A primal-dual Jacobi–Davidson-like method for nonlinear eigenvalue problems

Hubert Schwetlick and Kathrin Schreiber

ZIH-IR-0613

Oktober 2006

corrected version 02.03.07

Submitted to ETNA, the Electronic Transactions on Numerical Analysis, a special volume on Applied Linear Algebra
A PRIMAL-DUAL JACOBI–DAVIDSON-LIKE METHOD FOR NONLINEAR EIGENVALUE PROBLEMS

HUBERT SCHWETLICK†‡ AND KATHRIN SCHREIBER†§¶

Abstract. We propose a method for the solution of non-Hermitian ill-conditioned nonlinear eigenvalue problems \( T(\lambda)x = 0 \), which is based on singularity theory of nonlinear equations. The singularity of \( T(\lambda) \) is characterized by a scalar condition \( \mu(\lambda) = 0 \) where the singularity function \( \mu \) is implicitly defined by a nonsingular linear system with the appropriately bordered matrix \( T(\lambda) \). One Newton step for \( \mu(\lambda) = 0 \) is performed which leads to a generalized Rayleigh quotient as new eigenvalue approximation, and the bordering vectors are updated, too. Quadratic convergence is appropriately bordered matrix \((ZIH)\), TU Dresden, thanks to Prof. W. Nagel.

Key words. nonlinear eigenvalue problem, generalized RQI, Jacobi–Davidson, Newton’s method

AMS subject classifications. 65F15, 65H10, 65H17

1. Introduction. We consider the nonlinear eigenvalue problem

\[
T(\lambda)x = 0
\]

(1.1)

where \( T(\cdot) : \lambda \in \mathbb{C} \mapsto T(\lambda) \in \mathbb{C}^{n \times n} \) is a matrix-valued mapping. If equation (1.1) has a nontrivial solution \( (x_*, \lambda_*) \), then \( \lambda_* \) is called an eigenvalue and \( x_* \) a right eigenvector. We speak of \( y_* \neq 0 \) as a left eigenvector with respect to \( \lambda_* \) if \( y_*^H T(\lambda_*) = 0 \), i.e., \( T(\lambda_*)^H y_* = 0 \).

A state of the art survey on nonlinear eigenvalue problems, their occurrence and origins as well as available methods is given in [8].

The Newton step \( (u, \lambda) \mapsto (u_+, \lambda_+) \) for the system (1.1) extended by the normalization equation \( w^H T(\lambda) x = 1 \) with \( 0 \neq w \in \mathbb{C}^n \) is given by the linear system

\[
\begin{bmatrix}
T(\lambda) & \dot{T}(\lambda)u \\
w^H & 0
\end{bmatrix}
\begin{bmatrix}
u_+ - u \\
\lambda_+ - \lambda
\end{bmatrix}
= -
\begin{bmatrix}
T(\lambda)u \\
w^H u - 1
\end{bmatrix}
\iff
T(\lambda)u_+ + (\lambda_+ - \lambda)\dot{T}(\lambda)u = 0
w^H u_+ = 1,
\]

(1.2)

see [3] and [10]. The upper block is equivalent to

\[
u_+ = (\lambda_+ - \lambda)T(\lambda)^{-1}\dot{T}(\lambda)u
\]

if \( \lambda \) is not an eigenvalue of \( T \), which is usually considered as one step of Inverse iteration. Moreover, if \( u^H u = 1 \) and \( w = u \) is taken, system (1.2) yields the projected linear system

\[
\left( I - \frac{T(\lambda) uu^H}{u^H T(\lambda) u} \right) T(\lambda) (I - uu^H) s = - \left( I - \frac{\dot{T}(\lambda) uu^H}{u^H \dot{T}(\lambda) u} \right) T(\lambda) u, \quad u^H s = 0
\]

(1.3)

for the correction \( s \equiv u_+ - u \). The choice of \( \lambda \) such that \( u^H T(\lambda) u = 0 \), cf. [10], simplifies the right hand side to \( -T(\lambda) u \), leading to the correction equation in the nonlinear Jacobi–Davidson method [5].

---

†Hubert.Schwetlick@tu-dresden.de
‡Kathrin.Schreiber@mailbox.tu-dresden.de
§Technische Universität Dresden, Fakultät Mathematik und Naturwissenschaften, Institut für Numerische Mathematik, 01062 Dresden, Germany
¶The research of this author was partially supported by Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH), TU Dresden, thanks to Prof. W. Nagel
In the present paper we want to extend the singularity theory based algorithm given in [11] and reformulated in [12], namely our algorithm GRQI which was designed for highly nonnormal linear problems \( T(\lambda) = A - \lambda I \), to nonlinear eigenvalue problems (1.1).

In Section 2, we derive the new algorithm. There the singularity of \( T(\lambda) \) is characterized by the so-called singularity system

\[
\begin{bmatrix}
T(\lambda) & v \\
u^H & 0
\end{bmatrix}
\begin{bmatrix}
x \\
\mu
\end{bmatrix}
= \begin{bmatrix}
0 \\
1
\end{bmatrix}
\iff
T(\lambda)x + v\mu = 0
\text{ \hspace{1cm} } u^Hx = 1
\tag{1.4}
\]

which uses the bordered matrix

\[
C(\lambda, u, v) = \begin{bmatrix}
T(\lambda) & v \\
u^H & 0
\end{bmatrix}, \quad \|u\| = \|v\| = 1.
\tag{1.5}
\]

The norm is always the Euclidean vector norm or the spectral norm of a matrix. Obviously, if \( \lambda_* \) is simple, the matrix \( C(\lambda_*, u, v) \) is nonsingular if and only if \( u^Hx_* \neq 0 \) and \( v^Hy_* \neq 0 \), cf. §3. Moreover, at \( \lambda = \lambda_* \), the borderings \( u = x_* \) and \( v = y_* \) are optimal in that they minimize the norm of the inverse whereas the right bordering \( v = \hat{T}(\lambda_*)x_* \), prescribed by the Newton system (1.2), may lead to an arbitrarily large inverse if \( v^H\hat{T}(\lambda_*)x_* \) is small, cf. §3. This situation can occur if \( \hat{T}(\lambda_*) \) is strongly nonnormal.

Hence, if \( u, v \) are appropriately chosen and \( \lambda \) is sufficiently close to \( \lambda_* \), equation (1.4) is uniquely solvable and defines \( x \equiv x(\lambda) \), \( \mu \equiv \mu(\lambda) \) as functions of \( \lambda \). For \( \lambda = \lambda_* \), system (1.4) is solved by \( x(\lambda_*) = x_*/u^Hx_* \), \( \mu(\lambda_*) = 0 \), and for \( \lambda \) close to a geometrically simple eigenvalue \( \lambda_* \) one has \( \mu(\lambda) = 0 \) iff rank \( T(\lambda) = n - 1 \).

So we try to determine \( \lambda \) such that the scalar equation \( \mu(\lambda) = 0 \) is satisfied by applying Newton’s method. It turns out that, for computing the derivative \( \hat{\mu}(\lambda) \), the system

\[
\begin{bmatrix}
T(\lambda)^H & u \\
v^H & 0
\end{bmatrix}
\begin{bmatrix}
y \\
\bar{\nu}
\end{bmatrix}
= \begin{bmatrix}
0 \\
1
\end{bmatrix}
\iff
T(\lambda)^Hy + u\bar{\nu} = 0
\text{ \hspace{1cm} } v^Hy = 1
\tag{1.6}
\]

which is dual to (1.4) has to be solved, too. Here \( \bar{\nu} \) denotes the conjugate complex number to \( \nu \). Note, that we always have \( \mu = \nu \). Moreover, the borderings \( u, v \) will be replaced by improved ones, namely the normalized vectors \( x, y \) from (1.4), (1.6), resp.

Section 3 is pure convergence analysis, quadratic convergence is shown for eigenvalue and eigenvectors simultaneously, provided that \( \lambda_* \) is a simple eigenvalue.

Reformulation of this 2-vector algorithm in §4 gives a Jacobi–Davidson-like formulation called GJD with the primal

\[
(I - vv^H)T(\lambda)(I - uu^H)s = -T(\lambda)u, \quad u^Hs = 0
\]

and the dual

\[
(I - uu^H)T(\lambda)^H(I - vv^H)t = -T(\lambda)^Hv, \quad v^Ht = 0
\]

correction equation where \( u, v \) are the current normalized approximations for the right and left eigenvector, respectively, and \( \lambda \) is chosen such that \( v^HT(\lambda)u = 0 \). Notice the difference to the standard Jacobi–Davidson correction equation (1.3) where the
derivative $\dot{T}(\lambda)$ is involved. Thus JD works with a skew left projector so the symmetry of Hermitian problems is lost in contrary to the equations above when $v \equiv u$.

Section 5 provides several numerical results. The method for linear problems is especially suited for highly nonnormal matrices, where left and right eigenvector are almost orthogonal. We expect this property to hold also for nonlinear problems. That means we focus on non-Hermitian problems, where the angle between the eigenvectors corresponding to the target eigenvalue is close to $\pi/2$.

After a short view on the single and two-vector methods we compare GJD with other Jacobi–Davidson methods, as there are the alternating and two-sided versions of Hochstenbach [6], additionally to the classic one. Results here are much more distinguishing than those of the one-, respectively, two-vector iterations. GJD clearly shows better performance than the others do.

We will use $(\lambda, x^*, y^*)$ when speaking of an exact eigentriplet and $(\lambda, u, v)$ for some approximation.

2. Motivation of the algorithm. Our former algorithm GRQI, see [11], [12], emerged from bifurcation theory for parameter dependent nonlinear equations. We want to motivate the new one in an equivalent but shorter, more direct way; cf. [14] for the Hermitian case.

Consider system (1.4) where $u$ and $v$ are approximations to $x^*$ and $y^*$, resp. The nonsingularity of the matrix $C(\lambda, u, v)$ is proved in Section 3, provided that $\lambda^*$ is simple, $x^H u \neq 0$, $y^H v \neq 0$, and $\lambda$ is close to $\lambda^*$.

Now we perform one Newton step for $\mu(\lambda) = 0$ as solution of (1.4) from the current eigenvalue approximation $\lambda$ using the current eigenvector approximations $u$, $v$ as borderings.

From the definition of $\mu$ by (1.4) we get
\[
\mu(\lambda) = e_{n+1}^T C(\lambda, u, v)^{-1} e_{n+1} = e_{n+1}^T C(\lambda)^{-1} e_{n+1}
\]
where $C(\lambda) = C(\lambda, u, v)$. An alternative representation is delivered by applying Cramer’s rule to (1.4), namely
\[
\mu = \mu(\lambda) = e_{n+1}^T C(\lambda)^{-1} e_{n+1} = \begin{vmatrix} T(\lambda) & 0 \\ u^H & 1 \end{vmatrix} \begin{vmatrix} T(\lambda) & 0 \\ v & u^H \end{vmatrix} = \frac{\det T(\lambda)}{\det C(\lambda)} =: \frac{p(\lambda)}{\gamma(\lambda)}.
\]

Hence, $\mu(\lambda)$ is a scaled value of $p(\lambda) := \det T(\lambda)$; cf. [14] again for the Hermitian case, where the method therefore is called implicit determinant method. Finally, by multiplying the upper block of (1.4) with $y^H$ and considering the second equation of (1.6), we obtain the third representation
\[
\mu(\lambda) = -y^H(\lambda)x(\lambda).
\]

Differentiation of (2.1) yields
\[
\dot{\mu}(\lambda) = -e_{n+1}^T C(\lambda)^{-1} \dot{C}(\lambda) C(\lambda)^{-1} e_{n+1} = - \begin{bmatrix} y^H \nu \\ \dot{T}(\lambda) \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = -y^H \dot{T}(\lambda)x
\]
where $(y^H, \nu)^H = e_{n+1}^T C(\lambda)^{-1}$ is defined by the dual system (1.6).
Thus, using (2.3), the Newton step $\lambda \mapsto \lambda_+$ is given by

$$
\lambda_+ = \lambda - \frac{\mu(\lambda)}{\mu(\lambda)} = \lambda - \frac{y^HT(\lambda)x}{y^HT(\lambda)x}.
$$

(2.5)

In [7] and [16] this expression is called *generalized Rayleigh quotient*. It is equivalent to one Newton step $\lambda \mapsto \lambda_+$ for the scalar equation $g_+(\lambda) = y^HT(\lambda)x = 0$. Alternatively, $\lambda_+$ may be defined as solution of this equation, cf. [10]. Let us mention that, in case of the generalized linear eigenvalue problem $T(\lambda) = A - \lambda B$, both rules deliver the generalized Rayleigh quotient $\lambda_+ = \lambda_+(x, y) = y^HAx/y^HBx$ independent of $\lambda$.

Since $x(\lambda_*)$ and $y(\lambda_*)$ are multiples of $x_*$ and $y_*$, resp., cf. §3, we expect $x(\lambda)$ and $y(\lambda)$ to be better approximations then $u$ and $v$ and, therefore, update $u$, $v$ by the normalized vectors $x(\lambda)$ and $y(\lambda)$ as in [11], [12], cf. [1].

For computational purposes it is more convenient to work with the correction form of the vector updates, i.e., instead of $x$ we are interested in the correction $s \equiv x - u$. Replacing $x = u + s$ in (1.4) yields the residual formulation

$$
\begin{bmatrix}
  T(\lambda) & v \\
  u^H & 0
\end{bmatrix}
\begin{bmatrix}
  s \\
  \mu
\end{bmatrix}
= -
\begin{bmatrix}
  T(\lambda)u \\
  0
\end{bmatrix}
\iff T(\lambda)s + v\mu = -T(\lambda)u =: -ru
$$

s.t. $u^Hs = 0$.

(2.6)

The dual system is handled in the same way setting $y = v + t$. Hence, the final algorithm looks as follows

**Algorithm 1. GRQI**

<table>
<thead>
<tr>
<th>Input: $(\lambda_0, u_0, v_0)$ where $u_0^Hu_0 = v_0^Hv_0 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>for $i = 0, 1, 2, \ldots$</td>
</tr>
</tbody>
</table>
| S1: Set $C_i = C_i(\lambda_i, u_i, v_i) = \begin{bmatrix}
  T(\lambda_i) \\
  u_i^H \\
  0
\end{bmatrix}$ |
| S2: Solve $C_i \begin{bmatrix}
  s_i \\
  \mu_i
\end{bmatrix} = -\begin{bmatrix}
  T(\lambda_i)u_i \\
  0
\end{bmatrix}$, set $u_{i+1} = \frac{u_i + s_i}{\|u_i + s_i\|}$ |
| S3: Solve $C_i^H \begin{bmatrix}
  t_i \\
  \nu_i
\end{bmatrix} = -\begin{bmatrix}
  T(\lambda_i)^Hv_i \\
  0
\end{bmatrix}$, set $v_{i+1} = \frac{v_i + t_i}{\|v_i + t_i\|}$ |
| S4: Compute $\lambda_{i+1} = \lambda_i - \frac{\mu_i}{(v_i + t_i)^H\tilde{T}(\lambda_i)(u_i + s_i)}$ |

Several similar methods with bordered matrices can be found in [2], as well as a remark, that the condition number of the bordered matrix in the limit $\lambda = \lambda_*$ is minimal for borderings with $x_*, y_*$ as in our case.

If $T(\lambda)$ is Hermitian, i.e., $T(\lambda) = T(\lambda)^H \forall \lambda$, then we have $y_* = x_*$ so we can use symmetric borderings $v = u$. This implies $t_i = s_i$, i.e., we only have to solve the system in step S2, and the matrix $C_i = C(\lambda_i, u_i, u_i)$ is Hermitian. For this special case algorithm GRQI has been introduced by SPENCE/POLTON [14]. Since $C_i$ is Hermitian (but in general indefinite), Krylov solvers designed for Hermitian systems as, e.g., MINRES can be used whereas the standard Newton update (1.2) with $w = u_i$ which is equivalent to the standard Jacobi–Davidson update leads to $C_i = C(\lambda_i, \tilde{T}(\lambda_i)u_i, u_i)$ which is not Hermitian, in general.
3. Convergence analysis. Before we analyze the proposed algorithm, some basic facts will be introduced.

We begin with a characterization of simple eigenvalues. A necessary condition for the nonsingularity of \( C \) is that its upper block \([T \; v]\) has full rank \( n \). So our approach requires rank \( T(\lambda_s) = n - 1 \), i.e., \( \lambda_s \) be geometrically simple. The question when \( \lambda_s \) is also algebraically simple is answered by the following Lemma.

**Lemma 1.** Let \( \lambda_s \) be a geometrically simple eigenvalue of (1.1), i.e., \( \det T(\lambda_s) = 0 \) and \( \dim \ker T(\lambda_s) = 1 \), \( \ker T(\lambda_s) = \text{span} \{ x_s \} \), \( \ker T(\lambda_s)^H = \text{span} \{ y_s \} \), \( \| x_s \| = \| y_s \| = 1 \), and let \( p(\lambda) := \det T(\lambda) \). Then there holds

\[
\dot{p}(\lambda_s) = \gamma_s \cdot y_s^H \dot{T}(\lambda_s)x_s \quad \text{with} \quad \gamma_s := \det C(\lambda_s, x_s, y_s) \neq 0,
\]  

(i.e., we have

\[
\dot{p}(\lambda_s) \neq 0, \quad \text{i.e., } \lambda_s \text{ is algebraically simple} \iff y_s^H \dot{T}(\lambda_s)x_s \neq 0.
\]

**Proof.** Let \( C_0(\lambda) := C(\lambda, x_s, y_s) \), and consider the system \( C_0(\lambda) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \), i.e., (1.4) with \( u = x_s, \; v = y_s \). Now set \( \lambda = \lambda_s \). Multiplying the upper block by \( y_s^H \) yields

\[
y_s^H T(\lambda_s)x + y_s^H y_s \mu = 0, \quad \text{hence, } \mu = \mu(\lambda_s) = 0.
\]

Using this, the upper block now reads as \( T(\lambda_s)x = 0 \), hence, \( x = x_s \alpha \), and the normalization condition \( x_s^H x = x_s^H x_s \alpha = 1 \) gives \( x = x(\lambda_s) = x_s \). Therefore, the system is uniquely solvable with \( x = x_s, \; \mu = \mu_s = 0 \), and \( C_0(\lambda_s) \) is nonsingular. This implies unique solvability also for \( \lambda \) close to \( \lambda_s \). From (2.2) with \( u = x_s, \; v = y_s \) we obtain \( p(\lambda) = \mu(\lambda)\gamma_0(\lambda) \) with

\[
\gamma_0(\lambda) = \det C_0(\lambda) \quad \text{and} \quad \dot{p}(\lambda) = \mu(\lambda_s)\gamma_0(\lambda_s) + \mu(\lambda_s)\gamma_0(\lambda_s) = \dot{\mu}(\lambda_s)\gamma_0(\lambda_s).
\]

For the derivative we get from (2.4) with \( u = x_s, \; v = y_s \)

\[
\dot{\mu}(\lambda_s) = -\epsilon_{n+1}^T C_0(\lambda_s)^{-1} \dot{C}_0(\lambda_s) C_0(\lambda_s)^{-1} \epsilon_{n+1} = -\begin{bmatrix} y_s \\ 0 \end{bmatrix}^H \begin{bmatrix} \dot{T}(\lambda_s) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ 0 \end{bmatrix}
\]

which proves (3.1). Here we have used that the left factor equals

\[
C_0(\lambda_s)^{-H} \epsilon_{n+1} = \begin{bmatrix} y_s \\ \bar{\nu} \end{bmatrix} = \begin{bmatrix} y(\lambda_s) \\ \bar{\nu}(\lambda_s) \end{bmatrix} = \begin{bmatrix} y_s \\ 0 \end{bmatrix}.
\]

Recall that Neumaier [9] proved the equivalence

\[
\dot{p}(\lambda_s) \neq 0 \iff \text{corank } T(\lambda_s) = 1, \; y_s^H \dot{T}(\lambda_s)x_s \neq 0
\]

using some results from determinant theory, but we preferred to give a short direct proof in the context of our approach.

Next we will summarize assumptions we need for the following convergence analysis.

**Definition 2.** Let \( D \subset \mathbb{C} \) be an open set. We say that \( T : D \to \mathbb{C}^{n \times n} \) applies to Assumption (A), if the following holds:

(i) There exists \( \lambda_s \in D \) and \( \tau_s > 0 \) such that

\[
S_s := \mathbb{S}(\lambda_s, \tau_s) = \{ \lambda \in \mathbb{C} : |\lambda - \lambda_s| \leq \tau_s \} \subset \text{int } D
\]

(ii) \( T \) is complex differentiable on an open neighborhood \( \tilde{S}_s \) of \( S_s \), hence, \( T \) is holomorphic on \( \tilde{S}_s \) and there exists a constant \( L_1 > 0 \) such that

\[
\| \dot{T}(\lambda) - \dot{T}(\mu) \| \leq L_1 |\lambda - \mu| \quad \forall \lambda, \mu \in \tilde{S}_s
\]

(3.3)
(iii) \( \lambda_* \) is an algebraically simple eigenvalue of \( T \), i.e., rank \( T(\lambda_*) = n - 1 \), that means there exist \( x_*, y_* \in \mathbb{C}^n, \|x_*\| = \|y_*\| = 1 \), such that

\[
\ker T(\lambda_*) = \text{span} \{x_*\}, \quad \ker T(\lambda_*)^H = \text{span} \{y_*\}, \quad \text{and } \alpha_* := y_*^H T(\lambda_*) x_* \neq 0.
\]

Remark. If \( D \subset \mathbb{R} \) and \( T : D \to \mathbb{R}^{n \times n} \) is a real matrix-valued function then assumptions in (ii) change to requiring \( T \) to be (real) differentiable on \( S_* \) and \( \hat{T} \) to be Lipschitz continuous on \( S_* \) with constant \( L_1 > 0 \).

Let (A) hold in the following sections. This implies boundedness of \( T \) and \( \hat{T} \) and Lipschitz continuity of \( T \) on \( S_* \), since for all \( \lambda, \mu \in S_* \)

\[
\|\hat{T}(\lambda)\| \leq \|\hat{T}(\lambda_*)\| + L_1 \tau_* =: M_1
\]

\[
\|T(\lambda) - T(\mu)\| = \|\int_0^1 \frac{\hat{T}(\lambda + t(\mu - \lambda))}{\mu - \lambda} \, dt\| \leq M_1 |\lambda - \mu|
\]

\[
\|T(\lambda)\| \leq \|T(\lambda_*)\| + M_1 \tau_* =: M_0.
\]

Naturally, the last inequality implies the first one for holomorphic functions as well as boundedness of all derivatives of \( T \) on \( S_* \).

Now, for \( \lambda \in D \) and approximations \( u, v \in \mathbb{C}^n, \|u\| = \|v\| = 1 \), we consider the bordered matrix \( C = C(\lambda) = C(\lambda, u, v) \), as introduced in (1.5). The next Lemma states the nonsingularity of \( C \) and characterizes the norm of its inverse in dependence on the bordering vectors \( u, v \).

**Lemma 3.** Assume (A), and define

\[
C_* := C(\lambda_*, u, v) = \begin{bmatrix} T(\lambda_*) & v \\ u^H & 0 \end{bmatrix}.
\]

Then

(i) \( C_* \) is nonsingular, if and only if

\[
\cos \xi = |u_2| = |x_*^H u| > 0, \quad \cos \eta = |v_2| = |y_*^H v| > 0 \tag{3.4}
\]

where \( \xi = \angle (\text{span} \{u\}, \text{span} \{x_*\}) \), \( \eta = \angle (\text{span} \{v\}, \text{span} \{y_*\}) \).

(ii) Let (3.4) hold. Then

\[
\|C_*^{-1}\| \leq \sqrt{\frac{\|T(\lambda_*)^H\|^2 + 2}{\cos \eta \cos \xi}} =: N(\xi, \eta) \tag{3.5}
\]

where \( M^1 \) denotes the Moore–Penrose Pseudo–Inverse of \( M \).

(iii) Assume (3.4) and let

\[
|\lambda - \lambda_*| \leq \tau(\xi, \eta) := \min \left\{ \tau_*, \frac{1}{2M_1 N(\xi, \eta)} \right\}.
\]

Then \( C \) is nonsingular and

\[
\|C^{-1}\| \leq \frac{N(\xi, \eta)}{1 - M_1 |\lambda - \lambda_*| N(\xi, \eta)} \leq 2N(\xi, \eta). \tag{3.6}
\]

**Proof.** Let

\[
T_* = T(\lambda_*) = Y \Sigma^* X^H = \begin{bmatrix} Y_1 & y_* \end{bmatrix} \begin{bmatrix} \Sigma_1^* & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 & x_* \end{bmatrix}^H \tag{3.7}
\]
be the SVD of $T(\lambda_*)$, where $\Sigma_1^* = \text{diag}(\sigma_1^*, ..., \sigma_{n-1}^*) \in \mathbb{R}^{(n-1) \times (n-1)}$ is nonsingular and $Y, X \in \mathbb{C}^{n \times n}$ are unitary matrices. We decompose

$$u = XX^H u = X_1 u^1 + x_* u_2, \quad u^1 = X_1^H u, \quad u_2 = x_*^H u$$

$$v = YY^H v = Y_1 v^1 + y_* v_2, \quad v^1 = Y_1^H v, \quad v_2 = y_*^H v$$

and obtain

$$\hat{\mathbf{C}}_* := \begin{bmatrix} \frac{Y^H}{0^T} & 0 \end{bmatrix} C_* \begin{bmatrix} X & 0 \end{bmatrix} = \begin{bmatrix} \frac{\Sigma_1^*}{0^T} & \frac{v^1}{0} \\ \frac{u^1 H}{u^2} & 0 \end{bmatrix}.$$ 

Now, $\hat{\mathbf{C}}_*$ and, hence, $\mathbf{C}$ is nonsingular, iff the Schur complement

$$\begin{bmatrix} 0 & v_2 \\ \bar{u}_2 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ u^1 H \end{bmatrix} \Sigma_1^{-1} \begin{bmatrix} 0 \\ v^1 \end{bmatrix} = \begin{bmatrix} 0 & v_2 \\ \bar{u}_2 - u^1 H \Sigma_1^{-1} v^1 \end{bmatrix}$$

of $\Sigma_1^*$ in $\hat{\mathbf{C}}_*$ is nonsingular, i.e., iff $|\bar{u}_2| = |u_2| = |x_*^H u| = \cos \xi > 0$ and $|v_2| = |y_*^H v| = \cos \eta > 0$. But this is (3.4).

Now, let (3.4) hold. Then block elimination gives

$$\hat{\mathbf{C}}_*^{-1} = \begin{bmatrix} (\Sigma_1^*)^{-1} & - (\Sigma_1^*)^{-1} v_2^{-1} v^1 \\ - u^1 H \bar{u}_2^{-1} (\Sigma_1^*)^{-1} & u^1 H \bar{u}_2^{-1} (\Sigma_1^*)^{-1} v_2^{-1} v^1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & 0 \end{bmatrix}$$

where $B_{11} = \begin{bmatrix} I \\ - u^1 H \bar{u}_2^{-1} \end{bmatrix} (\Sigma_1^*)^{-1} \begin{bmatrix} I | - v_2^{-1} v^1 \end{bmatrix}$, $B_{12} = \begin{bmatrix} 0 \\ \bar{u}_2^{-1} \end{bmatrix}$ and $B_{21} = \begin{bmatrix} 0 \\ v_2^{-1} \end{bmatrix}$. This yields

$$\|C_*^{-1}\| = \|\hat{\mathbf{C}}_*^{-1}\| \leq \sqrt{\|B_{11}\|^2 + \|B_{12}\|^2 + \|B_{21}\|^2}.$$ 

With

$$\left\| \begin{bmatrix} I & - u_1^H \end{bmatrix} \right\| = \sqrt{1 + \frac{|u_1|^2}{\bar{u}_2}} \leq \frac{1}{\cos \xi}, \quad \left\| \begin{bmatrix} I & v_1 \end{bmatrix} \right\| = \sqrt{1 + \frac{|v_1|^2}{v_2}} \leq \frac{1}{\cos \eta}$$

we get $\|B_{11}\| \leq \frac{\|\Sigma_1^{-1}\|}{\cos \xi \cos \eta}$. Because of $\|B_{12}\| = 1/\cos \xi$, $\|B_{21}\| = 1/\cos \eta$ we end up with

$$\|C_*^{-1}\| = \|\hat{\mathbf{C}}_*^{-1}\| \leq \sqrt{\|\Sigma_1^{-1}\|^2 + \cos^2 \eta + \cos^2 \xi} \leq \sqrt{\|\Sigma_1^{-1}\|^2 + \frac{2}{\cos \eta \cos \xi}} = N(\xi, \eta).$$

Recall that $T(\lambda_*)^\dagger = X_1^H (\Sigma_1^*)^{-1} Y_1$, $\|T(\lambda_*)^\dagger\| = \|\Sigma_1^{-1}\|$. Nonsingularity of $C$ now follows from the perturbation lemma with the following considerations: For

$$\delta C := C - C_* = \begin{bmatrix} T(\lambda) - T(\lambda_*) & 0 \\ 0^T & 0 \end{bmatrix}$$

we obtain $\|\delta C\| = \|T(\lambda) - T(\lambda_*)\| \leq M_1 |\lambda - \lambda_*|$, and assuming $|\lambda - \lambda_*| \leq \tau(\xi, \eta) \leq 1/(2N(\xi, \eta) M_1)$, we end up with

$$\|\delta C\||C_*^{-1}\| \leq M_1 |\lambda - \lambda_*| N(\xi, \eta) \leq M_1 \tau(\xi, \eta) N(\xi, \eta) \leq \frac{1}{2} < 1.$$
giving us nonsingularity of $C$ and the bound (3.6). \[ \square \]

Furthermore, we may deduce uniform boundedness of the inverse matrices $C^{-1}$ from Lemma 3, when we assume uniform boundedness of the angles $\xi, \eta$ towards $\pi/2$ and if $\lambda$ is sufficiently close to $\lambda_*$. In order to get a better handling of feasible vectors we define the 'double cone'

$$
K(\varepsilon) := \{(u, v) : u, v \in \mathbb{C}^n, \|u\| = \|v\| = 1, \xi \leq \varepsilon, \eta \leq \varepsilon\}
$$

and obtain

**Corollary 4.** Assume $(A)$. Then, for all $\varepsilon_0$ with $0 < \varepsilon_0 < \pi/2$, there exists a radius $\tau_1$, $0 < \tau_1 \leq \tau_*$, such that for all $(u, v) \in K(\varepsilon_0)$ and all $\lambda, \mu \in S_1 := S(\lambda_*, \tau_1)$ the following holds:

(i) $C(\lambda, u, v)$ is nonsingular and its inverse is bounded by

$$
\|C(\lambda, u, v)^{-1}\| \leq \frac{N_0}{1 - M_1N_0|\lambda - \lambda_*|} \leq 2N_0 \quad (3.8)
$$

with $N_0 := N(\varepsilon_0, \varepsilon_0)$ from (3.5).

(ii) $C(\cdot, u, v)^{-1}$ is Lipschitz continuous

$$
\|C(\lambda, u, v)^{-1} - C(\mu, u, v)^{-1}\| \leq L C^{-1}|\lambda - \mu|
$$

where $L C^{-1} = 4N_0^2M_1$.

**Proof.** Set $\tau_1 = \tau(\varepsilon_0, \varepsilon_0) = \min\{\tau_*, 1/(2N_0M_1)\}$. Assertion (i) then follows directly from Lemma 3, part (iii). Moreover, for all $\lambda, \mu \in S_1$ one obtains

$$
\|C(\lambda)^{-1} - C(\mu)^{-1}\| = \|C(\lambda)^{-1} [C(\lambda) - C(\mu)] C(\mu)^{-1}\|
\leq \|C(\lambda)^{-1}\| \|C(\mu)^{-1}\| \|T(\lambda) - T(\mu)\| \leq 4N_0^2M_1|\lambda - \mu|.
\[ \square \]

From now on we assume

$$(u, v) \in K(\varepsilon_0) \quad \text{with fixed angle} \quad 0 < \varepsilon_0 < \pi/2. \quad (3.9)$$

Provided (3.9) and $\lambda \in S_1$, we have unique solutions

$$
\hat{x}(\lambda) := \begin{bmatrix} x(\lambda) \\ \mu(\lambda) \end{bmatrix} = C(\lambda)^{-1}e_{n+1}, \quad \hat{y}(\lambda) := \begin{bmatrix} y(\lambda) \\ \bar{\nu}(\lambda) \end{bmatrix} = C(\lambda)^{-H}e_{n+1}
$$

of the primal and the dual system (1.4), (1.6), resp. In particular, setting $\lambda = \lambda_*$ yields

$$
\hat{u}_* := \begin{bmatrix} u_* \\ \mu_* \end{bmatrix} := C_*^{-1}e_{n+1} = \begin{bmatrix} x_*/\bar{u}_2 \\ 0 \end{bmatrix}, \quad \hat{v}_* := \begin{bmatrix} v_* \\ \bar{\nu}_* \end{bmatrix} := C_*^{-H}e_{n+1} = \begin{bmatrix} y_*/\bar{v}_2 \\ 0 \end{bmatrix}. \quad (3.10)
$$

Next we show that $\xi_+ (\lambda), \eta_+ (\lambda) = O(|\lambda - \lambda_*|)$, where

$$
\xi_+ := \xi_+(\lambda) = \angle(\text{span} \{x(\lambda)\}, \text{span} \{x_*\}), \quad \eta_+ := \eta_+(\lambda) = \angle(\text{span} \{y(\lambda)\}, \text{span} \{y_*\})
$$

**Lemma 5.** Let $(u, v) \in K(\varepsilon_0)$ and $|\lambda - \lambda_*| \leq \tau_1$, $\tau_1$ defined as in Corollary 4. Then

$$
\sin \xi_+ \leq \frac{2N_0M_1}{\cos \xi} |\lambda - \lambda_*| \leq \frac{2N_0M_1}{\cos \varepsilon_0} |\lambda - \lambda_*| \quad (3.11)
$$

$$
\sin \eta_+ \leq \frac{2N_0M_1}{\cos \eta} |\lambda - \lambda_*| \leq \frac{2N_0M_1}{\cos \varepsilon_0} |\lambda - \lambda_*|. \quad (3.12)
$$
Proof. With \( x = u_* + \delta u \) and (3.8) and (3.10) one gets
\[
\|\delta u\| = \|x - u_*\| \leq \|\dot{x} - \dot{u}_*\| = \|C^{-1}e_{n+1} - C_*^{-1}e_{n+1}\|
\]
\[
= \|C^{-1}[C - C_*]C_*^{-1}e_{n+1}\| \leq \|C^{-1}\|\|T - T_*\|\|\dot{u}_*\|
\]
\[
\leq 2N_0M_1|\lambda - \lambda_*| \left(\frac{1}{\bar{u}_2}\right) = \frac{2N_0M_1}{\cos \xi_0}|\lambda - \lambda_*| =: \varepsilon
\]

Now we have
\[
\|x\|^2 = x^Hx = \frac{1}{|u_2|^2} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + \|\delta u\|^2 = \frac{1}{\cos^2 \xi} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + \|\delta u\|^2 \geq 1,
\]
and, because of \( x_*^Hx = x_*^H(u_* + \delta u) = x_*^Hu_* + x_*^H\delta u = \frac{1}{u_2} + x_*^H\delta u \), also
\[
|x_*^Hx|^2 = (x_*^Hx)(x_*^Hx) = \left( \frac{1}{u_2} + x_*^H\delta u \right) \left( \frac{1}{u_2} + x_*^H\delta u \right)
\]
\[
= \frac{1}{|u_2|^2} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + |x_*^H\delta u|^2 = \frac{1}{\cos^2 \xi} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + |x_*^H\delta u|^2.
\]
Thus, we obtain
\[
\cos^2 \xi_* = \left[ \frac{|x_*^Hx|}{\|x_*\| \|x\|} \right]^2 = \frac{\frac{1}{\cos^2 \xi} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + |x_*^H\delta u|^2 + \|\delta u\|^2 - \|\delta u\|^2}{\frac{1}{\cos^2 \xi} + 2\Re \left( \frac{x_*^H\delta u}{u_2} \right) + \|\delta u\|^2}
\]
\[
= 1 - \frac{\|\delta u\|^2 - |x_*^H\delta u|^2}{\|x\|^2} =: 1 - \omega,
\]
and eventually \( \sin^2 \xi_* = 1 - \cos^2 \xi_* = \omega = \frac{\|\delta u\|^2 - |x_*^H\delta u|^2}{\|x\|^2} \leq \left( \frac{\|\delta u\|}{\|x\|} \right)^2 \leq \varepsilon^2 \). Recall that \( 1 = u^Hx \leq \|x\| \). Bound (3.12) is proved in the same manner.

Now we are able to show that the Newton iterates \( \{\lambda_i\} \) are well-defined and converge quadratically. Instead of exploiting general Newton theory, we present a direct proof. Let us point out that we update the borderings which implies that the equations \( \mu(\lambda) = \mu(\lambda, u_i, v_i) = 0 \) to which one Newton step is applied are different for each \( i \). So a straightforward application of local Newton theorems is not possible.

Recall that we have \( \alpha_* = y_*^HT(\lambda_0)x_* \neq 0 \) since \( \lambda_* \) is an algebraically simple eigenvalue. At \( \lambda = \lambda_* \) and for \( (u, v) \in \mathcal{K}(\varepsilon_0) \) there holds
\[
\dot{\mu}(\lambda_*) = \dot{\mu}(\lambda_*, u, v) = -\frac{y_*^HT(\lambda_*)x_*}{u_2\bar{v}_2}, \quad \text{hence} \quad |\dot{\mu}(\lambda_*)| = \frac{|y_*^HT(\lambda_*)x_*|}{\cos \eta \cos \xi} \geq |\alpha_*| > 0.
\]

Next we want to prove Lipschitz continuity of \( \dot{\mu}(\cdot) \) on \( S_1 \). Examination of (2.4) shows that every factor there is Lipschitz continuous, namely \( C(\cdot)^{-1} \) with constant \( L_{C^{-1}} = 4N_0^2M_1 \) and \( C(\cdot) \) with constant \( L_1 \) from (3.3). Furthermore, these factors are bounded on \( S_1 \), in that \( \|C(\lambda)^{-1}\| \leq \max_{\lambda \in S_1} \|C(\lambda)^{-1}\| \leq 2N_0, \quad \|\hat{C}(\lambda)\| = \|\hat{T}(\lambda)\| \leq \max_{\lambda \in S_1} \|\hat{T}(\lambda)\| \leq M_1 \). This gives us
\[
|\dot{\mu}(\lambda) - \dot{\mu}(\rho)| \leq \left[ 2L_{C^{-1}} \max_{\lambda \in S_1} \|C(\lambda)^{-1}\| \max_{\lambda \in S_1} \|\hat{T}(\lambda)\| + L_1 \max_{\lambda \in S_1} \|C(\lambda)^{-1}\|^2 \right] |\lambda - \rho|
\]
\[
\leq L\mu|\lambda - \rho|
\]
for all \( \lambda, \rho \in S_1 \), where \( L_{\mu} := 2L_{C^{-1}}(2N_0)M_1 + L_1(2N_0)^2 = 16N_0^3M_1^2 + 4N_0^2L_1 \). This implies that \(|\dot{\mu}(\lambda)|\) is uniformly bounded from below, whenever \(|\lambda - \lambda_*|\) is sufficiently small. And to be more precise

**Lemma 6.** For all \( \varepsilon_0 \) with \( 0 < \varepsilon_0 < \pi/2 \) there exists a radius \( \tau_2, \; 0 < \tau_2 \leq \tau_1 \), such that

\[
|\dot{\mu}(\lambda, u, v)| \geq \frac{1}{2} |\alpha_*|
\]

holds for all \((u, v)\in K(\varepsilon_0)\) and all \( \lambda \in S_2 = \overline{S}(\lambda_*, \tau_2) \).

**Proof.** Set \( \tau_2 := \min \left\{ \tau_1, \frac{|\alpha_*|}{2\mu} \right\} \). For all \( \mu \in S_2 \) we get immediately

\[
|\dot{\mu}(\lambda)| \geq |\dot{\mu}(\lambda_*)| - |\dot{\mu}(\lambda) - \dot{\mu}(\lambda_*)| \geq |\alpha_*| - L\mu \tau_2 \geq \frac{1}{2} |\alpha_*|. \quad \Box
\]

Thus, the Newton step (2.5) starting from \( \lambda \) and arriving at \( \lambda_* \) is well-defined if \( \lambda \in S_2 \). The error is given by

\[
|\lambda + \lambda_*| = \left| \lambda - \lambda_* - \frac{\mu(\lambda)}{\dot{\mu}(\lambda)} \right| = \frac{1}{|\dot{\mu}(\lambda)|} |\mu(\lambda) - \mu(\lambda_*) - \dot{\mu}(\lambda)(\lambda - \lambda_*)| \leq \frac{2}{|\alpha_*|} \frac{L\mu}{2} |\lambda - \lambda_*|^2 = Q_0|\lambda - \lambda_*|^2,
\]

where \( Q_0 := \frac{L\mu}{2}|\alpha_*| \). Notice that \( \mu(\lambda_*, u, v) = 0 \).

If additionally \( |\lambda - \lambda_*| \leq \tau_3 := \min \left\{ \tau_2, \frac{1}{2Q_0} \right\} \), then we can estimate further

\[
|\lambda + \lambda_*| \leq Q_0\tau_3|\lambda - \lambda_*| \leq \frac{1}{2}|\lambda - \lambda_*|.
\]

In particular, we have \( |\lambda + \lambda_*| \leq \tau_3 \leq \tau_2 \), i.e., \( \lambda_* \in S_2 \) giving us a well-defined next Newton step \( \lambda_* \rightarrow \lambda_+ = \mu(\lambda_+)/\dot{\mu}(\lambda_*) \) provided that the updated borderings \( u_* = x/\|x\|, \; v_* = y/\|y\| \) then used belong to \( K(\varepsilon_0) \). Hence, they have to satisfy

\[
\xi_* = \angle(\text{span} \{x\}, \text{span} \{x_*\}) \leq \varepsilon_0, \quad \eta_* = \angle(\text{span} \{y\}, \text{span} \{y_*\}) \leq \varepsilon_0.
\]

But this is true because of Lemma 5, whenever

\[
\sin \xi_*, \; \sin \eta_* \leq \frac{2N_0M_1}{\cos \varepsilon_0}|\lambda - \lambda_*| \leq \sin \varepsilon_0
\]

which is guaranteed if \( |\lambda - \lambda_*| \leq \tau_4 := \frac{\sin \varepsilon_0 \cos \varepsilon_0}{2N_0M_1} \).

Altogether we end up with the following convergence theorem.

**Theorem 7.** Assume (A). Then for every \( \varepsilon_0 \), \( 0 < \varepsilon_0 < \pi/2 \), there exists a radius \( \tau_0 > 0 \) and constants \( Q_0 > 0, \; K_0 > 0 \) with \( Q_0\tau_0 \leq 1/2 \), such that GRQI is well defined for every initial triplet \((\lambda_0, u_0, v_0)\) with \((u_0, v_0)\in K(\varepsilon_0)\), \( \lambda_0 \in S_0 = \overline{S}(\lambda_*, \tau_0) \), and converges in the following sense

\[
\lim_{i \to \infty} \lambda_i = \lambda_*, \quad \lim_{i \to \infty} \xi_i = \lim_{i \to \infty} \eta_i = 0,
\]

where \( \xi_i = \angle(\text{span} \{u_i\}, \text{span} \{x_*\}), \; \eta_i = \angle(\text{span} \{v_i\}, \text{span} \{y_*\}) \). The rate of convergence is characterized by

\[
|\lambda_{i+1} - \lambda_*| \leq Q_0|\lambda_i - \lambda_*|^2 \leq \frac{1}{2}|\lambda_i - \lambda_*| \quad (3.15)
\]
and
\[
\sin \xi_{i+1} \leq K_0 |\lambda_i - \lambda_*| \leq \frac{K_0}{Q_0} \left( \frac{1}{2} \right)^{2i}.
\]  
(3.16)

**Proof.** Set \( \tau_0 = \min \left\{ \tau_*, \frac{\sin \xi_0 \cos \xi_0}{2N_0M_1}, \frac{1}{2L_u}, \frac{1}{2Q_0} \right\} \), \( Q_0 = \frac{L_u}{|a_1|} \), \( K_0 = \frac{2N_0M_1}{\cos \xi_0} \). Note that \( \tau_0 \leq \tau_2 \leq \tau_1 \) so Corollary 4 and Lemma 5 are valid. Then (3.13) and (3.14) yield (3.15). Moreover, because of \( Q_0 |\lambda_0 - \lambda_*| \leq Q_0 \tau_0 \leq 1/2 \), the left inequality of (3.15) implies
\[
|\lambda_i - \lambda_*| \leq \frac{1}{Q_0} \left( \frac{1}{2} \right)^{2i}
\]  
(3.17)

which, together with Lemma 5, gives (3.16). \( \Box \)

The right inequality in (3.15) states at least \( Q \)-linear convergence with factor 1/2, the left one the \( Q \)-quadratic convergence of the sequence of eigenvalue approximations \( \{\lambda_i\} \) towards \( \lambda_* \) and implies the \( R \)-quadratic convergence displayed in (3.17). Inequality (3.16) states the \( R \)-quadratic convergence of the sequences of angles \( \xi_i \), \( \eta_i \) towards zero.

Note that the factor 1/2 can be replaced by any value in \( (0, 1) \) with appropriately modified constants in the proofs.

4. A Jacobi–Davidson-like extension. As in the linear case we can reformulate the algorithm—which is currently a 2-vector iteration—to a subspace iterating one and in this coherence to one of Jacobi–Davidson kind.

Let \( U \) with basis \( U \) denote the subspace giving approximations for the right eigenvector, and \( V \) with basis \( V \) denote the one containing approximations for the left eigenvector, where \( U^H U = V^H V = I \).

To determine new approximations \( u \in U \) and \( v \in V \) we consider Petrov–Galerkin conditions on the right and left residual, namely
\[
r_u = T(\lambda) u \perp V \quad \text{and} \quad r_v = T(\lambda)^H v \perp U.
\]
Writing \( u = Uc \) and \( v = Vd \) yields \( \lambda \) as eigenvalue and \( c, d \neq 0 \) as right and left eigenvectors of the \( k \)-dimensional projected eigenvalue problem defined by \( T_k(\lambda) := V^H T(\lambda) U \in \mathbb{C}^{k \times k} \), i.e.,
\[
V^H T(\lambda) U c = 0 \quad \text{and} \quad U^H T^H(\lambda) V d = 0.
\]  
(4.1)

Dependent on the properties of the projected problem one gets a number of eigen-triplets \( (\lambda, c, d) \) and has in this sense the opportunity to select an appropriate triplet \( (\lambda, u, v) \) with \( v^H T(\lambda) u = 0 \), \( u = Uc \), \( v = Vd \) and force convergence in a special direction in contrast to the single- or two-vector algorithms.

Whether the projected nonlinear eigenvalue problems (4.1) are solvable in the general case and if so how \( \lambda - \lambda_* \) depends on the angles \( \angle(\text{span} \{x_*\}, \text{im} \ U) \) and \( \angle(\text{span} \{y_*\}, \text{im} \ V) \) is still an open problem. When \( T(\lambda) \) is a polynomial of degree \( l \) then (4.1) is also a polynomial eigenvalue problem.

The approximations \( u, v \) are improved by solving correction equations which are equivalent to the primal and dual system (1.4), (1.6) of the GRQI. We deduce them from the residual formulation (2.6), i.e., from
\[
\begin{bmatrix}
T(\lambda) & v \\
u^H & 0
\end{bmatrix}
\begin{bmatrix}
s \\
\mu
\end{bmatrix}
= 
\begin{bmatrix}
-r_u \\
0
\end{bmatrix}
\iff
T(\lambda)s + v\mu = -r_u
\quad \text{and} \quad u^H s = 0.
\]
Multiplying the first equation from left by \( v^T \) gives 

\[
v^T \mu = -v^T r_u = 0
\]

since \( v^T r_u = v^T T(\lambda) u = 0 \). Hence, we have \( \mu = -v^T T(\lambda) s \). Substituting \( \mu \) in the first equation by this expression yields 

\[
T(\lambda)s - v^T T(\lambda)s = -r_u.
\]

Since \( (I - uu^T)s = s \), we end up with

\[
(I - vv^T)T(\lambda)(I - uu^T)s = -r_u, \quad u^H s = 0
\]  

(4.2)

The same procedure for the adjoint system produces the adjoint correction equation

\[
(I - uu^T)T^H(\lambda)(I - vv^T)t = -r_v, \quad v^H t = 0.
\]  

(4.3)

The solutions \( s, t \) are obtained with preconditioned GMRES and expand the subspaces after orthogonalization with respect to \( U, V \), resp., using modified Gram–Schmidt (MGS) and subsequent normalization. We named the method GJD—on account of Generalized Jacobi–Davidson.

**Algorithm 2. GJD**

**Input:** approximations \((u, v)\) s.t. \( u^H u = v^H v = 1 \), tolerance \( \epsilon > 0 \)

**S1:** Set \( U = [u], V = [v] \)

**for** \( i = 1, 2, \ldots \)

**S2:** Solve \( V^H T(\lambda) U c = 0 \) and \( U^H T(\lambda)^H V d = 0 \) for \( (\lambda, c, d) \)

**S3:** \( u = Uc, \ v = Vd, \ r_u = T(\lambda) u, \ r_v = T(\lambda)^H v \)

**S4:** if \( \min\{\|r_u\|/\|u\|, \|r_v\|/\|v\|\} < \epsilon \) stop

**S5:** Compute \( s \perp u, \ t \perp v \) from

\[
(I - vv^T)T(\lambda)(I - uu^T)s = -r_u \quad (I - uu^T)T^H(\lambda)(I - vv^T)t = -r_v
\]

**S6:** MGS\((U, s)\), MGS\((V, t)\), normalize \( s, t \), set \( U = [U, s] \), \( V = [V, t] \)

We terminate if at least one of the residuals is sufficiently small in order to have a fair comparison with standard nonlinear JD which determines only right eigenvectors. Though, if both eigenvectors are of interest, just one further step is needed, in general, to compute the second one to satisfying accuracy.

The projected systems are preconditioned with ’projected preconditioners’ along the lines of [4].

Recall that standard nonlinear Jacobi–Davidson works only with approximations to the right eigenvector. There \( (\lambda, c) \) is determined from the projected problem \( U^H T(\lambda) U c = 0 \) and gives \( u = Uc \) so we have \( u^H T(\lambda) u = 0 \). Then the corrections \( s \) is computed as solution of

\[
\left( I - \frac{T(\lambda) uu^T}{u^H T(\lambda) u} \right) T(\lambda)(I - uu^T)s = -r_u, \quad u^H s = 0,
\]  

(4.4)

cf. (1.3) and [5].

The major difference of our correction formulae (4.2), (4.3) to the nonlinear Jacobi–Davidson correction formula (4.4) is the absence of the derivative of \( T \) in the projectors and the fact that only orthogonal and no skew projectors occur.
Of course, solving the two systems (4.2), (4.3) per step instead of the one system (4.4) in standard nonlinear Jacobi–Davidson means double costs. However, we hope for superior convergence properties in case of ill-conditioned problems that overcomes this drawback. Solving these systems in a parallel implementation also seems suitable.

In the Hermitian case we have \( x_\ast = y_\ast \). Hence, we can choose \( v = u \), cf. [14], so we have to solve only one projected system

\[
(I - uu^H)T(\lambda)(I - uu^H)s = -T(\lambda)u = r_u, \quad s \perp u
\]

(4.5)

the matrix of which is Hermitian whereas the matrix of the standard nonlinear Jacobi–Davidson correction formula (4.4) is not Hermitian, in general. Notice that in case of linear problems \( T(\lambda) = A - \lambda I \) equation (4.5) is the correction formula of the standard linear Jacobi–Davidson method [13] so for Hermitian linear problems our method reduces to that method.

5. Comparison of asymptotic condition numbers. In the standard nonlinear JD method parameters \( (\lambda, u) \) with \( u^HT(\lambda)u = 0 \) are used, and one has to solve the correction equations

\[
\left( I - \frac{pu^H}{u^Hp} \right) T(\lambda)(I - uu^H)s = -r_u \perp u, \quad u^Hs = 0
\]

where \( p = \hat{T}(\lambda)u \), \( r_u = T(\lambda)u \). In the new method GJD the parameters \( (\lambda, u, v) \) are chosen such that \( v^HT(\lambda)u = 0 \), and correction equations

\[
(I - vv^H)T(\lambda)(I - uu^H)s = -r_u \perp v, \quad u^Hs = 0
\]

and the corresponding dual ones have to be solved. The projected matrices

\[
A_{JD}(\lambda, u) = \left( I - \frac{pu^H}{u^Hp} \right) T(\lambda)(I - uu^H)
\]

\[
A_{GJD}(\lambda, u, v) = (I - vv^H)T(\lambda)(I - uu^H)
\]

used above define nonsingular \((n - 1)\)-dimensional mappings

\[
A_{JD} : s \perp u \mapsto A_{JD}s \perp u, \quad A_{GJD} : s \perp u \mapsto A_{GJD}s \perp v
\]

(5.1)

provided that \( \lambda, u, v \) are sufficiently good and, for JD, that \( x_\ast^Hp_\ast \neq 0 \) where \( p_\ast = \hat{T}(\lambda_\ast)x_\ast \). The latter condition guarantees \( u^Hp \neq 0 \) and does, for nonnormal \( T \), not follow from the simplicity condition \( y_\ast^Hp \neq 0 \), in general. For GJD, this condition is not required.

Consider the mappings (5.1) at the solution \( (\lambda, u, v) = (\lambda_\ast, x_\ast, y_\ast) \). Using the SVD (3.7) of \( T(\lambda_\ast) \), the identities \( I - x_\ast x_\ast^H = X_1X_1^H \), \( I - y_\ast y_\ast^H = Y_1Y_1^H \) and the normalized vector \( q_\ast = p_\ast/\|p_\ast\| \) we obtain

\[
A_{JD} = A_{JD}(\lambda_\ast, x_\ast) = PY_1 \Sigma_1^*X_1^H, \quad A_{GJD} = A_{GJD}(\lambda_\ast, x_\ast, y_\ast) = Y_1 \Sigma_1^*X_1^H
\]

(5.2)

with the skew projector \( P = I - \frac{p_\ast x_\ast^H}{x_\ast^Hp_\ast} \). We have rank \( P = n - 1 \) and \( Pq_\ast = PHx_\ast = 0 \) so

\[
P = X_1 \Pi Q^H = \begin{bmatrix} X_1 & 0 \end{bmatrix} \begin{bmatrix} \Pi_1^* & 0 \\ 0^T & 0 \end{bmatrix} \begin{bmatrix} Q_1 | q_\ast \end{bmatrix}^H = X_1 \Pi_1^*Q_1^H
\]

(5.3)
is a unitary reduction of $P$ with $\Pi_1^s = (Q_1^H X_1)^{-1}$ and

$$
\|P\| = \|\Pi_1^s\| = \|(Q_1^H X_1)^{-1}\| = \frac{1}{|x_s^H q_s|} = \frac{1}{\cos \psi_s}, \quad \|(\Pi_1^s)^{-1}\| = 1 (5.4)
$$

where

$$
\psi_s = \angle (\text{span} \{p_s\}, \text{span} \{x_s\}) = \angle (\text{span} \{q_s\}, \text{span} \{x_s\}) = \angle (\text{im} Q_1, \text{im} X_1) < \pi/2.
$$

Replacing $P$ in (5.2) by the representation (5.3), we end up with

$$
A_{JD} = X_1 \Pi_1^s Q_1^H Y_1 \Sigma_1^s X_1^H, \quad A_{GJD} = Y_1 \Sigma_1^s X_1^H.
$$

Hence, the matrix representation $M_{JD}$ of $A_{JD}$ with respect to the orthogonal basis $X_1$ of the orthogonal complement $\{s : s \perp x_s\}$ and the representation $M_{GJD}$ of $A_{GJD}$ with respect to the orthogonal bases $X_1$ of $\{s : s \perp x_s\}$ and $Y_1$ of $\{t : t \perp y_s\}$, cf. (5.1), are the nonsingular $(n-1) \times (n-1)$ matrices

$$
M_{JD} = \Pi_1^s (Q_1^H Y_1) \Sigma_1^s, \quad M_{GJD} = \Sigma_1^s,
$$

resp. So for GJD we immediately obtain

$$
\|A_{GJD}^{-1}\| = \|M_{GJD}^{-1}\| = \|(\Sigma_1^s)^{-1}\| = \|T(\lambda_s)^\dagger\|.
$$

For JD we have $M_{JD}^{-1} = \Sigma_1^s (Q_1^H Y_1)^{-1} (\Pi_1^s)^{-1}$ which, considering (5.4) and the analogue relation

$$
\|(Q_1^H Y_1)^{-1}\| = 1/|y_s^H q_s| = 1/\cos \varphi_s
$$

where $\varphi_s = \angle (\text{span} \{p_s\}, \text{span} \{y_s\}) = \angle (\text{span} \{q_s\}, \text{span} \{y_s\}) = \angle (\text{im} Q_1, \text{im} Y_1) < \pi/2$, leads to

$$
\frac{1}{\cos \varphi_s} \cdot \frac{\cos \psi_s}{\|T(\lambda_s)\|} \leq \|A_{JD}^{-1}\| = \|M_{JD}^{-1}\| \leq \frac{1}{\cos \varphi_s} \cdot \|T(\lambda_s)^\dagger\| (5.5)
$$

If $\varphi_s$ is very close to $\pi/2$, i.e., if $\lambda_s$ is extremely sensitive, then $\|A_{JD}^{-1}\|$ will be very large which may cause problems when solving the correction equation using Krylov methods.

Notice that for the special case $T(\lambda) = A - \lambda I$ we have $\dot{T}(\lambda_s) = -I$, hence, $q_s = -x_s$, $\psi_s = 0$ so (5.5) reduces to

$$
\frac{1}{\cos \varphi_s} \cdot \frac{1}{\|T(\lambda_s)\|} \leq \|A_{JD}^{-1}\| = \|M_{JD}^{-1}\| \leq \frac{1}{\cos \varphi_s} \cdot \|T(\lambda_s)^\dagger\|
$$

with the denominator $\cos \varphi_s = |y_s^H x_s|$ which may be arbitrarily small when $A$ is strongly nonnormal.

6. Numerical Examples. In case of nonlinear eigenproblems one comes to facing the problem of finding a solution of the projected small dimensional system in S2 of GJD, which is a nonlinear eigenvalue problem itself. Voss [16] made several suggestions for such dense (and also for sparse) problems. Especially suitable seems the method of successive linear problems (introduced in [10]), where linear eigenproblems are to be solved, and a selection of the wanted eigenvalue is possible. This is also an inner iterative process which should be solved up to a certain accuracy as is the solution of the correction equation.
If one uses one-vector iterations—which are, in general, highly dependent on the starting value and vector—for the solution of the projected eigenvalue problem, then convergence to the same eigenvalue is very unlikely for different JD-like methods, in that in every outer iteration another starting pair/triplet may be produced. In order to get an idea of convergence behavior for different JD methods including GJD we need them to converge to the same eigenvalue.

However, in cases of polynomial eigenvalue problems there is no need to deal with this issue, since several methods giving all eigenvalues exist. MATLAB realizes this with the function \texttt{polyeig}, which is based on the QZ algorithm. Therefore, in this paper we only look at polynomial nonlinear problems, where we compute all eigenvalues of the projected problems.

6.1. Bordered systems: Single vector algorithms – Newton vs GRQI. Before addressing subspace iterating methods we take a look at the previously defined algorithm GRQI and want to compare it with different variants of Newton’s method and Rayleigh quotient iteration. The following table lists the names of the methods which were implemented including the systems to be solved and the corresponding \( \lambda \)-update formulae.

**GRQI:** Step \((\lambda, u, v) \mapsto (\lambda_+, u_+, v_+)\)

\[
\begin{bmatrix}
T(\lambda)
\begin{bmatrix}
T(\lambda)u
u^H
0
\end{bmatrix}
\begin{bmatrix}
s
\mu
\end{bmatrix}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
= - \begin{bmatrix}
T(\lambda)u
0
\end{bmatrix}
\begin{bmatrix}
t
\bar{v}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
\]

\(\lambda_+ = \lambda - \frac{\mu}{(v + t)^H \hat{T}(\lambda)(u + s)}\), \(u_+ = \frac{u + s}{\|u + s\|}\), \(v_+ = \frac{v + t}{\|v + t\|}\)

**RQI:** Step \((\lambda, u) \mapsto (\lambda_+, u_+)\)

\[
\begin{bmatrix}
T(\lambda)
\begin{bmatrix}
T(\lambda)u
u^H
0
\end{bmatrix}
\begin{bmatrix}
s
\mu
\end{bmatrix}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
= - \begin{bmatrix}
T(\lambda)u
0
\end{bmatrix}
\begin{bmatrix}
t
\bar{v}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
\]

\(\lambda_+ = \lambda - \frac{(u + s)^H T(\lambda)(u + s)}{(u + s)^H \hat{T}(\lambda)(u + s)}\), \(u_+ = \frac{u + s}{\|u + s\|}\)

**Two-sided RQI:** Step \((\lambda, u, v) \mapsto (\lambda_+, u_+, v_+)\)

\[
\begin{bmatrix}
T(\lambda)
\begin{bmatrix}
T(\lambda)u
u^H
0
\end{bmatrix}
\begin{bmatrix}
s
\mu
\end{bmatrix}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
= - \begin{bmatrix}
T(\lambda)u
0
\end{bmatrix}
\begin{bmatrix}
t
\bar{v}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
\]

\(\lambda_+ = \lambda - \frac{(v + t)^H \hat{T}(\lambda)(u + s)}{(v + t)^H \hat{T}(\lambda)(u + s)}\), \(u_+ = \frac{u + s}{\|u + s\|}\), \(v_+ = \frac{v + t}{\|v + t\|}\)

**Newton:** Step \((\lambda, u) \mapsto (\lambda_+, u_+)\)

\[
\begin{bmatrix}
T(\lambda)
\begin{bmatrix}
T(\lambda)u
u^H
0
\end{bmatrix}
\begin{bmatrix}
s
\mu
\end{bmatrix}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
= - \begin{bmatrix}
T(\lambda)u
0
\end{bmatrix}
\begin{bmatrix}
t
\bar{v}
\end{bmatrix}
- \begin{bmatrix}
0
\end{bmatrix}
\]

\(\lambda_+ = \lambda + \mu\), \(u_+ = \frac{u + s}{\|u + s\|}\)

Newton’s iteration was defined in (1.2) for the nonlinear eigenvalue problem. Note, that the eigenvalue approximation here comes as part of the solution of the system and not as Rayleigh quotient.

**Example 1.** Since we are interested in the performance of the algorithms with respect to ill-conditioned problems, we created the following quadratic eigenvalue problem

\[T(\lambda)x = (\lambda^2 A + \lambda B + C)x = 0\]
where \( A = I, \ B = [B_1 \ e_n] \). \( B_1 \in \mathbb{R}^{n \times (n-1)} \) is random sparse with just 5 diagonals arranged around the main diagonal; \( e_n \) denotes the \( n \)-th unit vector.

\( C \) is a especially constructed sparse matrix of order \( n = 1000 \) with \( \lambda_* = 0 \) corresponding to an adjustable angle \( \varphi \) between left and right eigenvector. The class of matrices was introduced in [12]. When generating \( C \) the user is asked to set the average number of nonzeros per row, which was set to 8 in our case, and additionally the condition number \( \kappa \) of the matrix \( \Sigma_1 \) coming from the SVD

\[
C - \lambda_* I = C = Y \Sigma X^H = [Y_1|x_*] \begin{bmatrix} \Sigma_1 & \cdot \cdot \cdot & \cdot \cdot \cdot \\ \cdot \cdot \cdot & \cdot \cdot \cdot & \cdot \cdot \cdot \\ \cdot \cdot \cdot & \cdot \cdot \cdot & 0 \end{bmatrix} [X_1|x_*]^H.
\]

The aim to leave the choice of the angle to the user was obtained by constructing \( C \) via its SVD as follows: Set \( x_* = e_n \) and choose \( [X_1|x_*] = I_n \) for simplicity. I.e., \( C \) is generated by \( Y \) and \( \Sigma \). The corresponding left singular vector \( y_* \) which, in this case, is also the left eigenvector is determined by the given angle \( \varphi \) to its right eigenvector. It is built with an equally distributed sparsity pattern which carries over to the the matrix \( Y \) of left singular vectors and hence to \( C \). \( Y_1 \) is determined by Householder transformation. The singular values on the diagonal of \( \Sigma_1 \) are random in the interval \([100/\kappa, 100]\) with \( \sigma_1 = 100, \sigma_{n-1} = 100/\kappa \).

But how do these properties carry over to the quadratic problem above? Let \((\lambda_* = 0, x_*, y_*)\) be an eigentriplet of \( C \). Then it forms an eigentriplet of \( T \) as well. To see this, we linearize \( T(\lambda)x \) by substituting \( \lambda x \) by \( u \), see [15]. Thus we get

\[
\lambda A u + Bu + Cx = 0
\]

which is equivalent to

\[
\begin{bmatrix} 0 & -I \\ C & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

\[
\iff \tilde{T}(\lambda) \begin{bmatrix} x \\ u \end{bmatrix} := \begin{bmatrix} \lambda I & -I \\ C & B + \lambda A \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

Since \( C \) has rank drop one by definition \( \det(\tilde{T}(0)) = 0 \), hence \( \lambda_* = 0 \) is an eigenvalue of \( \tilde{T} \), hence of \( T \). And due to \( Cx_* = 0 \cdot x_* = 0 \) we have \( \tilde{T}(0)[x_*^H 0]^H = 0 \), i.e. \( x_* \) is the corresponding right eigenvector with respect to \( \lambda_* = 0 \).

Applying the same procedure with \( u = \tilde{\lambda} x \) to \( T(\lambda)^H y = \tilde{\lambda}^2 A^H y + \tilde{\lambda} B^H y + C^H y = 0 \) yields \( y_* \), as associated left eigenvector. Moreover, since \( \tilde{T}(\lambda_*) = B \) we have \( \alpha_* = y_*^H B x_* = y_*^H x_* \) as adjustable parameter.

Table 1 shows the characteristics of the convergence history for the methods where \( \text{iter} \) gives the number of iterations till convergence, the columns \( \text{gm}_1/\text{iter} \) and \( \text{gm}_2/\text{iter} \) provide information about the sum over all iterations of GMRES during the process for the primal and the dual system, respectively, and behind the backslash this number divided by the number of outer iterations is displayed, hence one can get an impression of the average number of GMRES-steps.

The last column shows \( \text{cpu-time} \), which is not that reliable but gives some idea of taken time. All experiments were run under Matlab 7.1 on an Intel Pentium M processor with 1.3 GHz and 512 MB RAM. Stopping criterion for each algorithm was a residual norm of one of the residuals below \( 1e-09 \) or the reaching of 100 iterations.

Recall, that the term ‘ill-conditioned’ with respect to the angle of eigenvectors does not start with 70 degree or something close to it. Even when \( \varphi = 89^\circ \), one gets
1/ \cos \varphi \approx 57.3. We run the problem twice, first with \varphi = 89^\circ, see Table 1, second with \varphi = 89.99999^\circ, see Table 2, and fixed cond \Sigma_1 = 1000. Apparently, in the case of single-vector iterations, results depend more heavily on initial approximations and not so much on this precise angle of the solution vectors. For the first run—although better conditioned—only GRQI managed to find the wanted eigenvalue, when all other methods break with another one. The initial value was \lambda_0 = 0.1. In the second run only the simple Newton iterations fail, GRQI and two-sided GRQI give almost identical results. Figure 1 and 2 present residual norms versus iteration numbers correlated to Tables 1 and 2.

### Table 1

Example 1 with \( n = 1000 \), \( \lambda_* = 0 \), \( \varphi = 89^\circ \), cond \( \Sigma_1 = 1000 \)

| \(|\lambda_{iter} - \lambda_*|\) | iter | \(|\text{res}_u\|\) | \(|\text{res}_v\|\) | gm1/iter | gm2/iter | cpu |
|-----------------|------|-----------------|-----------------|--------|--------|-----|
| GRQI            | 4.0e−15 | 11 | 2.6e−11 | 3.8e−09 | 45/4.1 | 47/4.3 | 14.0 |
| RQI             | 1.4e−04 | 14 | 7.3e−13 | 75/5.4  | 12.6   |
| Newton          | 1.0e+00 | 11 | 9.7e−12 | 103/9.4 | 37.1   |
| Two-sided RQI   | 1.4e−04 | 6  | 2.8e−11 | 6.5e−14 | 27/4.5 | 31/5.2 | 10.0 |

### Table 2

Example 1 with \( n = 1000 \), \( \lambda_* = 0 \), \( \varphi = 89.99999^\circ \), cond \( \Sigma_1 = 1000 \)

| \(|\lambda_{iter} - \lambda_*|\) | iter | \(|\text{res}_u\|\) | \(|\text{res}_v\|\) | gm1/iter | gm2/iter | cpu |
|-----------------|------|-----------------|-----------------|--------|--------|-----|
| GRQI            | 6.5e−10 | 5  | 6.6e−10 | 1.4e−07 | 590/5.9 | 6.6   |
| RQI             | 1.4e−09 | 7  | 6.6e−10 | 27/3.9  | 100.2  |
| Newton          | 1.9e−11 | 100 | 5.8e−08 | 590/5.9 | 100.2  |
| Two-sided RQI   | 2.4e−11 | 5  | 9.7e−11 | 5.6e−10 | 23/4.6 | 29/5.8 | 8.8  |

**Fig. 1.** Residual norms of GRQI, RQI, Newton and two-sided RQI against iteration number for Example 1 corresponding to Table 1.

### 6.2. Projected systems: Jacobi–Davidson type methods.

We compared our method GJD with nonlinear versions of JD and alternating and two-sided JD suggested by Hochstenbach [6], denoted by AJD and TJD. The latter were especially designed
for computing eigenvalues of (slightly) nonnormal matrices. The alternating version switches between the computation of the left and the right eigenvector, whilst the two-sided one solves two correction equations with skew projections—one for each eigenvector. The two-sided variant for polynomial problems is also described in [6], whereas we adjusted the alternating one to nonlinear problems by taking the standard nonlinear update (4.4) and its dual as correction equations.

Correction equations are solved by a maximum of 15 iterations of preconditioned GMRES per outer iteration $i$, where the residual tolerance, which is moderate while launching the program, is lowered proportional to the norm of the outer residual. To be more precise: We start with a tolerance $\text{tol} = 0.01$ and set $\text{tol}$ afterwards always about one decimal place smaller than the norm of the actual outer residual. Table 3 shows that in this example a maximum of 3 GMRES-steps is needed.

Since we are primarily interested in the performances of the methods compared to each other, we use the matrices produced by MATLAB’s $\text{luinc}$ with drop tolerance 0.005 as preconditioner, although this might not be the very best choice. With this intention we also renounce on restarting the methods with selected vectors, i.e., the number of vectors spanning the search space is equivalent to the number of the outer iteration. The process is terminated when one of the residual norms is smaller than $1e^{-09}$.

Table 3 and Figure 3 show the results for the quadratic problem defined in Example 1 in the ill-conditioned case, where the angle between the eigenvectors corresponding to the eigenvalue $\lambda_\star = 0$ is approximately $\varphi = 89.9999^\circ$.

Benefits of GJD are easily visible in contrary to the previous section. It converges in half of JD’s iterations and is faster, although the number of solved systems is almost equivalent. AJD did not converge within 80 iterations, TJD needs some more than JD, but roughly thrice its time. The fact, that left residuals are not small at all is due to very poor starting vectors. To check this we started the methods with
Table 3
Example 1 with $n = 1000$, $\lambda_* = 0$, $\varphi = 89.9999^\circ$, cond $\Sigma_1 = 1000$

| Method | $|\lambda_{iter} - \lambda_*|$ | iter | $\|\text{res}_u\|$ | $\|\text{res}_v\|$ | $gm_1/\text{iter}$ | $gm_2/\text{iter}$ | cpu |
|--------|-------------------------------|------|---------------------|---------------------|-------------------|-------------------|-----|
| JD     | 2.5e-10                       | 44   | 9.0e-11             |                     | 81/1.8            |                   | 33.6|
| AJD    | 8.2e-01                       | 80   | 1.4e+00             |                     | 55/0.7            |                   | 183.9|
| TJD    | 5.4e-10                       | 47   | 9.7e-10             | 4.6e-01             | 92/2.0            | 120/2.6          | 94.6|
| GJD    | 5.4e-12                       | 21   | 2.4e-11             | 1.6e+00             | 50/2.4            | 64/3.0            | 28.2|

Fig. 3. Residual norms of JD, alternating and two-sided JD and GJD against iteration number for Example 1 corresponding to Table 3. For the latter ones right and left residuals are displayed.

$x_0 = y_0$ and found, that at least 80 iterations were needed.

7. Conclusions. We presented a method that computes simple eigenvalues of nonlinear eigenvalue problems. At this time there are lots of well-working methods available and efforts in this direction are made for half a century now. We showed that GJD is particularly suited for strongly ill-conditioned problems, where $|y^H J(\lambda_*) x_*|$ is very small, but naturally also works for well-conditioned problems. However, we have to solve two instead of one correction equations and hence have almost double costs compared to standard Jacobi–Davidson. Conversely, the experiments have shown that, because of less iterations, GJD can be faster in spite of that.

Our recommendation is to try standard solvers like Jacobi–Davidson first, and in case of breakdown or slow convergence switch to GJD. If the left eigenvector is of interest, too, it may be preferable to use GJD firsthand.

REFERENCES


