QR are

```
eigenvalues of B by QR                         condition numbers

0.10174382731821 + 0.33490379864001i               1.887
0.10174382731821 - 0.33490379864001i               1.887
2.15380211678568 + 0.06628105515802i           36054.
2.15380211678568 - 0.06628105515802i           36054.
2.05686339846006 + 0.15411717201049i           40732.
2.05686339846006 - 0.15411717201049i           40732.
1.93336955871027 + 0.14380540950961i           50109.
1.93336955871027 - 0.14380540950961i           50109.
1.85596567090505 + 0.05701606186128i           57472.
1.85596567090505 - 0.05701606186128i           57472.
```

That is, the perturbation of magnitude 10^{-6} on the matrix causes QR's results deviate as large as 0.15. However, by computing them as an 8-fold eigenvalue, the iteration and the auxiliary variables are:

$$\begin{array}{rcl} \lambda_0 & = & 2.1 \\ \lambda_1 & = & 2.03 \\ \lambda_2 & = & 2.003 \\ \lambda_3 & = & 2.00003 \\ \lambda_4 & = & 1.9999999 \\ \lambda_5 & = & 1.999999870 \\ \lambda^* = \lambda_6 & = & 1.99999986978124 \end{array} \qquad \begin{pmatrix} t_1^* \\ t_2^* \\ t_3^* \\ t_4^* \\ t_5^* \\ t_6^* \\ t_7^* \end{pmatrix} = 10^{-6} \begin{pmatrix} 0.4562499978658 \\ 1.4372014231023 \\ 1.7139977872110 \\ -0.4530018548542 \\ -2.1762844787344 \\ 0.1969728713613 \\ 3.3943752282809 \end{pmatrix}.$$

The numerical multiple eigenvalue has a forward error 1.3×10^{-7} as the result of the perturbation E. More importantly, the auxiliary variables t_i^*'s indicate that the eigenvalue $\lambda^* = 1.9999927407264$ as an 8-fold eigenvalue has backward error in the magnitude of 3.4×10^{-6}, which is a close overestimate of the actual magnitude of the backward error.

5.2 Example 2: Defective eigenvalues with large multiplicities

Let $A = diag(J_{50}(2.0), J_{30}(-3.0), R_{20})$. That is, A has a 50-fold defective eigenvalue $\lambda_1 = 2.0$, a 30-fold defective eigenvalue $\lambda_2 = -3.0$, and 20 well-conditioned simple eigenvalues. The eigenvalue distribution calculated by QR is plotted in Fig 1. The two multiple eigenvalues explode under QR into eighty "ill-conditioned" eigenvalues and form two circles with a radius about 0.5.

We first scramble the matrix to $PAP^\top$ with an orthogonal matrix P. Thus λ_1 and λ_2 are in fact very stable numerical multiple eigenvalues. The iterations and magnitudes of auxiliary variables are

$$\begin{aligned}
\lambda_1^{(0)} &= 2.001 \\
\lambda_1^{(1)} &= 2.0002 \\
\lambda_1^{(2)} &= 2.00001 \\
\lambda_1^{(3)} &= 2.00000003 \\
\lambda_1^* = \lambda_1^{(4)} &= 2.0000000000004 \\
\left\|(t_{1,1}^*, \cdots, t_{1,49}^*)^\top\right\|_\infty &= 7.6 \times 10^{-12}
\end{aligned}
\qquad
\begin{aligned}
\lambda_2^{(0)} &= -2.999 \\
\lambda_2^{(1)} &= -3.0004 \\
\lambda_2^{(2)} &= -3.00003 \\
\lambda_2^{(3)} &= -3.0000002 \\
\lambda_2^* = \lambda_2^{(4)} &= -2.999999999997 \\
\left\|(t_{2,1}^*, \cdots, t_{2,29}^*)^\top\right\|_\infty &= 2.7 \times 10^{-11}
\end{aligned}$$

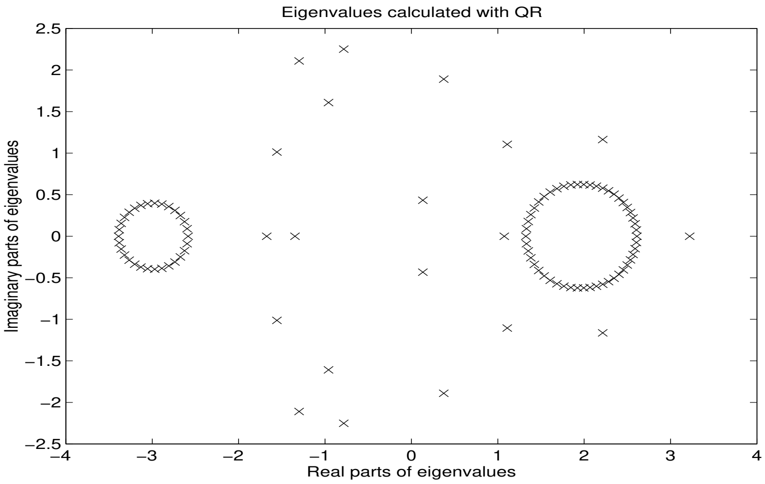

Figure 1: The eigenvalue distribution of a 100×100 matrix calculated by QR

Both eigenvalues are calculated with 13-digit forward accuracy and backward error in the range of 10^{-11}. We then scramble the matrix A to BAB^{-1} using a random matrix $B = R_{100}$ with condition number $cond(B) = 201.8$. Roughly speaking, both the forward and the backward errors are magnified by the condition number.:

$$\begin{aligned}
\lambda_1^{(0)} &= 2.0001 \\
\lambda_1^{(1)} &= 2.000003 \\
\lambda_1^{(2)} &= 2.000000003 \\
\lambda_1^* = \lambda_1^{(3)} &= 2.00000000005306 \\
\left\|(t_{1,1}^*, \cdots, t_{1,49}^*)^\top\right\|_\infty &= 4.0 \times 10^{-9}
\end{aligned}
\qquad
\begin{aligned}
\lambda_2^{(0)} &= -2.999 \\
\lambda_2^{(1)} &= -3.0004 \\
\lambda_2^{(2)} &= -3.00004 \\
\lambda_2^{(3)} &= -3.0000003 \\
\lambda_2^{(4)} &= -2.9999999995 \\
\lambda_2^* = \lambda_2^{(5)} &= -2.999999999553 \\
\left\|(t_{2,1}^*, \cdots, t_{2,29}^*)^\top\right\|_\infty &= 2.9 \times 10^{-8}
\end{aligned} .$$

5.3 Example 3: A derogatory eigenvalue of large multiplicity

Let $A = diag(J_{10}(2.0), J_{20}(2.0), J_{30}(2.0), J_{40}(2.0), R_{20})$. That is, A has a defective and derogatory multiple eigenvalue $\lambda_1 = 2.0$. The matrix A is then scrambled to BAB^{-1} using a random 120×120 matrix $B = R_{120}$. The 120 eigenvalues calculated by QR are plotted in Fig. 2. The 100 "ill-conditioned" QR-eigenvalues are plotted in Fig. 3.

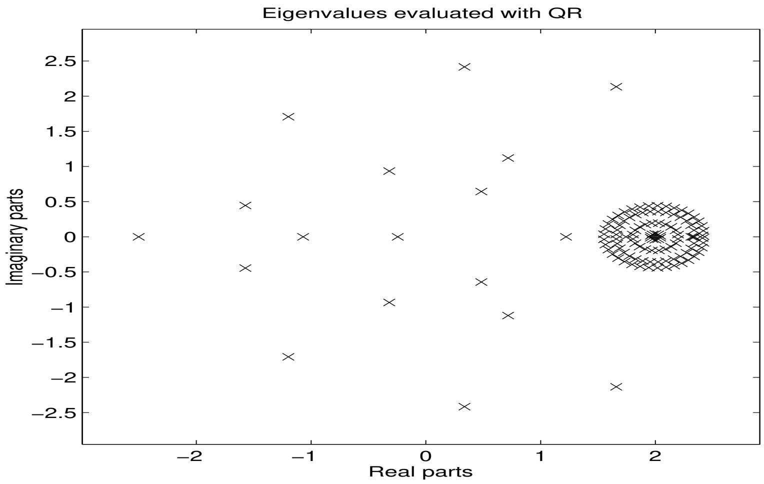

Figure 2: The eigenvalue distribution of a 120×120 matrix calculated by QR

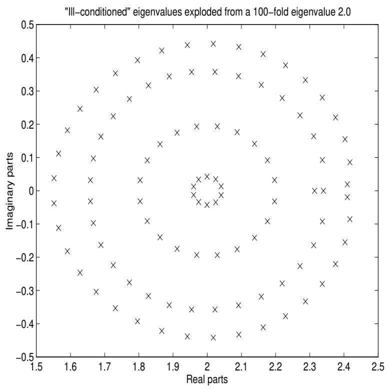

Figure 3: The 100 QR-eigenvalues exploded from the 100-fold eigenvalue $\lambda = 2.0$

The iteration for the numerical multiple eigenvalue and the magnitude of auxiliary variables are:

$$\begin{array}{rcl} \lambda_1^{(0)} & = & 2.0001 \\ \lambda_1^{(1)} & = & 2.000004 \\ \lambda_1^{(2)} & = & 2.000000005 \\ \lambda_1^* = \lambda_1^{(3)} & = & 2.0000000000003 \end{array} \qquad \left\| \begin{pmatrix} t_1^* \\ \vdots \\ t_{99}^* \end{pmatrix} \right\|_\infty = 2.0 \times 10^{-10}.$$

We can further confirm this accuracy by calculating the the singular values of the matrix $BAB^{-1} - 2.0000000000003 I_{120}$. The smallest 6 singular values are

$$\begin{array}{l} \cdots \ \cdots \\ 0.03243152758926348 \\ 0.01804602808483129 \\ 0.00000000000000434 \\ 0.00000000000000125 \\ 0.00000000000000051 \\ 0.00000000000000021. \end{array}$$

The singular values clearly indicate that $\lambda_1 = 2.0000000000003$ is a derogatory eigenvalue corresponding to 4 Jordan blocks.

5.4 Example 4: Computing Jordan a canonical form

The following matrix is a classical example for testing computations of Jordan canonical form [10]

$$A = \begin{pmatrix} 1 & 1 & 1 & -2 & 1 & -1 & 2 & -2 & 4 & -3 \\ -1 & 2 & 3 & -4 & 2 & -2 & 4 & -4 & 8 & -6 \\ -1 & 0 & 5 & -5 & 3 & -3 & 6 & -6 & 12 & -9 \\ -1 & 0 & 3 & -4 & 4 & -4 & 8 & -8 & 16 & -12 \\ -1 & 0 & 3 & -6 & 5 & -4 & 10 & -10 & 20 & -15 \\ -1 & 0 & 3 & -6 & 2 & -2 & 12 & -12 & 24 & -18 \\ -1 & 0 & 3 & -6 & 2 & -5 & 15 & -13 & 28 & -21 \\ -1 & 0 & 3 & -6 & 2 & -5 & 12 & -11 & 32 & -24 \\ -1 & 0 & 3 & -6 & 2 & -5 & 12 & -14 & 37 & -26 \\ -1 & 0 & 3 & -6 & 2 & -5 & 12 & -14 & 36 & -25 \end{pmatrix}. \tag{10}$$

The Hessenberg reduction results in a numerically reducible upper-Hessenberg matrix with diagonal blocks H_1 of 2×2, H_2 of 4×4 and H_3 of 4×4. Their approximate eigenvalues calculated by QR and condition numbers are,

```
 Eigenvalues by QR                             Condition numbers

1.99999996667999                                       5001999.
2.00000003332001                                       5001999.

3.00000000000000 + 0.00000001937343i                  13650592.
3.00000000000000 - 0.00000001937343i                  13650592.
2.00000000000000 + 0.00000004969745i                   1058412.
2.00000000000000 - 0.00000004969745i                   1058412.

3.00000010568030                                       1161073.
2.99999989431971                                       1161073.
2.00000000000000                                           1.47
1.00000000000000                                           1.26
```

Checking the derivatives f' and f'' of the characteristic polynomial f at all eight QR-eigenvalues above with large condition numbers, we observed a gap of 10^7 in magnitudes, indicating they are all numerical double eigenvalues of the corresponding diagonal blocks. Using those QR-eigenvalues as starting iterates with multiplicity 2, we recalculate the eigenvalues with their backward errors, grouped as eigenvalues of H_1, H_2 and H_3 respectively:

```
      Refined eigenvalues                         Backward errors
(repeated according to multiplicity)

      2.00000000000000                               9.99e-016
      2.00000000000000                               9.99e-016

      3.00000000000000 + 0.00000000000000i           1.44e-015
      3.00000000000000 - 0.00000000000000i           1.44e-015
      2.00000000000000 + 0.00000000000000i           7.47e-015
      2.00000000000000 - 0.00000000000000i           7.47e-015

      3.00000000000001                               2.61e-014
      3.00000000000001                               2.61e-014
      2.00000000000000                               3.12e-015
      1.00000000000000                               6.88e-015.
```

Therefore, it is clear that the matrix A in (10) has a numerical 5-fold eigenvalue 2.0, a numerical 4-fold eigenvalue 3.0, and a simple eigenvalue 1.0. The Jordan structure of A can be determined by the singular values of $(A - \lambda I)^k$'s:

Singular values of $A-2I$	Singular values of $(A-2I)^2$	Singular values of $A-3I$	Singular values of $(A-3I)^2$
$\cdots\ \cdots$	$\cdots\ \cdots$	$\cdots\ \cdots$	$\cdots\ \cdots$
0.237920273240	0.401358847554	0.441640607758	0.581708725435
0.096366416213	0.067192074052	0.153907600163	0.072422782669
0.000000000000	0.000000000000	0.000000000000	0.000000000000
0.000000000000	0.000000000000	0.000000000000	0.000000000000
	0.000000000000		0.000000000000
	0.000000000000		0.000000000000

Thus, eigenvalues 2 and 3 are both from two Jordan blocks of sizes at least 2. Therefore, A has the Jordan canonical form

$$J = diag\,(J_3(2.0), J_2(2.0), J_2(3.0), J_2(3.0), J_1(1.0))$$

Even if the matrix A is perturbed with a random matrix E with $\|E\|_2 \approx 10^{-8}$, the above process still determines the numerical Jordan canonical structure easily.

5.5 Example 5: Ill-conditioned numerical multiple eigenvalues

The test matrix is known as the Frank matrix [8]. Its entries are

$$A = (a_{ij}) \quad \text{where} \quad a_{ij} = \begin{cases} n+1-\max\{i,j\}, & j \geq i-1 \\ 0, & j < i-1 \end{cases}.$$

Eigenvalues of the Frank matrix with small magnitudes are ill-conditioned. For example, for a 12×12 Frank matrix A, the eigenvalues and corresponding condition numbers are

```
QR-eigenvalues          Condition numbers

32.22889150157214              3.29
20.19898864587707              4.98
12.31107740086851              3.14
 6.96153308556711              1.71
 3.51185594858070              6.92
 1.55398870913212            216.
 0.64350531891624          14467.
 0.28474972647719         560310.
```

0.14364644045883	6701452.
0.08122798609205	26645187.
0.04950694629965	38773986.
0.03102829015834	18284203..

The smallest five or six eigenvalues can be considered ill-conditioned in traditional sense. We apply Newton's iteration on the extended polynomial operator (4) with initial iterate $\lambda = 0$ and various attempted multiplicity m. The following table lists obtained numerical multiple eigenvalue corresponding to different attempted multiplicity. The backward error suggested that none of the result is backward accurate:

attempted multiplicity	numerical multiple eigenvalue	backward errors
m = 2	0.03861651128815	0.000076
m = 3	0.05031683035298	0.0052
m = 4	0.06988614253113	0.0029
m = 5	0.10596312822852	0.016
m = 6	0.17866718710829	0.097.

It is clear that those eigenvalues are inherently ill-conditioned, either under traditional definition or as numerical multiple eigenvalues.

6 Concluding remarks

In this paper we discussed the computations of numerical multiple eigenvalues used to be considered ill-conditioned. They can be calculated accurately in forward sense, provided that the multiplicity is known and a good initial iterate is given. For eigenvalues with low multiplicity, eigenvalues calculated with any traditional method such as QR can be used as initial iterates. The multiplicity can be obtained by observing the magnitude gaps between adjacent derivatives $\frac{f(\lambda)}{0!}, \frac{f'(\lambda)}{1!}, \frac{f''(\lambda)}{2!} \cdots$. There are also other methods that can be used to estimate the multiplicity of a multiple root [11, 20]. If the multiplicity is high, eigenvalues calculated by QR could be too inaccurate to be used as initial iterates. The multiplicity could be difficult to determine. On the other hand, if a group of QR-eigenvalues exploded from a numerical multiple eigenvalues, computational experience shows that the mean of the eigenvalue group is usually very good approximate of the numerical multiple eigenvalue and should be used as the initial iterate. Actually, this strategy is often used in numerical computation of Jordan canonical forms [10].

The concept of numerical multiple eigenvalue can be traced back to earlier literatures such as [10, 16, 21], where the stability of numerical multiple eigenvalues has been suggested. It has long been known that the mean of perturbed eigenvalues (or polynomial roots) corresponding to the multiple one is stable [3, 22]. The analysis and numerical results in this paper further confirm that, among the traditionally defined ill-conditioned eigenvalues, some are indeed well-conditioned and can be computed with remarkable forward accuracy. Moreover, as a by-product of the computation, the backward errors can be estimated via the auxiliary variables. If the data matrix warrants such stability and accuracy, there seems little reason to settle on the results with huge forward errors.

Acknowledgments The author wishes to thank Jim Demmel for a number of fruitful discussions on the subject and related issues.

References

[1] T. Beelen and P. Van Dooren, *Computational aspects of the Jordan canonical form*, in Reliable Numerical Computation, M.G. Cox and S. Hammerling, eds., Clarendon Press, Oxford, 1990, pp57-72.

[2] F. Chatelin, *Ill conditioned eigenproblems*, in Large Scale Eigenvalue Problems, J. Cullum and R. A. Willoughby, eds, Elsevier Science Publishers B. V. North-Holland, Amsterdam, 1986.

[3] F. Chaitin-Chatelin and V. Frayssé, *Lectures on Finite Precision Computations*, SIAM, Philadelphia, 1996.

[4] J. G. F. Francis, The QR transformation, Parts I and II, Computer J. Vol. 4 (1961, 1962), pp 265–271, pp 332–345.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore and London, 1996.

[6] G. H. Golub and J. H. Wilkinson, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Review, Vol. 18 (1976), pp578-619.

[7] M. A. Hyman, *Eigenvalues and eigenvectors of general matrices*, Twelfth National Meeting A.C.M., Houston, Texas (1957).

[8] N. J. Higham, *The test matrix toolbox for Matlab*, Numerical Analysis Report No. 237, Department of Mathematics, University of Manchester (1993).

[9] M. Igarashi and T. Ypma, *Relationships between order and efficiency of a class of methods for multiple zeros of polynomials*, J. Comput. Appl. Math., Vol 60 (1995), pp101–113.

[10] B. Kågström and A. Ruhe, *An algorithm for numerical computation of the Jordan normal form of a complex matrix*, ACM Trans. Math. Software, Vol. 6 (1980), pp 398-419.

[11] T. Y. Li and Z. Zeng, *The homotopy continuation algorithm for the real nonsymmetric eigenproblem, further development and implementation*, to appear: SIAM J. Sci. Comput.

[12] J. McNamee, *A bibliography on roots of polynomials*, J. Comput. Appl. Math., Vol. 47 (1993), pp 391–394.

[13] T. Miyakoda, *Iterative methods for multiple zeros of a polynomial by clustering*, J. Comput. Appl. Math., Vol. 28 (1989), pp 315–326.

[14] H. M. Möller and H. J. Stetter, *Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems*, Numer. Math., Vol. 70 (1995), pp 311-329.

[15] A. Ruhe, *Properties of a matrix with a very ill-conditioned eigenproblem*, Numer. Math., Vol. 15 (1970), pp 57–60.

[16] A. Ruhe, *An algorithm for numerical determination of the structure of a general matrix*, BIT, Vol. 10 (1970), pp196-216.

[17] J. A. Stolan, *An improved Šiljak's algorithm for solving polynomial equations converges qadratically to multiple zeros*, J. Comput. Appl. Math., Vol 64 (1995), pp 247–268.

[18] M. V. Straeten and H. Van de Vel, *Multiple root-finding methods*, J. Comput. Appl. Math., Vol. 40 (1992), pp 105-114.

[19] L. N. Trefethen, *Maxims about numerical mathematics, computers, science and life*, SIAM News, Vol. 31, No. 1, p 4, 1998.

[20] C. W. Ueberhuber, *Numerical Computation 2*, Springer-Verlag, Berlin, Heidelberg, New York, 1997.

[21] J.M. Varah,*Computing invariant subspaces of a general matrix when the eigensystem is poorly conditioned*, Math. Comp., Vol. 24 (1970), pp137-149.

[22] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J. , 1963.

[23] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965.

[24] J. H. Wilkinson, *Note on matrices with a very ill-conditioned eigenproblem*, Numer. Math. Vol. 19, pp 176–178, 1972.

[25] T. J. Ypma, *Finding a multiple zero by transformations and Newton-like methods*, SIAM Review, Vol. 25 (1983), pp 365–378.

Department of Mathematics, Northeastern Illinois University, Chicago, IL 60626

E-mail address: `zzeng@neiu.edu`, `http://www.neiu.edu/~zzeng`