


4. $u' = Bw$, \hspace{1em} \gamma = \| u' \|_2^2$.
5. $u = \frac{1}{\gamma} u'$.
6. $B' = B + 2\gamma (wu' - rw')$.

By a similar analysis we get the same perturbation bounds as for the symmetric case.

REFERENCES

Furthermore, we will more often have, \( \| A \|_2 \sim \| \bar{A} \|_2 \), so that we can drop the factor \((k + 1)^{1/2}\). Consider the second term in (187). Then, from (144) and (168), we obtain, 

\[
(190) \quad \bar{A} = A + E, \quad |E| \leq \theta_B |A| + O(\theta^2),
\]

so that,

\[
(191) \quad |\bar{A}_i - \lambda_i| \leq \theta_B \| A \|_2 + O(\theta^2) \leq (k + 1)^{1/2} \theta_B \| A \|_2 + O(\theta^2) \ll \epsilon \| A \|_2.
\]

We then conclude that,

\[
(192) \quad |\bar{A}_1 - \lambda_i| \leq \lambda_1 |(1 + \theta_B^\prime) + O(\theta^2),
\]

with \( \theta_B^\prime = (k + 1)^{1/2} \theta_B \). Finally we obtain,

\[
(193) \quad |\bar{A}_1 - \lambda_i| \leq \epsilon (k + 1)^{1/2} |\lambda_1| + O(\epsilon^2),
\]

and more often the better bound

\[
(194) \quad |\bar{A}_1 - \lambda_i| \leq \epsilon |\lambda_1| + O(\epsilon^2).
\]

These bounds are better by approximately a factor of 25 or \(25(k + 1)^{1/2}\) respectively than the corresponding bounds in Wilkinson [29] [Page 161].

**A.4. The Householder transformation for skew-symmetric matrices.** Let \( A \in \mathbb{R}^{(k+1) \times (k+1)} \) be a real skew-symmetric matrix of the form,

\[
(195) \quad A = \begin{pmatrix}
B & b \\
-b^T & 0
\end{pmatrix}, \quad B \in \mathbb{R}^{k \times k}, \quad b \in \mathbb{R}^k,
\]

and let \( P \) be the orthogonal transformation such that,

\[
(196) \quad A' = P^T A P = \begin{pmatrix}
B' & \beta \\
-\beta & 0
\end{pmatrix}, \quad P = \begin{pmatrix}
Q & 0 \\
0 & \pm 1
\end{pmatrix}, \quad P^T P = I,
\]

where \( \beta = \| b \|_2 \), and

\[
(197) \quad Q = I - 2ww^T, \quad w = \frac{v}{\| v \|_2}, \quad v = b + \text{sign}(b_k) \beta e_k,
\]

is the corresponding Householder transformation. Then,

\[
(198) \quad B' = B + 2\gamma (wu^T - uw^T),
\]

where \( \| u \|_2 = 1 \), and \( \gamma \leq \| B \|_2 \), see the discussion in Section 3. We then implement this transformation by

1. \( \beta = \| b \|_2 \).
2. \( \zeta = \| v \|_2 = \beta [2(1 + \| b_k \|)]^{1/2} \).
3. \( w = \frac{1}{\zeta} v, \quad v = b + \text{sign}(b_k) \beta e_k \).
A.2.7. Computing the vector $r$. We compute the components of the vector $r$ as follows,

$$
\dot{r}_i = f_l(\dot{u}_i - \hat{\delta}\dot{w}_i) = (\dot{u}_i - \hat{\delta}\dot{w}_i(1 + \eta_1))(1 + \eta_2)
$$
$$
= (\tau_i(1 + \eta\tau_i) - \omega_i(\delta + \eta\delta)(1 + \eta\omega_i))(1 + \eta_1))(1 + \eta_2)
$$
$$
= \tau_i + \eta\tau_i,
$$

where

$$
|\eta\tau_i| \leq \theta_\tau = \max_{1 \leq j \leq k} |\eta\tau_j|, \quad \theta_\tau \leq \theta_\tau + \theta_\omega + \theta_\omega + 2\theta + O(\theta^2) \ll \epsilon.
$$

A.2.8. Computing the the transformed matrix $B'$. We obtain the following formula for the elements of the transformed matrix,

$$
\dot{b}'_{ij} = f_l(b_{ij} - 2\tau(\dot{w}_i\dot{r}_j + \dot{r}_i\dot{w}_j))
$$
$$
= (b_{ij} - 2\tau(\dot{w}_i\dot{r}_j(1 + \eta_1) + \dot{r}_i\dot{w}_j(1 + \eta_2))(1 + \eta_3))(1 + \epsilon_1),
$$

with $|\eta_1|, |\eta_2|, |\eta_3| \leq \theta$, and $|\epsilon_1| \leq \epsilon$. Replacing $\dot{w}, \dot{r}$ by the corresponding values of $\omega$ and $\tau$ we get,

$$
\dot{b}'_{ij} = (b_{ij} - 2\tau(\tau_i\tau_j + \tau_i\omega_j + \eta_{i,j}))(1 + \epsilon_1),
$$

where,

$$
|\eta_{i,j}| \leq \theta_\tau \omega = \max_{1 \leq i, j \leq k} |\eta_{i,j}| \leq 2(\theta_\tau + \theta_\omega) + 4\theta + O(\theta^2) \ll \epsilon.
$$

Replacing $b_{ij}$ by $\bar{b}_{ij}$ we conclude that,

$$
\dot{b}'_{ij} = (\bar{b}_{ij} - 2\tau(\tau_i\tau_j + \tau_i\omega_j))(1 + \epsilon_1) - (b_{ij}\eta B_{i,j} + 2\tau\eta_{i,j}))(1 + \epsilon_1)
$$
$$
= \bar{b}_{ij}(1 + \epsilon_1) + O(\epsilon^2), \quad |\epsilon_1| \leq \epsilon.
$$

A.3. Perturbation bounds for the eigenvalues of the transformed matrix. Let us denote by $\lambda_i, i = 1, \ldots, (k + 1)$ the eigenvalues of $A$ in a decreasing order of magnitude, by $\bar{\lambda}_i$ the corresponding eigenvalues of $\bar{A}$, and by $\tilde{\lambda}_i$ the corresponding eigenvalues of $\tilde{A}$. Then,

$$
|\tilde{\lambda}_i - \lambda_i| \leq |\lambda_i - \bar{\lambda}_i| + |\bar{\lambda}_i - \lambda_i|.
$$

Consider the first term, we observe from (186) that

$$
\tilde{A} = A + \bar{A}, \quad |\bar{A}| \leq \epsilon|A| + O(\epsilon^2),
$$

where $|\cdot|$ stands for the matrix with absolute values. Then, from standard perturbation theory,

$$
|\tilde{\lambda}_i - \bar{\lambda}_i| \leq \epsilon \|A\|_2 + O(\epsilon^2) \leq \epsilon(k + 1)^{1/2}\bar{\lambda}_1 + O(\epsilon^2).
$$
Returning to the computation of \( u' \) we see that,

\[
\bar{u}' = \sum_{j=1}^{k} b_{i,j} \bar{w}_j (1 + \eta B_{i,j}) (1 + \eta \hat{u}'_i) = \sum_{j=1}^{k} (b_{i,j} (1 + \eta B_{i,j})) \bar{w}_j = \sum_{j=1}^{k} b_{i,j} \bar{w}_j,
\]

with \( \bar{b}_{i,j} = b_{i,j} (1 + \eta B_{i,j}) \), and

\[
\eta B_{i,j} \leq \theta_B = \max_{1 \leq i,j \leq k} |\eta B_{i,j}|, \quad \theta_B \leq \theta_{\bar{w}} + (1.5r + 2.5) \theta + O(\theta^2) \ll \epsilon.
\]

We can now rephrase the whole computation as follows:
We may assume that we have started with the matrix,

\[
\bar{A} = \begin{pmatrix} \bar{b} & \bar{b} \\ \bar{b} & \alpha \end{pmatrix}, \quad \bar{b} \in \mathcal{R}^{k \times k}, \quad \alpha \in \mathcal{R},
\]

and we denote by \( \bar{\beta}, \bar{\zeta}, \bar{\pi}, \bar{\gamma}, \bar{\eta}, \bar{\delta}, \bar{\tau} \) and by \( \bar{\beta}' \) the corresponding values for the transformation,

\[
\bar{A}' = \bar{P}' \bar{A} \bar{P} = \begin{pmatrix} \bar{\beta}' & \bar{\beta} \\ \bar{\beta} & \alpha \end{pmatrix}, \quad \bar{P} = \begin{pmatrix} \bar{\zeta} & 0 \\ 0 & \pm 1 \end{pmatrix}, \quad \bar{P}' \bar{P} = I.
\]

We have so far computed the values of \( \bar{\beta} \) and \( \bar{\tau} \) exactly, and \( \hat{u}'_i = \bar{u}'_i / (1 + \eta \hat{u}'_i) \).

**A.2.5. Computing the vector \( u \).** We compute the components of the vector \( u \) as follows,

\[
\hat{u}_i = fl(\frac{\hat{u}'_i}{\gamma}) = \frac{\hat{u}'_i}{\gamma} (1 + \eta_i) = \frac{\bar{w}_i}{\gamma} \frac{1 + \eta}{1 + \eta \hat{u}'_i} = \bar{u}_i (1 + \eta \bar{u}_i),
\]

with,

\[
|\eta \bar{u}_i| \leq \theta_{\bar{w}} = \max_{1 \leq j \leq k} |\eta \bar{w}_j|, \quad \theta_{\bar{w}} \leq \frac{1}{2} (r + 5) \theta + O(\theta^2) \ll \epsilon.
\]

**A.2.6. Computing \( \delta \).** We compute the value of \( \delta \) by the divide and conquer technique, so that,

\[
\hat{\delta} = fl(\sum_{i=1}^{k} \hat{\omega}_i \hat{u}_i) = \sum_{i=1}^{k} \hat{\omega}_i \hat{u}_i \prod_{j=1}^{r+1} (1 + \eta_{i,j}), \quad |\eta_{i,j}| \leq \theta, \quad r = [\log_2 k].
\]

Replacing \( \hat{\omega}_i \) by \( \bar{\omega}_i \) and \( \hat{u}_i \) by \( \bar{u}_i \) we get

\[
\hat{\delta} = \sum_{i=1}^{k} \bar{\omega}_i \bar{u}_i (1 + \eta_i) = \bar{\delta} + \eta \bar{\delta},
\]

where,

\[
|\eta \bar{\delta}| \leq \theta_{\bar{w}} = \max_{1 \leq j \leq k} |\eta_j| \leq \theta_{\bar{w}} + \theta_{\bar{w}} + (r + 1) \theta + O(\theta^2) \ll \epsilon.
\]

29
with $|\eta_{k,1}|, |\eta_{k,2}| \leq \theta$. Replacing $b_k$ by $\overline{b}_k$ and $\hat{\zeta}$ by $\overline{\zeta}$ we obtain,

$$\hat{w}_k = \text{sign}(\hat{b}_k) \frac{(|\overline{b}_k| + |\hat{\beta}|)(1 + |\overline{\eta}_{k,1}|)}{|\zeta| (1 + |\overline{\eta}_{k,2}|)(1 + \eta_{k,2})} \equiv \overline{w}_k (1 + \eta \overline{w}_k).$$

Here we have,

$$(|\overline{b}_k| + |\hat{\beta}|)(1 + |\overline{\eta}_{k,1}|) = (|b_k| + |\beta| + \eta b_k |b_k|) (1 + |\overline{\eta}_{k,1}|) = (|b_k| + |\beta|)(1 + \eta_{k,1}),$$

so that,

$$\overline{\eta}_{k,1} = \eta_{k,1} - \eta b_k \frac{|b_k|}{|b_k| + |\beta|} + O(\theta^2).$$

We conclude that for $1 \leq i \leq k$,

$$|\eta \overline{w}_i| \leq \theta \overline{w} = \max_{1 \leq j \leq k} |\eta \overline{w}_j|, \quad \theta \overline{w} \leq \theta_{\overline{b}} + \theta_{\overline{c}} + 2\theta + O(\theta^2) \ll \epsilon.$$

**A.2.4. Computing $u'$ and $\gamma$.** We compute the value of $u'$ and $\gamma$, using the divide and conquer technique as before. We first compute,

$$\hat{u}'_i = \text{fl} \left( \sum_{j=1}^{k} b_{i,j} \hat{w}_j \right) = \sum_{j=1}^{k} b_{i,j} \hat{w}_j \prod_{j=1}^{r+1} (1 + \eta_{i,j}), \quad r = \lceil \log_2 k \rceil.$$

Then, replacing $\hat{w}_j$ by $\overline{w}_j$ we get,

$$\hat{u}'_i = \sum_{j=1}^{k} b_{i,j} \overline{w}_j (1 + \eta B_{i,j}^{(1)}),$$

where,

$$|\eta B_{i,j}^{(1)}| \leq \theta \overline{w} + (r + 1)\theta + O(\theta^2) \ll \epsilon.$$

We then compute,

$$\hat{\gamma} = \text{fl} \left( \sum_{i=1}^{k} (\hat{u}'_i)^2 \right)^{1/2} = \left[ \sum_{i=1}^{k} (\hat{u}'_i)^2 \prod_{j=1}^{r+1} (1 + \eta_{i,j}) \right]^{1/2}(1 + \eta_{i})$$

$$= \left[ \sum_{i=1}^{k} (\hat{u}'_i (1 + \eta \hat{u}'_i))^2 \right]^{1/2} = \left[ \sum_{i=1}^{k} (\overline{u}_i)^2 \right]^{1/2} = \overline{\gamma},$$

with $|\eta_{i}|, |\eta_{i,j}| \leq \theta$. Hence,

$$\overline{\eta} = \hat{u}'_i (1 + \eta \hat{u}'_i), \quad |\eta \hat{u}'_i| \leq \frac{1}{2} (r + 3)\theta + O(\theta^2) \ll \epsilon.$$
A.2.2. Computing $\zeta$. We compute the value of $\zeta$ using the computed value of $\overline{\beta}$.

Hence,

\begin{equation}
(146) \quad \hat{\zeta} = fl(\overline{\beta}[2(1 + \frac{|b_k|}{\beta})]^{1/2}) = \overline{\beta}[2(1 + \frac{|b_k|}{\beta}(1 + \eta_1))(1 + \eta_2)]^{1/2}(1 + \eta_3)(1 + \eta_4),
\end{equation}

with $|\eta_i| \leq \theta$, $1 \leq i \leq 4$. Replacing $b_k$ by $\overline{b_k}$ we obtain,

\begin{equation}
(147) \quad \hat{\zeta} = \overline{\beta}[2(1 + \frac{|\overline{b_k}|}{\beta}(1 + \overline{\eta}_1))(1 + \eta_2)]^{1/2}(1 + \eta_3)(1 + \eta_4), \quad 1 + \overline{\eta}_1 = \frac{1 + \eta_1}{1 + \eta b_k},
\end{equation}

and $0 \leq |\overline{b_k}|/\beta < 1$. Now, for $0 \leq x \leq 1$ and

\begin{equation}
(148) \quad [1 + x(1 + \eta)]^{1/2} = (1 + x)^{1/2}(1 + \eta')^{1/2},
\end{equation}

we have

\begin{equation}
(149) \quad x \eta = (1 + x)\eta', \quad |\eta'| \leq \frac{x}{1 + x} |\eta| \leq \frac{1}{2} |\eta|.
\end{equation}

Therefore,

\begin{equation}
(150) \quad \hat{\zeta} = \overline{\beta}[2(1 + \frac{|\overline{b_k}|}{\beta})]^{1/2}[(1 + \overline{\eta}_1)(1 + \eta_2)]^{1/2}(1 + \eta_3)(1 + \eta_4) \quad |\overline{\eta}_1| \leq \frac{1}{2} |\overline{\eta}_1|
\end{equation}

\begin{equation}
(151) \quad \equiv \overline{\zeta}(1 + \eta \overline{\zeta}),
\end{equation}

with,

\begin{equation}
(152) \quad |\eta \overline{\zeta}| = \theta \overline{\gamma}, \quad \theta \overline{\gamma} \leq \frac{1}{4} \theta \overline{\gamma} + 2\frac{3}{4} \theta + O(\theta^2) \ll \epsilon.
\end{equation}

A.2.3. Computing $w$. We compute the first components of $w_i$, $1 \leq i < k$ by,

\begin{equation}
(153) \quad \hat{w}_i = fl(\overline{b}_k) = \frac{b_k}{\zeta}(1 + \eta_i), \quad |\eta_i| \leq \theta,
\end{equation}

\begin{equation}
(154) \quad = \frac{\overline{b}_k}{\zeta} \frac{(1 + \eta_i)}{(1 + \eta \overline{\zeta})(1 + \eta \overline{\zeta})}
\end{equation}

\begin{equation}
(155) \quad \equiv \overline{w}_i(1 + \eta \overline{w}_i).
\end{equation}

Then, for $i = k$ we have,

\begin{equation}
(156) \quad \hat{w}_k = sign(b_k) fl(\frac{|b_k| + \overline{\beta}}{\zeta})
\end{equation}

\begin{equation}
(157) \quad = sign(b_k) \frac{(|b_k| + \overline{\beta})(1 + \eta_{k,1})}{\zeta}(1 + \eta_{k,2}),
\end{equation}

\begin{equation}
(158) \quad 27
\end{equation}
6. $\delta = w^t u$.
7. $r = u - \delta w$.
8. $B' = B - 2\gamma (wr^t + rw^t)$.

in the following subsection, and discuss its numerical properties.

**A.2. Error analysis of the Householder transformation.** We assume that the input matrix is given in single precision denoted by $\epsilon$ and that the computation is performed in double precision denoted by $\theta$. Here we store the vectors $w, u, r$ in double precision, compute the transformed matrix $B'$ in double precision, but store the final resulting matrix in single precision. The additional required storage is therefore of order $k$ which is negligible as compared to $k^2$ the size of the matrix. These assumptions are very similar to those of Wilkinson \cite{29} [page 152-160], but we will obtain better bounds. We denote the computed value of say $a$ by $\hat{a} = fl(a)$, and assume that the standard operations satisfy

$$fl(a \text{ opr } b) = (a \text{ opr } b)(1 + \eta), \quad |\eta| \leq \theta.$$  

where opr stands for $+, -, \ast, /$ and $\sqrt{\cdot}$. We shall further use the following matrix norms, the spectral norm,

$$(140) \quad \| A \|_2 = \max_{\|x\|_2 = 1} \| Ax \|_2 = \max_{1 \leq i \leq k+1} |\lambda_i|,$$

and the Euclidean norm,

$$(141) \quad \| A \|_E = (\sum_{i,j=1}^{k+1} a_{ij}^2)^{1/2} = (\sum_{i=1}^{k+1} \lambda_i^4)^{1/2}.$$  

**A.2.1. Computing $\beta$.** We will compute the value of $\beta$ using the tree sum divide and conquer method, see Heller \cite{13}. For example for $k = 4$ we compute the sum as,

$$(142) \quad \beta^2 = (b_1^2 + b_2^2) + (b_3^2 + b_4^2).$$

For general $k$ we get the computed value,

$$(143) \quad \hat{\beta} = fl((\sum_{i=1}^{k} b_i^2)^{1/2}) = [\sum_{i=1}^{k} b_i^2 \prod_{j=1}^{r+1} (1 + \eta_{i,j})]^{1/2}(1 + \eta_1),$$

where $r = \lceil \log_2 k \rceil$, and $|\eta_1|, |\eta_{i,j}| \leq \theta$. Hence,

$$(144) \quad \hat{\beta} = [\sum_{i=1}^{k} (b_i (1 + \eta b_i))^2]^{1/2} \equiv [\sum_{i=1}^{k} \tilde{b}_i^2]^{1/2} = \tilde{\beta},$$

with $\tilde{b}_i = b_i (1 + \eta b_i)$, and

$$|\eta b_i| \leq \theta_\beta = \max_{1 \leq j \leq k} |\eta b_j|, \quad \theta_\beta \leq \frac{1}{2} (r + 3) \theta + O(\theta^2) \ll \epsilon.$$
obtain the corresponding eigenvalues and eigenvectors. We have further presented an
application to the complex scaled discrete variable representation method in quantum
reaction dynamics, where the major computational part involves the diagonalization of
large complex symmetric matrices. For three and higher dimensional problems, the use
of parallel supercomputers is mandatory. We have presented some of the features for
the possible implementation of the new algorithm on such modern computers but more
research is needed in this field.

Acknowledgment. The authors are grateful to Prof. Nimrod Moiseyev for numerous
discussions and for his encouragement and advice in developing the code. We would
also like to thank Dr. Jane Cullum for providing us with the subroutines CMTQL1,
and INVERM, which were used in the final stages of the algorithm. V.R. acknowledges
a support from the Center of Absorption of Science, Israel Ministry of Immigrant
Absorption and from the Wolfson Family Charitable Trust Program.
The code is available on request from the authors by email,
baron@cs.technion.ac.il, or, chr05vr@chem.technion.ac.il .

A. Appendix.

A.1. The Householder transformation. Let $A \in \mathbb{R}^{(k+1) \times (k+1)}$ be a real sym-
metric matrix of the form,

\begin{equation}
A = \begin{pmatrix} B & b \\ b^t & \alpha \end{pmatrix}, \quad B \in \mathbb{R}^{k \times k}, \quad b \in \mathbb{R}^k, \quad \alpha \in \mathbb{R},
\end{equation}

and let $P$ be the orthogonal transformation such that,

\begin{equation}
A' = P^T AP = \begin{pmatrix} B' & \beta \\ \beta & \alpha \end{pmatrix}, \quad P = \begin{pmatrix} Q & 0 \\ 0 & \pm 1 \end{pmatrix}, \quad P^T P = I,
\end{equation}

where $\beta = \| b \|_2$, and

\begin{equation}
Q = I - 2ww^t, \quad w = \frac{v}{\| v \|_2}, \quad v = b + \text{sign}(b_k) \beta \epsilon_k,
\end{equation}

is the corresponding Householder transformation. Then,

\begin{equation}
B' = B - 2\gamma (wr^t + rw^t),
\end{equation}

where $\| w \|_2 = 1$, $\| r \|_2 \leq 1$, and $\gamma \leq \| B \|_2$, see the discussion in Section 2.2. We
present the following efficient implementation

1. $\beta = \| b \|_2$.
2. $\zeta = \| v \|_2 = \beta[2(1 + \frac{|b_k|}{\beta})]^{1/2}$.
3. $w = \frac{1}{\zeta}v$, \quad $v = b + \text{sign}(b_k) \beta \epsilon_k$.
4. $u' = Bw$, \quad $\gamma = \| u' \|_2$.
5. $u = \frac{1}{\gamma}u'$. 

25
Fig. 6.1. Typical spectrum of the complex scaled Hamiltonian. Scaling parameter $\theta = 0.1$. Real and imaginary parts of eigenvalues are given in spectroscopy units, cm$^{-1}$. Arrows indicate the decay thresholds corresponding to different vibrational states of the product diatomic CO($v$), with $v = 0, 1, 2, \ldots$. Regular sequences of eigenvalues which belong to rotated continua start at the threshold points. Eigenvalues located above the $v = 0$ threshold, which do not belong to continua, represent resonance states.
imaginary parts, in accordance with equation (115) provide the lifetimes of these states. Note that due to truncation of the basis set, resonance eigenvalues depend slightly on rotation angle $\theta$ and stationary positions of the resonances with respect to $\theta$ should be found to calculate exact resonance positions and lifetimes. It means that construction and diagonalization of the Hamiltonian is supposed to be carried out many times to get solution of chemical problem and high performance of the diagonalization routine is of crucial importance in this context. Typical accuracy of the diagonalization routine

registered in the case of chemical quantum dynamics problem is illustrated by Tables 6 to 9 for one of the ray Hamiltonians (corresponding to $\alpha$ value when a number of eigenvalues which meet the energy cutoff criteria is maximal). The basis set size was taken to be $n = 720$ and $n = 1080$. Note that the most time consuming stage of solving the dynamical problem is the diagonalization of the ray Hamiltonians and preparation of the basis set for construction of the total Hamiltonian, and the speed up is about 10 (as compared to the use of the standard EISPACK routine) whereas the accuracy of calculations was kept unchanged.

7. Conclusion. We have presented in this paper a new approach for the fast and efficient diagonalization of large complex symmetric matrices. The new algorithm is fast by an order of magnitude than the EISPACK routines for this problem, and has approximately the same accuracy. The main contribution of this paper is in the tridiagonalization procedure. Once we have obtained a tridiagonal matrix we can use fast and efficient methods (probably using higher precision with relatively insignificant cost) to
below a certain energy cutoff, are included in the final, reduced ray eigenvector basis used for construction of full non-Hermitian (complex and symmetric) Hamiltonian:

\[
H_{nm}^{(\gamma')} = \epsilon_n^{(\gamma')} \delta_{nm} \delta_{\gamma\gamma'} + \left( \sqrt{\omega_n \omega_{\gamma'}} \sum_{j=0}^{N} \frac{2j+1}{2} j(j+1) P_j(\cos \alpha) P_j(\cos \alpha_{\gamma'}) \right) \times \left< \psi_n^{(\gamma')} \right| \frac{\hbar^2}{2\mu_1 r^2} + \frac{\hbar^2}{2\mu_2 R^2} \left| \psi_m^{(\gamma')} \right>.
\]

Then eigenvalues and eigenvectors of the full Hamiltonian matrix provides resonance positions, widths and eigenfunctions of resonance states. From a computational point of view the method is reduced to the problem of diagonalising complex symmetric matrices.

6.1.1. An illustrative example. An illustrative example given below represents an application of the outlined procedure to studies of the decay dynamics of the formyl radical,

\[
\text{HCO} \rightarrow \text{H} + \text{CO},
\]
on the fitted ab initio potential energy surface \( V(R, r, \alpha) \) described in ref.[14]. Without entering the details of calculation we mention that the number of the DVR grid points was taken to be \( N_\alpha = 35 \) and the number of basis function of ray Hamiltonians \( \phi_m \) providing converged results with respect to the basis set size was 1080.

Therefore, computation reduces mainly to 35 successive diagonalizations of the ray Hamiltonian matrices \( H_{nm}^{(\gamma')} \) of the size \( n = 1080 \) and construction and diagonalization of the total Hamiltonian matrix. In accordance with the used energy cutoff criteria \( E_{\text{cut}} = 0.045 \text{a.u.} \approx 9900 \text{cm}^{-1} \) the size of the total Hamiltonian matrix (see equations (131) and (132)) was found to be equal to \( N = 1967 \).

Calculated complex eigenenergies of the total Hamiltonian are presented in Figure 6.1. Without entering into the details of the problem we mention here only general properties of the spectrum (see [27] for complete discussion). First of all, in accordance with a Balslev-Combes theorem[8, 16, 17, 30, 3, 31, 21] all eigenvalues of the complex scaled Hamiltonian are located on a real axis or in a lower energy half plane. Three types of eigenvalues can be separated. Eigenvalues, which forms regular structure clearly seen in the Figure 6.1 represent 2\( \theta \) rotated continua and are not of interest from physical point of view. Fifteen eigenvalues located on a real axes below dissociation threshold of HCO (1087 cm\(^{-1}\)) represent the bound states. Their imaginary parts should be exactly equal to zero but actually they deviate from zero by about \( 10^{-3} \text{ cm}^{-1} \). This deviation is primarily conditioned by truncation of basis set used for construction the Hamiltonian. Complex eigenvalues in a lower energy half plane

\[
E = E_{\text{res}} - \frac{i}{2} \Gamma_{\text{tot}}
\]

which do not belong to the regular structure of rotated continua, represent resonance states. Their real parts correspond to energy positions of quasibound states of HCO and
Hamiltonian are determined by numerical quadrature over the DVR points. For $J = 0$, the Gauss-Legendre quadrature points serve as a grid points, and the DVR representation is isomorphic with a finite basis set representation of the same number $N_j = N_\gamma$ of the Legendre polynomials. The two representations are related by an orthogonal transformation,

$$ T_{\gamma} = w_\gamma \mathbf{P}_j(x_\gamma), $$

where $x_\gamma = \cos \alpha_\gamma$ and $w_\gamma$ are the points and weights, respectively, of the $N_\gamma$ Gauss-Legendre quadrature, and

$$ \mathbf{P}_j = [(2j + 1)/2]^{1/2} \mathbf{P}_j, $$

are the normalized Legendre polynomials. The FBR matrix representation of the Hamiltonian (117) to (119), in a normalized Legendre polynomial basis set gives

$$ [t(\tilde{R}, r)I + G(\tilde{R}, r)j^2 + V(\tilde{R}, r) - \epsilon I]f(\tilde{R}, r) = 0, $$

where

$$ j^2 = j(j + 1)\delta_{jj'}, \quad V(\tilde{R}, r)_{j,j'} = < \mathbf{P}_j | V(\tilde{R}, r, \alpha) | \mathbf{P}_{j'} >. $$

Here $\epsilon$ and $f(\tilde{R}, r)$ are the eigenvalues and eigenvectors of the Hamiltonian respectively. Operating on this with the transformation defined by (122) we get,

$$ [t(\tilde{R}, r)I + G(\tilde{R}, r)T^T j^2 T + T^T V(\tilde{R}, r) T - \epsilon I]g(\tilde{R}, r) = 0, $$

where

$$ g(\tilde{R}, r) = T^T f(\tilde{R}, r). $$

The potential in the new representation is diagonal in angles,

$$ [T^T V(\tilde{R}, r) T]_{\gamma\gamma'} \sim \delta_{\gamma\gamma'} V(\tilde{R}, r, \alpha_\gamma), $$

and the only coupling between angles enters through the matrix elements of the angular momentum $j^2$. The DVR transformation results in a set of two-dimensional cuts through the three-dimensional potential energy surface, where each cut has its own ray Hamiltonian $H^{(\gamma)}(\tilde{R}, r)$, defined by equation

$$ H^{(\gamma)}_{nm} = < \phi_n | - \frac{\hbar}{2\mu_1} \frac{\partial^2}{\partial r^2} - \frac{\hbar^2}{2\mu_2} \frac{\partial^2}{\partial \tilde{R}^2} + V(\tilde{R}, r, \alpha_\gamma) | \phi_m >. $$

Here $| \phi_n >$, are basis functions depending on $r$ and reaction coordinate $\tilde{R}$ complex scaled according to (121). Eigenvectors $\psi^{(\gamma)}_n$ with eigenvalues $\epsilon^{(\gamma)}_n$ of the ray Hamiltonians

$$ H^{(\gamma)} \psi^{(\gamma)}_n = \epsilon^{(\gamma)}_n \psi^{(\gamma)}_n, $$

where $\tilde{R}$ and $\alpha_\gamma$ are the normalized reaction coordinate and reaction angle, respectively.
6.1. A n overview. We review in this Subsection one of the most promising quantum method that allows us to study chemical reaction dynamics of small polyatomic systems. The complex coordinate method (CCM)\cite{8, 16, 17, 30, 3, 31, 21} is an extension of quantum mechanics to non-Hermitian Hamiltonians, which is aimed at the efficient treatment of metastable quasibound resonance states. The complex scaled Hamiltonian is non-Hermitian and has complex eigenvalues. Upon complex scaling, eigenfunctions associated with either bound or resonance states become square integrable. The real part of the eigenvalues, $E_{\text{res}}$, corresponding to resonance states provides the energy position of these states and the imaginary part provides the resonance lifetime, $\tau$ (inverse width),

\begin{equation}
\tau = \hbar/\Gamma_{\text{tot}}, \quad \Gamma_{\text{tot}} = -2\text{Im}(E_{\text{res}}),
\end{equation}

where $\Gamma_{\text{tot}}$ denotes the total width of the resonance states. When using a finite basis set, resonance eigenvalues are evaluated at stationary points with respect to scaling parameter $\theta$, whereas eigenvalues describing the continuum are rotated into the complex state by an angle of $2\theta$. When treating three-dimensional quantum reactive problems (vibrational predissociation of triatomic species\cite{20}) the complex coordinate method have been used in combination with discrete variable representation (DVR) method developed a few years ago\cite{33, 32}. It is accepted to use Jacobi body fixed coordinates $R$, $r$ and $\alpha$ to describe a decay reaction

\begin{equation}
\text{ABC} \rightarrow \text{A} + \text{BC}.
\end{equation}

Here $R$ is a distance from atom $A$ to the center of mass $BC$, $r$ represents the $BC$ bond length and $\alpha$ is the angle enclosed by $R$ and $r$. Working in body fixed coordinates, the Hamiltonian is $(J = 0)$

\begin{align}
\hat{H}(\tilde{R}, r, \alpha) &= -\frac{\hbar^2}{2\mu_1} \frac{\partial^2}{\partial r^2} - \frac{\hbar^2}{2\mu_2} \frac{\partial^2}{\partial \tilde{R}^2} \\
&+ j^2 \left( \frac{1}{2\mu_1 r^2} + \frac{1}{2\mu_2 \tilde{R}^2} \right) + V(\tilde{R}, r, \alpha) \\
&= t(\tilde{R}, r) + G(\tilde{R}, r)j^2 + V(\tilde{R}, r, \alpha),
\end{align}

where

\begin{equation}
\mu_1 = \frac{m_B m_C}{m_A + m_C}, \quad \mu_2 = \frac{m_A m_{BC}}{m_A + m_{BC}},
\end{equation}

and $\tilde{R}$ is the reaction coordinate rotated into the complex plane, according to

\begin{equation}
\tilde{R} = (R - R_0) \exp(i\theta) + R_0
\end{equation}

and $R_0$ is the point around which complex rotation is carried out. The DVR is a grid representation of $N_\alpha$ points for the angular coordinate, which is isomorphic with an approximate finite basis representation (FBR) in which the matrix elements of the
consider the stability of the algorithm. Here, we denote the sequence of transformed matrices by,

\[ \hat{H}^{(t+1)} = (\hat{U}^{(t)})^T \hat{H}^{(t)} \hat{U}^{(t)} + E^{(t)}, \quad |E^{(t)}| \leq \epsilon |\hat{H}^{(t)}|, \]

see Appendix (A.3). Hence, the computed tridiagonal matrix is similar to the perturbed matrix,

\[ \hat{T} = V^T (H + E) V, \quad E = \sum_{t=1}^{m} V^{(t)} E^{(t)} (V^{(t)})^T, \quad V^{(t)} = \hat{U}^{(t)} \ldots \hat{U}^{(1)}, \]

where \( m = O(n) \). We will say that the algorithm is numerically stable if \( \| E \|_F = O(\epsilon n \| H \|_F) \), where we assume that the spectrum of \( H \) is insensitive to such perturbations (for otherwise the original problem is not well posed, and we have to increase the precision). We note however, that a reasonable measure for this norm can be deduced from comparing \( \| \hat{H}^{(t)} \|_F \) with \( \| H \|_F \). A relatively small growth implies \( \| E^{(t)} \|_F \sim \epsilon \| H \|_F \), and also quite certainly \( \| V^{(t)} \|_F = O(1) \). Here, we observe that the Householder transformations obey to this rule, and so do the exceptional transformations, assuming their number is relatively small. It then remains to consider the complex orthogonal transformations \( V_C \). Theoretically, it seems that they may lead to an exponential growth in the elements of the next to last current row. However, this is very unlikely from the following reason. An increase in the size of the elements of the next to last row also implies that \( |x + iy| \gg |z| \), and hence to a break in that growth. The accumulative effect is then only of an additive nature. In practice the norm growth was found to be bounded by a small constant, in our examples by \( 3 \), see Table 2. The difference between the computed eigenvalues and CG was then relatively small. We have next computed the eigenvectors of the tridiagonal matrix using the subroutine INVERM from the Lanczos package[4], which is a complex version of the inverse iteration method for real symmetric tridiagonal matrices. The eigenvectors of the original matrix were then computed by back transformation. The performance of the new algorithm as compared to the corresponding EISPACK routines is depicted in Tables 3 - 5. The advantage of having a tridiagonal matrix is then readily seen. The running time for computing the eigenvectors of the tridiagonal matrix is relatively insignificant, whereas for the Hessenberg matrix it is of the order of magnitude of the eigenvalue computation. The condition of a pair of eigenelements, \( \lambda, v \), that is \( \| Hv - \lambda v \|_2 \), was found to be approximately the same. The computed eigenvectors were found to be better orthogonal to each other, an advantage of using the complex orthogonal rotations over the unitary ones. We finally comment on the possible implementation of the new algorithm on vector and parallel supercomputers. The reduction stage is rich in vector and matrix operations, as the algorithm for complex Hermitian matrices (with slight variations for the exceptional transformations). The eigenvalues and eigenvectors can then be computed independently and in parallel by several processors as in the real symmetric case see Bar-On[2].

6. Complex scaled discrete variable representation method for quantum chemical reactive problems.
### Table 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>25.4</td>
<td>2.87</td>
<td>10.5</td>
<td>9.8</td>
<td>217.8</td>
<td>6.2</td>
<td>5.5d-12</td>
</tr>
<tr>
<td>1000</td>
<td>222.0</td>
<td>2.15</td>
<td>10.14</td>
<td>38.2</td>
<td>2298.0</td>
<td>8.8</td>
<td>1.2d-12</td>
</tr>
<tr>
<td>2000</td>
<td>1774.0</td>
<td>2.65</td>
<td>20.2</td>
<td>153.0</td>
<td>19876.3</td>
<td>10.3</td>
<td>6.5d-12</td>
</tr>
</tbody>
</table>

Eigenvalues computation, in seconds. Columns: n - matrix size, Red. - reduction time, Grw. - $\| T \|_F$ ($\| H \|_F = 1$), Brk. - threshold, # breakdowns, Diag. - diagonalization time, CG - EISPACK time, Spd. - Speed up, Acc. - maximum difference between computed eigenvalues and CG.

### Table 3

<table>
<thead>
<tr>
<th></th>
<th>Red.</th>
<th>Full</th>
<th>Cond.</th>
<th>Orthog.</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>3.1</td>
<td>14.3</td>
<td>4.4d-12</td>
<td>4.6d-14</td>
</tr>
<tr>
<td>EISP.</td>
<td>41.0</td>
<td>21.7</td>
<td>2.6d-13</td>
<td>4.2d-12</td>
</tr>
</tbody>
</table>

Eigenvectors computation for $n = 500$ and $k = n/4$ vectors. Columns: Red. - computing time for the reduced matrix, Full - restore time, Cond. - $\| Hv - \lambda v \|_2$, Orthog. - $\max \| v_i^T v_j \| / \| v_i \|_2 \| v_j \|_2$.

### Table 4

<table>
<thead>
<tr>
<th></th>
<th>Red.</th>
<th>Full</th>
<th>Cond.</th>
<th>Orthog.</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>12.5</td>
<td>123.8</td>
<td>1.4d-12</td>
<td>8.0d-14</td>
</tr>
<tr>
<td>EISP.</td>
<td>568.6</td>
<td>90.9</td>
<td>3.2d-13</td>
<td>4.5d-12</td>
</tr>
</tbody>
</table>

Eigenvectors computation for $n = 1000$ and $k = n/4$ vectors.

### Table 5

<table>
<thead>
<tr>
<th></th>
<th>Red.</th>
<th>Full</th>
<th>Cond.</th>
<th>Orthog.</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>51.3</td>
<td>984.7</td>
<td>2.6d-12</td>
<td>2.9d-13</td>
</tr>
<tr>
<td>EISP.</td>
<td>5274.8</td>
<td>748.5</td>
<td>2.3d-13</td>
<td>6.1d-12</td>
</tr>
</tbody>
</table>

Eigenvectors computation for $n = 2000$ and $k = n/4$ vectors.
We then apply a unitary complex Hermitian transformation

\[ Q = \begin{pmatrix} \bar{Q} \\ I_{n-k} \end{pmatrix}, \quad \bar{Q} = I - 2ww^*, \quad w = \frac{1}{\| v \|_2}, \]

with \( v = h \pm \| h \|_2 \epsilon_k \). Hence,

\[ H'_1 = \bar{Q}^* H_1 \bar{Q} = H_1 - 2\gamma(wu_1^* + u_2w^* - 2\delta w^*), \]

where,

\[ u_1 = \frac{1}{\gamma} u'_1, \quad u_2 = \frac{1}{\gamma} u'_2, \quad \delta = \frac{1}{\gamma} w^* H_1 w, \]

and,

\[ u'_1 = H'_1 w, \quad u'_2 = H_1 w, \quad \gamma = (\| u'_1 \|_2^2 + \| u'_2 \|_2^2)^{\frac{1}{2}}. \]

We finally get,

\[ H'_1 = H_1 - 2\gamma(\omega \omega^* + \omega \omega^*), \quad r = u_1 - \delta w, \quad s = u_2 - \delta w, \]

so the complexity is dominated by \( 4k^2 \) complex additions and multiplications. We then compute,

\[ G' = G \bar{Q} = G - 2(Gw)w^* = G - 2\gamma hw^*, \]

where

\[ \gamma = \| Gw \|_2, \quad h' = Gw, \quad h = \frac{1}{\gamma} h'. \]

This requires an additional \( 2k(n-k) \) complex additions and multiplications. The total complexity of the reductions stage for the non symmetric algorithm is then

\[ T(n) = \sum_{k=n-1}^{n-1} 4k^2 + 2k(n-k) \sim 2 \sum_{k=n}^{n-1} k^3 + n^3 \sim \frac{5}{3} n^3, \]

complex additions and multiplications. Hence, the reduction to a tridiagonal form is asymptotically 2.5 times faster than for non symmetric matrices. Moreover, the diagonalization of the tridiagonal matrix is relatively insignificant whereas it is asymptotically a process of order \( n^3 \) for Hessenberg matrices. Hence, the new algorithm is faster than that. We have compared the running time of the new algorithm with that of the corresponding EISPACK routine CG on the DEC AXP 3000-500 machine, under UNIX OSF/1 operating system, see Table 2. The number of breakdowns for \( \| V_C \|_2 \leq 10, 20 \), is seen to be small and is relatively insignificant in terms of running time. The final diagonalization of the complex symmetric tridiagonal matrix was done using a complex version of the QL algorithm, subroutine CMTQL1 from the Lanczos package [4]. The computation of the eigenvalues is seen to be asymptotically ten times faster. We next
5.2.2. Breakdown in the middle of a block. We may denote the tail of the matrix by the complex symmetric tridiagonal matrix,

\[
T^{(t)}_k = \begin{pmatrix}
\alpha^{(t)}_k & \beta^{(t)}_k & & \\
\beta^{(t)}_k & \alpha^{(t)}_{k+1} & \ddots & \\
& \ddots & \ddots & \beta^{(t)}_n \\
& & \beta^{(t)}_n & \alpha^{(t)}_n
\end{pmatrix}.
\]

Here, we begin with the diagonalization stage by applying to \( T^{(t)}_k \) one sweep of the complex variant of the QL algorithm. Hence, we obtain,

\[
H^{(t+1)} = \begin{pmatrix}
H^{(t+1)}_{k+1} & \beta^{(t+1)}_{k+2} & & \\
\beta^{(t+1)}_{k+2} & H^{(t+1)}_{k+2} & \ddots & \\
& \ddots & \ddots & \beta^{(t+1)}_n \\
& & \beta^{(t+1)}_n & H^{(t+1)}_n
\end{pmatrix},
\]

with,

\[
H^{(t+1)}_{k+1} = \begin{pmatrix}
H^{(t+1)}_{k-1} & -c_i h & s_i h \\
-c_i h^t & H^{(t+1)}_{k} & \beta^{(t+1)}_{k+1} \\
s_i h^t & \beta^{(t+1)}_{k+1} & H^{(t+1)}_{k+1}
\end{pmatrix}
\]

where,

\[
Q_i = \begin{pmatrix}
-c_i & s_i \\
s_i & c_i
\end{pmatrix}, \quad i = (n-1), \ldots, k
\]

corresponds to the successive orthogonal transformations of the QL algorithm. Note, that we have actually retreated one step backwards in the tridiagonalization stage, but this only at the cost of \( O(n) \) time. By choosing an appropriate shift, we can make sure that \( |\beta^{(t+1)}_{k+1}| \) is large enough so that the next transformation will succeed. In practice, we have found that a zero shift was very effective. The number of breakdowns was relatively small and recovery was immediate. We will describe these experiments in the following.

5.3. Analysis and Numerical examples. We analyze the complexity of the new reduction algorithm assuming that the number of breakdowns is relatively small, as it is in practice. Hence, the asymptotic complexity is the same as for complex Hermitian matrices (since the complexity of the exceptional transformations is of order \( n^2 \)). We next consider the complexity of the corresponding reduction algorithm for non-symmetric matrices. Here, at the \( k \)th step we have

\[
H = \begin{pmatrix}
H_1 & h & 0 \\
G & H_2
\end{pmatrix}, \quad \begin{array}{c}
H_1 \in \mathbb{C}^{k \times k}, \\
H_2 \in \mathbb{C}^{(n-k) \times (n-k)}, \\
G \in \mathbb{C}^{(n-k) \times k}, \\
h \in \mathbb{C}^k.
\end{array}
\]
where $Q = Q_t Q_R$, and $\tau$ is the given threshold. Note, that we have omitted the subscript $(t)$ for simplicity. For example, if $\| v \|_2 = 1$ and $\tau = 0.9998$ then $|v^t v| \leq 0.02$.

We then try to find some permutation for which the algorithm could proceed. Here, we start with the next to last column $w$ and proceed upwards, permuting the “good” column with the last one when it is found. Usually this would suffice. However, in the case that we find that we need to examine too many columns we will try instead to apply a Jacobi rotation between two rows and columns. For example, let us apply a Jacobi rotation to the last two rows and columns, with,

$$
P = \begin{pmatrix} I_{k-2} & J \end{pmatrix}, \quad J = \begin{pmatrix} -c & s \\ s & c \end{pmatrix}, \quad c = \cos(\theta), \quad s = \sin(\theta).
$$

Then,

$$
H'_k = P^T H_k P = \begin{pmatrix} H_{k-2} & \tilde{w} \\ \tilde{w}^t & \tilde{\alpha}_{k-1} \end{pmatrix} \begin{pmatrix} \tilde{w} \\ \tilde{v} \end{pmatrix}, \quad \tilde{v} = \begin{pmatrix} \tilde{v} \\ \tilde{\alpha}_k \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} \tilde{w} \\ \tilde{\alpha}_k \end{pmatrix}, \quad \tilde{b}_k = \sin(2\theta)\alpha - \cos(2\theta) b_k, \quad \alpha = \frac{1}{2}(\alpha_k - \alpha_{k-1}).
$$

We further let,

$$
u = \begin{cases} \hat{v} & |\hat{v}^t \hat{v}| \geq |\hat{w}^t \hat{w}|, \\ \hat{w} & |\hat{v}^t \hat{v}| < |\hat{w}^t \hat{w}|. \end{cases}
$$

We consider two options. In the first, $|b_k|$ is relatively large, and we take $\tan(2\theta) = b_k/\alpha$. Then, $\tilde{b}_k = 0$ and

$$
|u^t u| \geq \frac{1}{2} |\tilde{w}^t \tilde{w} + \tilde{v}^t \tilde{v}| = |\tilde{b}_k^2 - \frac{1}{2}(\tilde{w}^t w + \tilde{v}^t v)| \sim |b_k|^2.
$$

Hence, the transformation is likely to succeed. Here we tacitly assume that $\tan^2(2\theta)$ is away from $-1$. For example, $|1 + \tan^2(2\theta)| > \frac{1}{15}$ so that $|\cos(\theta)|, |\sin(\theta)| \leq 2$. The other option is to take $\theta = \pi/4$, so that $\tilde{b}_k = \alpha$. This transformation is likely to succeed in the following cases.

- When $|b_k|$ is relatively large and $\alpha \sim \pm ib_k$, (the case $\tan^2(2\theta) \sim -1$) since

  $$
  |u^t u| \geq (b_k^2 - \tilde{b}_k^2) - \frac{1}{2}(v^t v + w^t w) | \sim 2|b_k|^2.
  $$

- When $\alpha$ is relatively large since,

  $$
  |u^t u| \geq (\alpha^2 - b_k^2) + \frac{1}{2}(v^t v + w^t w) | \sim |\alpha|^2.
  $$

- When $|\bar{w}^t \bar{v}|$ is relatively large since $|u^t u| \geq |\bar{w}^t \bar{v}|$. 


Applying this transformation to $H^{(R)}$ we get

$$H^{(t+1)}_k = \begin{pmatrix} H^{(t+1)}_{k-1} & \pi \tau \\ \bar{w} \tau & G \end{pmatrix} = \begin{pmatrix} H^{(t+1)}_{k-1} & \beta_k^{(t+1)} \\ \beta_k \alpha_k & \alpha_k^{(t+1)} \end{pmatrix}.$$}

We will proceed in the next section to consider the more exceptional transformations.

5.2. The exceptional transformations. We present in this subsection the more exceptional but essential transformations that will make the algorithm work in the general case. We observe that,

$$V_C = \frac{1}{s_k} \begin{pmatrix} x + iy & z \\ -z & x + iy \end{pmatrix} = \frac{1}{s_k} XDX^*, \quad X = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix},$$

with $D = diag(x + i(y + z), x + i(y - z))$, and $X^*X = I$. Hence,

$$\|V_C\|_2^2 = \frac{x^2 + (|y| + |z|)^2}{|s_k|^2} = \frac{x^2 + y^2 + z^2 + 2|yz|}{(x^2 - y^2 + z^2)^2 + (2xy)^2},$$

$$\|V_C\|_2^2 = \frac{1 + \gamma}{(1 - \gamma^2)^2} = \left(\frac{1 + \gamma}{1 - \gamma}\right)^2, \quad \gamma = \frac{2yz}{x^2 + y^2 + z^2} \leq 1,$$

so that,

$$\|V_C\|_2 = \left(\frac{1 + \gamma}{1 - \gamma}\right)^{1/2}.$$}

We see that as $\gamma \to 1$, the norm of $V_C$ approaches infinity, and thus it may quite certainly increase the Euclidean norm of the matrix leading later to several numerical problems. A choice of $\|V_C\|_2 \leq 10$ that is $\gamma < \tau = 0.9998$, was found to be effective in the sense that it has resulted in few breakdowns, and in good numerical accuracy. We will assume in general that some threshold $\tau$ is given, and when it does not hold we will apply one of the following transformations.

5.2.1. The case of $\beta_{k+1}^{(t)} = 0$. Here, $H_k$ is an independent block,

$$H_k = \begin{pmatrix} H^{(t+1)}_{k-2} & \pi \tau \\ \bar{w} \tau & \alpha_{k-1} \beta_k \alpha_k \end{pmatrix}, \quad v = \begin{pmatrix} \pi \tau \\ \beta_k \alpha_k \end{pmatrix}, \quad w = \begin{pmatrix} \pi \tau \\ \beta_k \alpha_k \end{pmatrix},$$

and,

$$Q^Tv = \begin{pmatrix} 0 \\ z \\ x + iy \end{pmatrix}, \quad |v^Tv| = \|v\|_2^2 (1 - \gamma^2)^{1/2}, \quad \gamma > \tau,$$
5.1.2. The form of $Q_R$. We denote similarly the transformed sub matrix from
the previous step by,

$H^{(l)} = A^{(l)} + iB^{(l)}$, \quad $\left( A^{(l)} \right)^T = A^{(l)}$, \quad $\left( B^{(l)} \right)^T = B^{(l)}$,

with,

$A^{(l)} = \begin{pmatrix} A_{k-2}^{(l)} & \hat{a} & -\bar{a}^T \\ \bar{a}^T & \alpha' & x \\ -\bar{a} & x & \alpha \end{pmatrix}$, \quad $B^{(l)} = \begin{pmatrix} B_{k-2}^{(l)} & \hat{b} & \beta' \\ \bar{b}' & \beta & y \\ y & y & \beta \end{pmatrix}$.

Then, the transformation $Q_R$ will reduce the last row and column of $A^{(l)}$,

$A^{(R)} = Q_R^T A^{(l)} Q_R = \begin{pmatrix} A_{k-2}^{(R)} & \hat{a} & z \\ \bar{a}^T & \alpha' & x \\ z & x & \alpha \end{pmatrix}$, \quad $z = \pm \| \bar{a}^T \|_2$.

Here, again, the matrix $Q_R$ is defined by the Householder transformation,

$Q_R = \begin{pmatrix} \bar{Q}_R & 1 \\ 1 & 1 \end{pmatrix}$, \quad $\bar{Q}_R = I - 2w_Rw_R^T$, \quad $w_R = \frac{1}{\| v_R \|_2} v_R$,

where $v_R = \bar{a}^T + \text{sign}(\bar{a}^T_{k-2}) \| \bar{a}^T \|_2 e_{k-2}$, and

$A_{k-2}^{(R)} = \bar{Q}_R A_{k-2}^{(l)} \bar{Q}_R$, \quad $\hat{a} = \bar{Q}_R \bar{a}$.

We then complete the transformation by applying it to $B^{(l)}$ as well yielding,

$B^{(R)} = Q_R^T B^{(l)} Q_R = \begin{pmatrix} B_{k-2}^{(R)} & \hat{b} & \beta' \\ \bar{b}' & \beta & y \\ y & y & \beta \end{pmatrix}$, \quad $B_{k-2}^{(R)} = \bar{Q}_R B_{k-2}^{(l)} \bar{Q}_R$, \quad $\hat{b} = \bar{Q}_R \bar{b}$.

5.1.3. The form of $Q_C$. We finally denote the transformed sub matrix by,

$H^{(R)} = \begin{pmatrix} H_{k-3}^{(R)} & w & v \\ w^T & G & z \\ v^T & x + iy & \alpha_k^{(l)} \end{pmatrix}$, \quad $v, w \in C^{k-3}$, \quad $G \in C^{2 \times 2}$,

where we assume that,

$s_k^2 = (x + iy)^2 + z^2 \neq 0$.

Then we define the transformation $U_C^T$ by

$U_C^T = \begin{pmatrix} I_{k-3} & V_C^T \\ V_C^T & 1 \end{pmatrix}$, \quad $V_C^T = \frac{1}{s_k} \begin{pmatrix} x + iy & -z \\ z & x + iy \end{pmatrix}$.
as in the complex Hermitian case. Here,

\[(Q_I^{(t)})^T Q_I^{(t)} = I = (Q_R^{(t)})^T Q_R^{(t)},\]

are real orthogonal matrices, and

\[(70) \quad U_C^{(t)} = \begin{pmatrix} I_{k-3} & V_C^{(t)} \\ V_C^{(t)} & 1 \end{pmatrix}, \quad V_C^{(t)} \in \mathbb{C}^{2 \times 2}, \quad (V_C^{(t)})^T V_C^{(t)} = I,\]

is complex orthogonal.

**5.1. The general form of** \(U^{(t)}\). We will describe in this section the more common form of \(U^{(t)}\) as described above. For simplicity, we omit subscripts when possible.

**5.1.1. The form of** \(Q_I\). Let us denote the complex symmetric sub matrix \(H_k^{(t)}\) at step \(t\) by,

\[(71) \quad H = A + iB, \quad A^T = A, \quad B^T = B, \quad A, B \in \mathbb{R}^{k \times k},\]

with,

\[(72) \quad A = \begin{pmatrix} A \k_{l-1} & a \\ a^t & \alpha \end{pmatrix}, \quad B = \begin{pmatrix} B \k_{l-1} & b \\ b^t & \beta \end{pmatrix}, \quad A_{k-1}, B_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)}, \quad a, b \in \mathbb{R}^{(k-1)}.
\]

Then, the transformation \(Q_I\) will reduce the last row and column of \(B\),

\[(73) \quad B^{(l)} = Q_I^T B Q_I = \begin{pmatrix} B_{k-1}^{(l)} & y \\ y & \beta \end{pmatrix}, \quad y = \pm \| b \|_2,\]

using the Householder transformation,

\[(74) \quad Q_I = \begin{pmatrix} \overline{Q}_I & 0 \\ 0 & 1 \end{pmatrix}, \quad \overline{Q}_I = I - 2w_I w_I^t, \quad w_I = \frac{1}{\| v_I \|_2} v_I,\]

where, \(v_I = b + \text{sign}(b_{k-1}) \| b \|_2 e_{k-1}\). Hence, \(B_{k-1}^{(l)} = \overline{Q}_I^T B_{k-1} \overline{Q}_I\). We complete the transformation by applying it to \(A\) as well yielding,

\[(75) \quad A^{(l)} = Q_I^T A Q_I = \begin{pmatrix} A_{k-1}^{(l)} & \overline{a} \\ \overline{a}^t & \overline{\alpha} \end{pmatrix}, \quad A_{k-1}^{(l)} = \overline{Q}_I^T A_{k-1} \overline{Q}_I, \quad \overline{a} = \overline{Q}_I a.\]
This can then be simplified to,

$$A' = A - 2\theta(UW^T + WU^T), \quad U, W \in \mathcal{R}^{k \times 2},$$

where for $A$ symmetric

$$U = (w_R, v), \quad W = (g - \alpha w_R - \beta v, h - \beta w_R - \gamma v),$$

and for $A$ skew symmetric

$$U = (w_R, v), \quad W = (g, h).$$

This form better exploits the architecture of modern supercomputers see [12, 1].

5. A new fast algorithm for reducing a complex symmetric matrix to a tridiagonal form. We present in this section the new algorithm for reducing a complex symmetric matrix $H \in \mathcal{C}^{n \times n}, H^T = H$, to a tridiagonal form. The algorithm proceeds as follows:

$$H^{(t+1)} = (U^{(t)})^T H^{(t)} U^{(t)}, \quad t = 1, 2, \ldots,$$

with $H^{(1)} \equiv H$. Let,

$$H^{(t)} = \begin{pmatrix} H^{(t)}_{k, k} & \beta^{(t)}_{k+1} & \cdots & \beta^{(t)}_{n} \\ \beta^{(t)}_{k+1} & \alpha^{(t)}_{k+1} & \cdots & \beta^{(t)}_{n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{(t)}_{n} & \cdots & \cdots & \alpha^{(t)}_{n} \end{pmatrix}, \quad H^{(t)}_k \in \mathcal{C}^{k \times k}, \quad \beta^{(t)}_j, \alpha^{(t)}_j \in \mathcal{C}, \quad j = k + 1, \ldots, n, \quad k \geq n - t + 1$$

with,

$$H^{(t)}_k = \begin{pmatrix} H^{(t)}_{k-1} & h \\ h^T & \alpha^{(t)}_k \end{pmatrix}, \quad H^{(t)}_{k-1} \in \mathcal{C}^{(k-1) \times (k-1)}, \quad h \in \mathcal{C}^{(k-1)}.$$

We will then try to reduce the $k$th row and column of $H^{(t)}_k$ so that,

$$H^{(t+1)} = \begin{pmatrix} H^{(t+1)}_{k-1} & \beta^{(t+1)}_k & \cdots & \beta^{(t+1)}_n \\ \beta^{(t+1)}_k & \alpha^{(t+1)}_k & \cdots & \beta^{(t+1)}_n \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{(t+1)}_n & \cdots & \cdots & \alpha^{(t+1)}_n \end{pmatrix},$$

using three transformations,

$$U^{(t)} = Q^{(t)}_l Q^{(t)}_R U^{(t)}, \quad l = 1, 2, \ldots,$$
We next consider the complexity of the new algorithm as compared to the standard Hermitian analog of the real symmetric algorithm. The standard reduction algorithm requires asymptotically \( \frac{2}{3}n^3 \) complex additions and multiplications, equivalent to \( \frac{8}{3}n^3 \) such real operations. Hence, theoretically, the two methods have the same asymptotic complexity. We have compared the running times of the new algorithm with the corresponding EISPACK routine HTRIDI[24] on the DEC AXP 3000-500 machine, under the UNIX OSF/1 operating system, in Table (1). We observe that our code is much more efficient, and asymptotically 2.5 time faster. We note that the EISPACK routine is implemented using real operations as ours. We further observe that the running time for the final diagonalization of the tridiagonal matrix (the EISPACK routine TQLRAT) is relatively insignificant.

We finally consider the vectorization features of the new algorithm. Here, the application of the two successive orthogonal transformations \( Q_I \) and \( Q_R \) can be affected by one block Householder transformation as follows:

\[
Q^T = Q_R^T Q_I^T = (I - 2w_R w_I^t)(I - 2v_R v_I^t)
\]

\[
= I - 2(w_R w_I^t + w_I w_R^t - 2\delta w_R w_I^t)
\]

\[
= I - 2(w_R u^t + v w_I^t),
\]

where we have defined,

\[
u = w_R - \delta w_I, \quad v = w_I - \delta w_R, \quad \delta = w_R^t w_I,
\]

and \( |\delta| \leq 1 \). Hence, \( A' = Q^T A Q \) is given by,

\[
A' = A - 2\theta (w_R g' + v h' + g w_I^t + h v_I^t)
\]

\[
-2(\alpha w_R w_I^t + \beta w_R v^t + \gamma v w_I^t + \gamma v v_I^t),
\]

with,

\[
h = \frac{1}{\theta} h', \quad g = \frac{1}{\theta} g', \quad \theta = (\| h' \|_2^2 + \| g' \|_2^2)^{\frac{1}{2}},
\]

and

\[
h' =Aw_I, \quad g' = Au = Aw_R - \delta h',
\]

\[
\alpha = \frac{1}{\theta} w_I^t Au, \quad \beta = \frac{1}{\theta} v^t Aw_I, \quad \gamma = \frac{1}{\theta} w_I^t Aw_I.
\]
Here, the matrix $Q_R$ is defined by the Householder transformation,

$$Q_R = \begin{pmatrix} \overline{Q_R} & 1 \\ -1 \\ 1 \end{pmatrix}, \quad \overline{Q_R} = I - 2w_R w_R^t, \quad w_R = \frac{1}{\|v_R\|_2} v_R,$$

where $v_R = \overline{a} + \text{sign}(\overline{a}) \|\overline{a}\|_2 e_{k-2}$, and

$$A_{k-2}^{(R)} = \overline{Q_R} A_{k-2} Q_R, \quad \hat{a} = \overline{Q_R} \overline{a}.$$

We then complete the transformation by applying it to $B^{(I)}$ as well yielding,

$$B^{(R)} = \overline{Q_R} B^{(I)} Q_R = \begin{pmatrix} B_{k-2}^{(R)} & \hat{b} \\ -\hat{b}^t & 0 & y \\ -y & 0 \end{pmatrix}, \quad B_{k-2}^{(R)} = \overline{Q_R} B_{k-2}^{(I)} Q_R, \quad \hat{b} = \overline{Q_R} \hat{b}.$$

### 4.1.3. The form of $Q_C$.

Let us denote the transformed sub matrix by,

$$H^{(R)} = \begin{pmatrix} H_{k-3}^{(R)} & w & v & 0 \\ w^* & G & z \\ v^* & x + iy \\ 0 & z & x - iy & \alpha \end{pmatrix}, \quad v, w \in \mathbb{C}^{k-3}, \quad G \in \mathbb{C}^{2 \times 2}.$$

Then we define the transformation $U_C^*$ by

$$U_C^* = \begin{pmatrix} I_{k-3} & V_C^* \\ \overline{V_C^*} & 1 \end{pmatrix}, \quad V_C^* = \frac{1}{\beta} \begin{pmatrix} -(x + iy) \\ z \\ z \end{pmatrix},$$

with $\beta^2 = x^2 + y^2 + z^2$. Applying this transformation to $H^{(R)}$ we get

$$H_k^{(k)} = \begin{pmatrix} H_{k-3}^{(k)} & w & v \\ w^* & G & z \\ v^* & x + iy \\ \overline{G} & z & x - iy & \alpha \end{pmatrix} = \begin{pmatrix} H_{k-1}^{(k)} & \beta_k \\ \beta_k & \alpha_k \end{pmatrix},$$

as required.

### 4.2. Analysis and Numerical examples.

The application of the real orthogonal transformations $Q_I, Q_R$ to the symmetric and to the skew-symmetric parts of the matrix is numerically stable see the analysis in Appendix (A.2) and (A.4), where we present new improved perturbation bounds for the Householder transformation. The transformation $V_C$ is a unitary transformation applied only to the last two rows and columns of the matrix. Hence, the rounding errors resulting in this case are small. We conclude therefore that the algorithm is numerically stable.
4.1. The complex Hermitian transformation $U$.

4.1.1. The form of $Q_I$. Let us denote the sub matrix of (37) by

$$H = A + iB, \quad A^T = A, \quad B^T = -B, \quad A, B \in \mathbb{R}^{k \times k},$$

with,

$$A = \begin{pmatrix} A_{k-1} & a \\ a' & \alpha \end{pmatrix}, \quad B = \begin{pmatrix} B_{k-1} & b \\ -b' & 0 \end{pmatrix}, \quad A_{k-1}, B_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)},$$

$$a, b \in \mathbb{R}^{(k-1)}.$$

Then, the transformation $Q_I$ will reduce the last row and column of $B$,

$$B^{(I)} = Q_I^T B Q_I = \begin{pmatrix} B_{k-1}^{(I)} & y \\ -y & 0 \end{pmatrix}, \quad y = \pm \| b \|_2,$$

using the Householder transformation,

$$Q_I = \begin{pmatrix} \overline{Q}_I & 0 \\ 0 & 1 \end{pmatrix}, \quad \overline{Q}_I = I - 2w_I w_I^t, \quad w_I = \frac{1}{\| v_I \|_2} v_I,$$

where $v_I = b + \text{sign}(b_{k-1}) \| b \|_2 e_{k-1}$. Hence, $B^{(I)}_{k-1} = \overline{Q}^T_I B_{k-1} \overline{Q}_I$. We complete the transformation by applying it to $A$ as well yielding,

$$A^{(I)} = Q_I^T A Q_I = \begin{pmatrix} A_{k-1}^{(I)} & \overline{\alpha} \\ \overline{\alpha} & \alpha \end{pmatrix}, \quad A_{k-1}^{(I)} = \overline{Q}_I^T A_{k-1} \overline{Q}_I,$$

4.1.2. The form of $Q_R$. We denote the transformed sub matrix of the previous stage by,

$$H^{(I)} = A^{(I)} + iB^{(I)}, \quad (A^{(I)})^T = A^{(I)}, \quad (B^{(I)})^T = -B^{(I)},$$

with,

$$A^{(I)} = \begin{pmatrix} A_{k-2}^{(I)} & \hat{a} & x \\ \hat{a}^T & \alpha' & \alpha \\ x & \alpha & \alpha \end{pmatrix}, \quad B^{(I)} = \begin{pmatrix} B_{k-2}^{(I)} & \hat{b} \\ \hat{b}^T & 0 & y \\ -y & 0 \end{pmatrix}.$$ 

Then, the transformation $Q_R$ will reduce the last row and column of $A^{(I)}$,

$$A^{(R)} = Q_R^T A^{(I)} Q_R = \begin{pmatrix} A_{k-2}^{(R)} & \hat{a} & z \\ \hat{a}^T & \alpha' & x \\ z & x & \alpha \end{pmatrix}, \quad z = \pm \| \overline{\alpha} \|_2.$$
Let us denote $B_{k-1}^{(k+1)}$ by $B$ and $B_{k-1}^{(k)}$ by $B'$, then
\begin{align}
B' &= B - 2w(w^t B) - 2(Bw)w^t + 4(w^t Bw)w^t, \\
&= B + 2\gamma (wu^t - uw^t),
\end{align}
where
\begin{equation}
u = \frac{1}{\gamma} Bw, \quad \gamma = \| Bw \|_2,
\end{equation}
since $w^t Bw = -w^t Bw = 0$. We conclude that the algorithm is computationally and numerically as efficient as for the real case, see Appendix (A.4). Hence, let $i\delta_i$ denote the eigenvalues of $B^{(k+1)}$ in a decreasing order of magnitude,
\begin{equation}
|\delta_1| \geq |\delta_2| \geq \cdots \geq |\delta_n|, \quad \delta_i \in \mathcal{R},
\end{equation}
and $\hat{\delta}_i'$ the corresponding eigenvalues of the computed transformed matrix, then
\begin{equation}
|\hat{\delta}'_i - \delta_i| \leq c \|B^{(k+1)}\|_F + O(\epsilon^2) \leq cn^{1/2}|\delta_1| + O(\epsilon^2),
\end{equation}
and more often,
\begin{equation}
|\hat{\delta}'_i - \delta_i| \leq c \|B^{(k+1)}\|_2 + O(\epsilon^2) = c|\delta_1| + O(\epsilon^2),
\end{equation}
where $\epsilon$ is the precision of the data.

4. A fast algorithm for complex Hermitian matrices. We present in this section a new fast algorithm for reducing a complex Hermitian matrix $H \in \mathbb{C}^{n \times n}$, $H^* = H$, to a tridiagonal form. The algorithm proceeds in $(n - 2)$ stages as before,
\begin{equation}
H^{(k)} = U_k^* H^{(k+1)} U_k, \quad k = n, \ldots, 3,
\end{equation}
with $H^{(n+1)} = H$, and $U_k$ a complex unitary matrix, $U_k^* U_k = I$ where the * denotes complex conjugate values. Let,
\begin{equation}
H^{(k+1)} = 
\begin{pmatrix}
H^{(k+1)}_k & \beta_{k+1} & \cdots & \cdots \\
\beta_{k+1} & \alpha_{k+1} & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
\beta_n & \cdots & \cdots & \alpha_n
\end{pmatrix},
\end{equation}
with the leading sub matrix
\begin{equation}
H^{(k+1)}_k = 
\begin{pmatrix}
H^{(k+1)}_{k-1} & h \beta_k \\
h^* & \alpha_k
\end{pmatrix},
\end{equation}
where $H^{(k+1)}_k \in \mathbb{C}^{k \times k}$, $H^{(k+1)}_{k-1} \in \mathbb{C}^{(k-1) \times (k-1)}$, $\beta_k, \alpha_k \in \mathcal{R}$, $j = k + 1, \ldots, n$.

We will then reduce the $k$th row and column of $H^{(k+1)}_k$ using the three transformations $U = Q_l Q_r U_C$, where we have omitted the subscript $k$ for simplicity. Here, $Q_l^T Q_l = I = Q_r^T Q_r$, are real orthogonal matrices, and
\begin{equation}
U_C = 
\begin{pmatrix}
I_{k-3} & V_C \\
V_C & 1
\end{pmatrix},
\end{equation}
with $V \in \mathbb{C}^{2 \times 2}$, $V_C V_C^* = I$,
\begin{equation}
is complex unitary. Hence, we apply mainly real orthogonal transformations.
and the matrix is split into independent parts.

These examples demonstrate that there are very well conditioned matrices for which the complex variation of the real symmetric QR algorithm will not work. Although we have considered only the very first transformation, this can similarly happen throughout the reduction process. Furthermore, even with a moderate value of $|w^T w|$ in (16), we might get an exponential growth in the size of the elements of the matrix due to the accumulated effect of these transformations. As a matter of fact, we have actually applied this algorithm to the solution of the Schrödinger equation as discussed in Section (6), and have received very bad results. The accuracy was almost completely lost for matrices of order less than 64. We will describe our solution to this problem in the following sections.

3. A fast algorithm for real skew symmetric matrices. We begin our discussion with a new algorithm for reducing a real skew symmetric matrix $B \in \mathbb{R}^{n \times n}$, $B^T = -B$, to a tridiagonal form. The algorithm proceeds in $(n-2)$ stages as in the symmetric case by,

$$B^{(k)} = Q_k^T B^{(k+1)} Q_k, \quad k = n, \ldots, 3,$$

with $B^{(n+1)} = B$ and $Q_k$ a real orthogonal matrix. Hence, as before,

$$T = B^{(3)} = Q^T B Q, \quad Q = Q_{(n)} Q_{(n-2)} \cdots Q_{(3)}, \quad Q^T Q = I.$$

At the beginning of the $k$th step we have,

$$B^{(k+1)} = \begin{pmatrix} B_k^{(k+1)} & \beta_{k+1} \\ -\beta_{k+1} & 0 \\ \vdots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \beta_n \\ -\beta_n & 0 \end{pmatrix} \quad \text{with } B_{k+1}^{(k+1)} \in \mathbb{R}^{k \times k}, \quad \beta_j \in \mathbb{R}, \quad j = k+1, \ldots, n,$$

and the leading sub matrix is,

$$B_k^{(k+1)} = \begin{pmatrix} B_{k-1}^{(k+1)} & b \\ -b^T & 0 \end{pmatrix} \quad \text{with } B_{k-1}^{(k+1)} \in \mathbb{R}^{(k-1) \times (k-1)}, \quad b \in \mathbb{R}^{(k-1)}.$$

We now reduce the $k$th row and column of the matrix by,

$$B_{k-1}^{(k)} = \overline{Q}_k^T B_{k-1}^{(k+1)} \overline{Q}_k, \quad \overline{Q}_k^T b = \beta_k e_{k-1}, \quad Q_k = \begin{pmatrix} \overline{Q}_k \\ I_{n-k+1} \end{pmatrix},$$

where $\beta_k = \pm \| b \|_2$. Here, as in the real case, $\overline{Q}_k$ is given by,

$$\overline{Q}_k = I - 2ww^T, \quad w = \frac{v}{\| v \|_2}, \quad v = b + \text{sign}(b_{k-1}) \| b \|_2 e_{k-1}.$$
small at all. To demonstrate that such matrices are not special, we give three examples. In the first, the matrix,

\[ H = \begin{pmatrix} 0 & i & i \\ i & 2 & 1 \\ i & 1 & 2 \end{pmatrix}, \quad X = \begin{pmatrix} -3 & i & i \\ i & -1 & 1 \\ 2i & 2 & 1 \end{pmatrix}, \quad HX = XD, \quad D = \text{diag}(1, 1, 2), \]

demonstrates that we cannot apply the bottom-up algorithm discussed before, although the matrix has a well conditioned basis of eigenvectors. We can recover this problem if we proceed with the equivalent top-down procedure. In the second example the matrix,

\[ H = \begin{pmatrix} 0 & 1 & i \\ 1 & 0 & 1 \\ i & 1 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} 1 & 1 & 1 \\ -\lambda_2 & -\lambda_1 & 0 \\ 1 & 1 & -1 \end{pmatrix}, \quad \lambda_1 = \frac{1}{2}(\sqrt{7} + i), \quad \lambda_2 = -\frac{1}{2}(\sqrt{7} - i), \quad \lambda_3 = -i, \]

demonstrates that we cannot proceed with either the bottom-up or the top-down procedures. Again, the matrix has simple eigenvalues and well conditioned eigenvectors. We can now recover this problem if we first permute the last two rows and columns of the matrix, and then proceed with the bottom-up algorithm. Finally, consider the matrix of order \( n = 4 \),

\[ H = \begin{pmatrix} 0 & 3 & 4 & 5i \\ 3 & 0 & 5i & 4 \\ 4 & 5i & 0 & 3 \\ 5i & 4 & 3 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \quad \lambda_1 = 7 + 5i, \quad \lambda_2 = 1 - 5i, \quad \lambda_3 = -1 - 5i, \quad \lambda_4 = -7 + 5i, \]

Here we have again a well conditioned spectral problem, but none of the previous solutions will help. The answer is then to apply a Jacobi rotation, for example to annihilates the 4 in position (2, 4). Hence,

\[ H' = J^T H J, \quad J = J^T = \begin{pmatrix} 1 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & 0 & 1 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix}, \]

so that,

\[ H' = \begin{pmatrix} 0 & -\frac{1}{\sqrt{2}}(3 - 5i) & \frac{1}{\sqrt{2}}(3 + 5i) \\ -\frac{1}{\sqrt{2}}(3 - 5i) & -4 & \frac{1}{\sqrt{2}}(3 - 5i) \\ \frac{1}{\sqrt{2}}(3 + 5i) & \frac{1}{\sqrt{2}}(3 - 5i) & 0 \\ \frac{1}{\sqrt{2}}(3 + 5i) & 0 & \frac{1}{\sqrt{2}}(3 + 5i) \\ 0 & \frac{1}{\sqrt{2}}(3 + 5i) & -4 \end{pmatrix}. \]

Proceeding now with the first bottom-up transformation we get,

\[ H'' = Q_4^T H' Q_4 = \begin{pmatrix} -4 & (3 - 5i) \\ (3 - 5i) & -4 \\ 4 & (3 + 5i) \\ (3 + 5i) & 4 \end{pmatrix}, \]
where \( \beta_k = \pm \| b \|_2 \) and \( e_{k-1} \in \mathcal{R}^{(k-1)} \) is the corresponding \((k-1)\)th standard unit vector. The transformation \( Q_k \), known as Householder transformation, is given by

\[
(8) \quad Q_k = I - 2ww^t, \quad w = \frac{v}{\| v \|_2}, \quad v = b + \text{sign}(b_{k-1}) \| b \|_2 e_{k-1}.
\]

The transformation could then be implemented efficiently as follows,

\[
(9) \quad A' = A - 2w(w^tA) - 2(Aw)w^t + 4(w^tAw)w^t,
\]
\[
(10) \quad A' = A - 2\gamma(wu^t + uw^t - 2(w^tu)ww^t),
\]
\[
(11) \quad A' = A - 2\gamma(wr^t + rw^t),
\]

where for simplicity we denote \( A_{(k+1)}^{(k-1)} \) by \( A \) and \( A_{k-1}^{(k)} \) by \( A' \). Here,

\[
(12) \quad u = \frac{1}{2}Aw, \quad \gamma = \| Aw \|_2, \quad r = u - (w^tu)w \quad \| r \|_2 = (1 - (w^tu)^2)^{1/2} \leq 1.
\]

This requires only one matrix vector and one outer vector products. Furthermore, if we order the eigenvalues of \( A^{(k+1)} \) in a decreasing order of magnitude,

\[
(13) \quad |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|, \quad \lambda_i \in \mathcal{R},
\]

and \( \hat{\lambda}_i' \) are the corresponding eigenvalues of the computed transformed matrix, then

\[
(14) \quad |\hat{\lambda}_i' - \lambda_i| \leq \epsilon \| A^{(k+1)} \|_F + O(\epsilon^2) \leq \epsilon n^{1/2} |\lambda_1| + O(\epsilon^2),
\]

or more often we have the better bound,

\[
(15) \quad |\hat{\lambda}_i' - \lambda_i| \leq \epsilon \| A^{(k+1)} \|_2 + O(\epsilon^2) = \epsilon |\lambda_1| + O(\epsilon^2),
\]

where \( \epsilon \) is the precision of the data, see Appendix (A.3) for more details. Hence, the effect of rounding errors in the whole reduction stage is negligible and the algorithm is numerically stable.

### 2.3. Reduction of a complex symmetric matrix to a tridiagonal form

We denote complex numbers as elements \( \in \mathcal{C} \), complex \( n \) dimensional vectors as elements \( \in \mathcal{C}^n \), and complex \( m \) by \( n \) dimensional matrices as elements \( \in \mathcal{C}^{m\times n} \). Let \( A \in \mathcal{C}^{n\times n} \) be a complex symmetric matrix so that \( A^T = A \). We first consider the QR algorithm modified so as to work with complex symmetric matrices. Here, the transformation corresponding to (8) should be,

\[
(16) \quad \overline{Q}_k = I - 2ww^t, \quad w = \frac{v}{\| v \|_2}, \quad v = b \pm (b^t b)^{1/2} e_{k-1},
\]

where the sign is taken so that \( |v_{k-1}| \) is maximal. However, unlike the real case, it is now possible that \( w^t w = 0 \) or in numerical terms \( |w^t w| \ll 1 \) even though \( \| w \|_2 \) is not
see Parlett[22], or by some divide and conquer methods, see Cuppen[5] and Bar-On[2].

- For a given set of eigenvalues compute the corresponding set of eigenvectors of the tridiagonal matrix $T$ by Inverse Iteration, see Watkins[28].
- Back transform the computed eigenvectors into their original basis by applying to them the orthogonal transformations of the reduction stage.

The main advantage of the symmetric QR algorithm comes from the fact that we can reduce the matrix to a symmetric tridiagonal form. We can then employ very fast and efficient algorithms to compute the spectrum of the matrix. This is not so for the non-symmetric variant of the algorithm, where we can reduce the matrix only to an upper(lower) Hessenberg form. Furthermore, the reduction stage in the symmetric case is faster, and the whole process can be better exploited on modern vector and parallel supercomputers. The main purpose of this work is to try to present an analog procedure for complex symmetric matrices.

2.2. Reduction of a real symmetric matrix to a tridiagonal form. We begin with some elementary notations. We denote real numbers as elements $\in \mathcal{R}$, real $n$ dimensional vectors as elements $\in \mathcal{R}^n$, and real $m$ by $n$ dimensional matrices as elements $\in \mathcal{R}^{m \times n}$. Let $A \in \mathcal{R}^{n \times n}$ be a real symmetric matrix so that $A^T = A$. Then, the process of reducing $A$ to a tridiagonal form $T$ can be summarized in the following:

$$A^{(k)} = Q_k^T A^{(k+1)} Q_k, \quad k = n, \ldots, 3,$$

where $A^{(n+1)} = A$, and $Q_k$ is a real orthogonal matrix. Hence,

$$T = A^{(3)} = Q^T A Q, \quad Q = Q_{(n)} Q_{(n-2)} \cdots Q_{(3)}, \quad Q^T Q = I.$$

At the beginning of the $k$th step, the transformed matrix has the structure,

$$A^{(k+1)} = \begin{pmatrix} A_{k}^{(k+1)} & \beta_{k+1} & & & \\
\beta_{k+1} & \alpha_{k+1} & \ddots & & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & \beta_n \\
& & & \beta_n & \alpha_n \end{pmatrix}, \quad A_{k}^{(k+1)} \in \mathcal{R}^{k \times k}, \quad \beta_j, \alpha_j \in \mathcal{R}, \quad j = k+1, \ldots, n,$$

and the leading sub matrix is,

$$A_{k}^{(k+1)} = \begin{pmatrix} A_{k-1}^{(k+1)} & b \\
b & \alpha_k \end{pmatrix}, \quad A_{k-1}^{(k+1)} \in \mathcal{R}^{(k-1) \times (k-1)}, \quad b \in \mathcal{R}^{(k-1)}.$$

We now reduce the $k$th row and column of the matrix by the following orthogonal transformation:

$$A_{k-1}^{(k)} = \bar{Q}_k^T A_{k-1}^{(k+1)} \bar{Q}_k, \quad \bar{Q}_k^T b = \beta_k e_{k-1}, \quad Q_k = \begin{pmatrix} \bar{Q}_k \\
I_{n-k+1} \end{pmatrix},$$
tonian in the asymptotic regions of a scattering problem

\[ H(R) \rightarrow H(R) - i V_{opt}(R). \]

The most time consuming step in the numerical implementations of both methods is the diagonalization of complex symmetric matrices, and this crucially determines their applicability to real chemical reactive problems[20, 26, 7, 15, 6, 25].

We will present in this paper a new fast algorithm for computing the spectrum of complex symmetric matrices, and fast variations for skew symmetric and complex Hermitian matrices as well.

We begin with a review of some of the related work done in this field. A first approach to this problem was suggested by Seaton[23] and Eberlein[9]. They have used a Jacobi like method to reduce a non-defective complex symmetric matrix to a diagonal form, by iteratively applying to it complex plane rotations. As in the real symmetric case this method can be effective for small size, and for large sparse problems but it is inefficient for large and dense problems, see for example the related discussion in Parlett[22]. A second approach was suggested by Moro and Freed[19] and Cullum and Willoughby[4], using the Lanczos algorithm. Here again, the method is effective for large and sparse problems, especially with regard to the extreme spectrum, but it is not the preferred method for large dense problems. Our aim in this paper is to give a new QR type algorithm for the complex symmetric problem which is computationally as efficient as the related algorithm for Hermitian matrices. The new algorithm is also highly suitable for modern parallel supercomputers.

This paper is organized as follows. We begin with a brief overview of the real symmetric QR algorithm, and its possible adaptation to complex symmetric matrices in Section 2. We then describe a QR variant for real skew symmetric matrices in Section 3, and a new fast and efficient algorithm for complex Hermitian matrices in Section 4. We describe the new algorithm for complex symmetric matrices in Section 5. Finally we give an application of the algorithm to the complex scaled discrete variable method in quantum reaction dynamics in Section 6. We conclude with some open remaining research problems, and in the Appendix we present an improved perturbation analysis for the real Householder transformation.


2.1. The QR algorithm for Real Symmetric matrices. The QR algorithm for real symmetric matrices is one of the most efficient algorithms in numerical linear algebra, and has inspired the whole computational work done in this field. We begin with a summary of its main computational stages:

- Reduce the original dense matrix A to a tridiagonal form T by a sequence of orthogonal transformations. The tridiagonal matrix represents the same operator as the original matrix but in a more compact form.
- Compute the set of eigenvalues of T, which is also the set of eigenvalues of A, by any efficient method such as the QR algorithm, or the Bisection algorithm,
FAST DIAGONALIZATION OF LARGE AND DENSE COMPLEX SYMMETRIC MATRICES, WITH APPLICATIONS TO QUANTUM REACTION DYNAMICS

ILAN BAR-ON
DEPARTMENT OF COMPUTER SCIENCE
VICTOR RYABOY
DEPARTMENT OF CHEMISTRY
TECHNION, HAIFA 32000, ISRAEL

Abstract. We present a new fast and efficient algorithm for computing the eigenvalues and eigenvectors of large size complex symmetric dense matrices. The principal new idea is to reduce the matrix to a tridiagonal complex symmetric form. We can then compute the eigenvalues, very fast, using a complex versions of the QL algorithm for tridiagonal symmetric matrices. The corresponding eigenvectors are similarly computed using a complex version of the inverse iteration algorithm. We show that the new algorithm is faster by an order of magnitude than the corresponding EISPACK routines that are currently used for such problems. We present also similar methods for skew symmetric matrices, and for complex Hermitian matrices, the last being twice as fast as the corresponding EISPACK routine. The new algorithm is also highly suitable for modern parallel and vector supercomputers. The complex scaled discrete variable representation (DVR) method for quantum chemical reactive problems, relies heavily on the efficient diagonalization of complex symmetric matrices. We present an application to this problem which has motivated our work. We finally present new perturbation bounds for the Householder transformation which improve those results obtained before by Wilkinson.

Key words. Skew Symmetric, Complex Hermitian, Complex Symmetric, Diagonalization, QR, Schrodinger equation, Reaction Dynamics, Supercomputers.

1. Introduction. The problem of diagonalising large and dense complex symmetric matrices has received little attention in the past, although it is an intensive computational procedure, possibly for the following reasons:

- They appear less frequently in practice than for example real symmetric and complex Hermitian matrices.
- Any general complex matrix is similar to some complex symmetric matrix, see Gantmacher[10], and thus complex symmetric matrices do not necessarily possess any special spectral structure.
- Wilkinson in his monumental work which has inspired much of the work in this field, has paid little attention to this problem see [29]page 265.

Thus it is customary to use general purpose algorithms for this problem.

However several new methods have emerged recently in the field of quantum chemical reaction dynamics which aim at conversion of scattering problem into a bound state problem in non-Hermitian domain. We have in mind in this context different formulations of the complex rotation method[8, 16, 17, 30, 3, 31, 21] and the optical potential method[11, 18]. The first one is based on rotation of the reaction coordinate $R$ into the upper half of the complex plane by the angle $\theta$,

$$\tilde{R} = (R - R_0) \exp(i\theta) + R_0$$

the Hamiltonian being substituted by $H(R) \rightarrow H(\tilde{R})$. In the second one a complex "optical" potential which provides absorbing boundary condition is added to the Hamil-