NASA TECHNICAL NOTE


NASA TN D-6429



NASA TN D-6429

# A CONTRACTING-INTERVAL PROGRAM FOR THE DANILEWSKI METHOD

*by James D. Harris*

*Langley Research Center*
*Hampton, Va. 23365*

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • NOVEMBER 1971

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA TN D-6429 | | |

4. Title and Subtitle
A CONTRACTING-INTERVAL PROGRAM FOR THE DANILEWSKI METHOD

5. Report Date
November 1971

6. Performing Organization Code

7. Author(s)
James D. Harris

8. Performing Organization Report No.
L-7725

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23365

10. Work Unit No.
136-13-05-04

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes
This research was originally presented to the Faculty of the School of Engineering and Applied Science, University of Virginia, in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Applied Mathematics).

16. Abstract

The concept of contracting-interval programs is applied to the Danilewski method for finding the eigenvalues of a matrix. The development is a three-step process in which (1) a contracting-interval program is developed for the reduction of a matrix to Hessenberg form, (2) a contracting-interval program is developed for the reduction of a Hessenberg matrix to colleague form, and (3) the characteristic polynomial with interval coefficients is readily obtained from the interval of colleague matrices, and this interval polynomial is then factored into quadratic factors so that the eigenvalues may be obtained.

To develop a contracting-interval program for factoring this polynomial with interval coefficients it is necessary to have an iteration method which converges even in the presence of controlled rounding errors.

A theorem is stated giving sufficient conditions for the convergence of Newton's method when both the function and its Jacobian cannot be evaluated exactly but when the errors can be made proportional to the square of the norm of the difference between the previous two iterates. This theorem is applied to prove the convergence of the generalization of the Newton-Bairstow method that is used to obtain quadratic factors of the characteristic polynomial.

| 17. Key Words (Suggested by Author(s)) | 18. Distribution Statement |
|---|---|
| Matrix<br>Eigenvalue<br>Danilewski<br>Newton's method<br>Rounding errors<br>Interval arithmetic | Unclassified – Unlimited |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price* |
|---|---|---|---|
| Unclassified | Unclassified | 73 | $3.00 |

*For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

# A CONTRACTING-INTERVAL PROGRAM FOR THE DANILEWSKI METHOD*

By James D. Harris
Langley Research Center

## SUMMARY

The concept of contracting-interval programs is applied to the Danilewski method for finding the eigenvalues of a matrix. The development is a three-step process in which (1) a contracting-interval program is developed for the reduction of a matrix to Hessenberg form, (2) a contracting-interval program is developed for the reduction of a Hessenberg matrix to colleague form, and (3) the characteristic polynomial with interval coefficients is readily obtained from the interval of colleague matrices, and this interval polynomial is then factored into quadratic factors so that the eigenvalues may be obtained.

To develop a contracting-interval program for factoring this polynomial with interval coefficients it is necessary to have an iteration method which converges even in the presence of controlled rounding errors.

A theorem is stated giving sufficient conditions for the convergence of Newton's method when both the function and its Jacobian cannot be evaluated exactly but when the errors can be made proportional to the square of the norm of the difference between the previous two iterates. This theorem is applied to prove the convergence of the generalization of the Newton-Bairstow method that is used to obtain quadratic factors of the characteristic polynomial.

*This research was originally presented to the Faculty of the School of Engineering and Applied Science, University of Virginia, in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Applied Mathematics).

## SYMBOLS

Standard matrix notation is used throughout; that is, elements are designated by lower case letters corresponding to the matrix symbol and with appropriate subscripts.

| | |
|---|---|
| $A$ | real $n$ by $n$ matrix |
| $\tilde{A}$ | $A$ after row and column interchanges |
| $B$ | matrix containing rounding errors made in reduction of $H$ to $F$ |
| $E$ | matrix containing rounding errors made in converting $\tilde{A}$ to $H$ |
| $F$ | colleague matrix in chapter III |
| $F$ | defined as $NM^* - I$ in chapter II |
| $f(x)$ | vector whose components are functions of $x$ |
| $f_\xi(x)$ | approximation to $f$ |
| $H$ | upper Hessenberg matrix |
| $I$ | identity matrix |
| $I_{jj'}$ | identity matrix with columns $j$ and $j'$ interchanged |
| $J$ | Jacobian of $f$ |
| $J_{\xi_i}$ | approximation to $J$ |
| $M$ | inverse of $N$ |
| $N$ | lower triangular matrix |
| $n$ | order of matrix $A$ |
| $P(x)$ | polynomial of degree $n$ |
| $P_i(x)$ | polynomials satisfying recurrence relation |

$P_i$ real numbers satisfying recurrence relation

$R$ large positive number

$R^n$ n-dimensional real-coordinate space

$u(s,t), v(s,t)$ functions defined following algorithm H of chapter IV

$V$ upper triangular matrix

$W$ inverse of $V$

$\alpha_i$ superdiagonal element of colleague matrix

$\beta_i$ diagonal element in colleague matrix

$\delta f(x)$ positive continuous function of $x$

$\delta J(x)$ positive continuous function of $x$

$\delta P(x)$ polynomial of degree $n - 1$ with positive coefficients

$\lambda$ constant with same dimensions as $1/x$

Notation used is as follows:

$\|A\|$ where $A$ is a matrix $[a_{ij}]$ of order $n$, $\|A\| = \max\limits_{1\leq i\leq n} \left\{ \sum\limits_{j=1}^{n} |a_{ij}| \right\}$

$A \leqq B$ if $A$ and $B$ are vectors or matrices, this means that each component of $A$ is less than or equal to corresponding component of $B$

$[A \pm \delta A]$ $\{A^1 | A - \delta A \leqq A^1 \leqq A + \delta A\}$ or interval with midpoint $A$ and half-width $\delta A$

$*$ asterisk following a symbol denotes approximation; for instance, $u^*(s,t)$ is an approximation to $u(s,t)$

$f(S)$ $\{f(X) | X \in S\}$ where $f$ is a function and $S \subseteq R^n$

$\left[\frac{n}{2}\right]$ greatest integer less than $\frac{n}{2}$

$[P \pm \delta P]$ where $P(x) = a_n + a_{n-1}P_1 + \ldots + a_0P_n$ and $\delta P = \delta a_n + \delta a_{n-1}P_1 + \ldots + \delta a_1 P_{n-1}$ $(\delta a_i \geqq 0$ for each $i)$, $[P \pm \delta P]$ denotes set of all polynomials $P'(x) = a'_n + \ldots + a'_1 P_{n-1} + a_0 P_n$ such that $a'_i \in [a_i \pm \delta a_i]$ $(i = 1,2,\ldots,n)$

$\prod$ product symbol

sup supremum (least upper bound)

$\|x\|$ where, $x$ is a vector $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\|x\| = \max\limits_{1 \leqq i \leqq n} \{|x_i|\}$

$\{x|s\}$ set of all $x$ such that statement $s$ is true

$\min\{x|s\}$ least element of indicated set

# CHAPTER I

# INTRODUCTION

This paper presents the results of an investigation of techniques for applying the concept of contracting-interval programs (ref. 1) to the Danilewski method for finding the eigenvalues of a matrix.

## Justification

Most data used for input to a computer program are measured data or are the result of truncating exact data. These inexact data are then processed by a computer program which makes rounding errors. Thus, much analysis may be required to determine how closely the computer solution approximates the exact solution of the problem.

Contracting-interval programs present an alternative to this procedure. The user of a contracting-interval program can supply an interval which is known to contain the exact data. The contracting-interval program then gives him an interval, each element of which is the exact answer corresponding to some element in the input interval. This is as good a result as he could hope to obtain.

Notice that contracting-interval programs are not the same as expanding-interval programs. An expanding-interval program computes an output interval which contains all of the answers corresponding to elements of the input interval. However, there may be elements in the output interval of an expanding-interval program which do not correspond to any element in the input interval. For a further discussion of the differences between contracting- and expanding-interval programs, see reference 1.

The Danilewski method is particularly appropriate for testing the concept of contracting-interval programs, since it is a method which is very sensitive to rounding errors. When executed by using a fixed-precision arithmetic, it frequently gives unreliable results because of the rounding errors, but a contracting-interval program based on the Danilewski method yields exact results. Also, the Danilewski method is a fast method for obtaining eigenvalues and, even with the increased precision that a contracting-interval program requires, it may prove to be competitive in speed with other more inherently stable methods (ref. 2).

The Danilewski method used in the present paper is not the classical Danilewski method but is a three-step process whereby (1) a matrix is reduced to Hessenberg form by a similarity transformation, (2) the resulting Hessenberg matrix is reduced to colleague form (ref. 3) by a similarity transformation, and (3) the characteristic polynomial is obtained from the colleague matrix and the roots of this polynomial are calculated.

As Wilkinson (ref. 4) has shown, it is in this second step that problems develop. The resulting colleague matrix may be significantly more ill-conditioned than was the corresponding Hessenberg matrix.

Previously, contracting-interval programs have been developed for solving linear equations (ref. 1) and for finding real roots of a polynomial (ref. 5), but a contracting-interval program has not been attempted for a problem so complex as the matrix-eigenvalue problem.

## Preview of Remaining Chapters

Any exact computation defines a mapping $f$ from one vector space $R^n$ to another vector space $R^m$. If $S \subset R^n$, then $f(S) \equiv \{f(X) | X \in S\}$. If $[A \pm \delta A] \subseteq R^n$, then $f[A \pm \delta A]$ is not necessarily an interval. However, a contracting-interval program computes an interval $[B \pm \delta B]$ such that $[B \pm \delta B] \subseteq f[A \pm \delta A]$.

A two-step process is used here to develop contracting-interval programs. The first step obtains $B$ such that $B = f(A')$ where $|A - A'| \leq \frac{\delta A}{R}$, and $R$ is a large positive number. During this step the program must have the ability to control the rounding errors made in each arithmetic operation. The second step finds $\delta B$ such that $[B \pm \delta B] \subseteq f\left[A' \pm \frac{R - 1}{R} \delta A\right]$; that is, if $B' \in [B \pm \delta B]$, then $B' = f(A'')$ where $|A' - A''| \leq \frac{R - 1}{R} \delta A$.

Contracting-interval programs have the property (ref. 1) that, given contracting-interval programs for $f$ and $g$, there is automatically obtained a contracting-interval program for the composition of $f$ and $g$.

In chapter II a contracting-interval program is developed for reducing an interval $[A \pm \delta A]$ of real matrices to an interval $[H \pm \delta H]$ of upper Hessenberg matrices by a similarity transformation. Most of the material in chapter II is new, although the algorithm for obtaining it is based on the one given in reference 4 on page 357.

In chapter III a contracting-interval program for reducing an interval $[H \pm \delta H]$ of upper Hessenberg matrices to an interval $[F \pm \delta F]$ of colleague matrices by a similarity transformation is developed. The concept of a colleague matrix that is used here is a generalization of the one given in reference 3. The form of the colleague matrix as shown in chapter III was suggested by B. A. Chartres. Much of the work in chapter III is based on work done in reference 6 although, in reference 6, Chartres was concerned with obtaining a companion matrix $F$ which was similar to some $H' \in [H \pm \delta H]$. From $[F \pm \delta F]$ a set of polynomials $[P \pm \delta P]$ is obtained, such that each element is the characteristic polynomial of some $A' \in [A \pm \delta A]$.

In chapter IV an algorithm is developed for obtaining an interval $\left[b_1 \pm \delta b_1\right]$ and the intervals $\left[s_i \pm \delta s_i\right]$ and $\left[t_i \pm \delta t_i\right]$ $\left(i = 1,2,\ldots,\left[\frac{n}{2}\right]\right)$, where $\left[\frac{n}{2}\right]$ is the greatest integer in $\frac{n}{2}$ and $n$ is the degree of $P$ such that, if $s' \in \left[s_i \pm \delta s_i\right]$, $t' \in \left[t_i \pm \delta t_i\right]$, and $b' \in \left[b_1 \pm \delta b_1\right]$, then

$$\left(a_0 + b_1' P_1\right) \prod_{i=1}^{\left[\frac{n}{2}\right]} \left(P_2 - s_i' P_1 - t_i\right) \in [P \pm \delta P]$$

If $n$ is even, then $b_1 = \delta b_1 = 0$. The method used is a generalization of the Newton-Bairstow method. In appendix A, a proof is given that the algorithm developed in chapter IV leads to convergence.

From chapter IV it may be seen that $P_2 - s_i' P_1 - t_i'$ is a factor of the characteristic polynomial of some matrix $A' \in [A \pm \delta A]$ for $s_i' \in \left[s_i \pm \delta s_i\right]$ and $t_i' \in \left[t_i \pm \delta t_i\right]$ $\left(i = 1,2,\ldots,\left[\frac{n}{2}\right]\right)$. Thus, the roots of this quadratic polynomial are eigenvalues of the matrix $A'$. Intervals about the eigenvalues are not determined in this study.

## CHAPTER II

## REDUCTION OF AN INTERVAL OF REAL MATRICES TO AN INTERVAL OF UPPER HESSENBERG MATRICES

In chapter II an algorithm for converting an interval of real matrices into an interval of upper Hessenberg matrices by a similarity transformation is described.

Let $A$ be a real matrix of interval midpoints, and let $\delta A$ be a matrix whose elements are the interval half-widths. Let $H$ denote an upper Hessenberg matrix; that is, $H$ has the form

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ & h_{32} & h_{33} & \cdots & h_{3n} \\ & & h_{43} & \cdots & h_{4n} \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ & & & h_{n,n-1} & h_{nn} \end{bmatrix}$$

Let $N$ denote a matrix of the form

$$N = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & n_{32} & 1 & & \\ 0 & n_{42} & n_{43} & 1 & \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & n_{n2} & n_{n3} & n_{n4} & 1 \end{bmatrix}$$

Let $\delta H$ be an upper Hessenberg matrix of all positive elements. The matrices $H$ and $\delta H$ are determined such that if $H' \in [H \pm \delta H]$, then $H'$ is similar to some $A' \in [A \pm \delta A]$. To do this, an $N$ of the form given above is found such that if $H' \in [H \pm \delta H]$, then $H' = N^{-1}A'N$ for some $A' \in [A \pm \delta A]$; that is, $[H \pm \delta H] \subseteq N^{-1}[A \pm \delta A]N$.

Since A is the midpoint of the original interval, it is desirable that the midpoint of the interval of Hessenberg matrices satisfy $H = N^{-1}AN$ for some N; that is,

$$NH = AN \tag{1}$$

Such an H can be calculated by using exact arithmetic in the following algorithm as long as no zero elements are generated along the subdiagonal of H (ref. 4, p. 358):

Algorithm A

For $r = 1,2,\ldots,n$

  For $i = 1,2,\ldots,\min\{r+1,n\}$

$$h_{ir} = a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=2}^{i-1} n_{ik}h_{kr}$$

  For $i = r+2,\ldots,n$

$$n_{i,r+1} = \frac{a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=2}^{r} n_{ik}h_{kr}}{h_{r+1,r}}$$

Now some provision must be made so that the algorithm applies in the case when an $h_{r+1,r} = 0$ is calculated. This is done in the following manner:

Algorithm B

For $r = 1,2,\ldots,r$

  For $i = 1,2,\ldots,r$

$$h_{ir} = a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=2}^{i-1} n_{ik}h_{kr}$$

If $r < n$, then

For $i = r+1, r+2, \ldots, n$

$$\sigma_i = a_{ir} + \sum_{k=r+1}^{n} a_{ik} n_{kr} - \sum_{k=2}^{r} n_{ik} h_{kr}$$

Find $j$ such that

$$|\sigma_j| = \max \left\{ |\sigma_{r+1}|, |\sigma_{r+2}|, \ldots, |\sigma_n| \right\}$$

Interchange $(\sigma_{r+1}, \sigma_j)$

If $\sigma_{r+1} \neq 0$, then

For $i = r+1, r+2, \ldots, n$

Interchange $(a_{ji}, a_{r+1,i})$

For $i = 1, 2, \ldots, n$

Interchange $(a_{ij}, a_{i,r+1})$

For $i = 1, 2, \ldots, r$

Interchange $(n_{ji}, n_{r+1,i})$

$h_{r+1,r} = \sigma_{r+1}$

For $i = r+2, r+3, \ldots, n$

$$n_{i,r+1} = \frac{\sigma_i}{h_{r+1,r}}$$

If $\sigma_{r+1} = 0$, then

For $i = r+2, r+3, \ldots, n$

$n_{i,r+1} = 0$

With algorithm B, an $N$ and an $H$ are obtained, but equation (1) is no longer satisfied.

## Effect of Interchanges

The effect of interchanges is shown in theorem 1 in which $I_{jj'}$ denotes the identity matrix with columns j and j' interchanged. When $I_{jj'}$ multiplies a matrix on the left, the result is to interchange rows j and j' in that matrix. When $I_{jj'}$ multiplies a matrix on the right, the result is to interchange columns j and j' in that matrix.

Theorem 1 – Given an $n \times n$ matrix $A^0$, if N and H are calculated by algorithm B, then $NH = \tilde{A}N$ where

$$\tilde{A} = I_{n-1,(n-1)'} I_{n-2,(n-2)'} \cdots I_{22'} A^0 I_{22'} \cdots I_{n-2,(n-2)'} I_{n-1,(n-1)'}$$

and each i' corresponds to the j calculated by algorithm B when $r = i - 1$.

Proof – See reference 7.

## Bounds on Rounding Errors

If the interchanges which need to be made to A are known ahead of time, the algorithm for finding N and H from $\tilde{A}$ is the same as algorithm A.

Up to this point, rounding errors have been ignored. In reality, the computed H and N satisfy

For $r = 1,2,\ldots,n$

  For $i = 1,2,\ldots,\min\{r+1,n\}$

$$h_{ir} = \tilde{a}_{ir} + \sum_{k=r+1}^{n} \tilde{a}_{ir} n_{kr} - \sum_{k=2}^{i-1} n_{ik} h_{kr} + e_{ir}$$

  For $i = r+2, r+3, \ldots, n$

$$n_{i,r+1} = \frac{\tilde{a}_{ir} + \sum_{k=r+1}^{n} \tilde{a}_{ik} n_{kr} - \sum_{k=2}^{r} n_{ik} h_{kr} + e_{ir}}{h_{r+1,r}}$$

where $e_{ir}$ is the rounding error made at a given step.

Let $E = [e_{ij}]$; then

$$NH - \tilde{A}N = E \qquad (2)$$

Now bounds on $E$ which guarantee that $H$ is very close to a solution of $NH = \tilde{A}N$ must be found. Also, $\delta H$ must be found such that

$$[H \pm \delta H] \subseteq N^{-1}[\tilde{A} \pm \delta\tilde{A}]N$$

This can be done as follows: Let $H' \in [H \pm \delta H]$ and $A' = NH'N^{-1}$, then $NH' - A'N = 0$. Subtracting this from equation (2) yields

$$N(H - H') = (\tilde{A} - A')N + E$$

Therefore

$$\begin{aligned} |A' - \tilde{A}| &\leq |N(H' - H)N^{-1}| + |EN^{-1}| \\ &\leq |N||H - H'||N^{-1}| + |E||N^{-1}| \end{aligned}$$

Thus

$$|A' - \tilde{A}| \leq \delta\tilde{A}$$

if

$$|N||\delta H||N^{-1}| \leq \frac{R - 1}{R}\,\delta\tilde{A} \tag{3}$$

and

$$|E||N^{-1}| \leq \frac{1}{R}\,\delta\tilde{A} \tag{4}$$

where $R$ is a large positive number.

If $R$ is chosen to be very large then $|E|$ must be very small, but $\delta H$ gets larger as $R$ gets larger. If $|E|$ is very small then a large amount of precision must be used in the computations. Since in the algorithms for computing $\delta H$, no attempt is made to compute the largest possible $\delta H$ satisfying equation (3), there is very little to be gained by choosing an extremely large value for $R$. Usually a satisfactory choice is $R = 10$. Then $\delta H$ is taken to be as large as possible and still satisfy equation (3). The algorithm used to determine $\delta H$ is described in the last section of this chapter.

Let $G = [\gamma_{ij}]$ where $\gamma_{ij} > 0$ for all $i$ and $j$. If

$$G|N^{-1}| \leq \frac{1}{R}\,\delta\tilde{A} \tag{5}$$

then equation (4) can be satisfied by requiring that $|E| \leq G$.

Let $M = N^{-1}$, where $M$ has the same form as $N$. Therefore,

$$G|M| = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nn} \end{bmatrix} \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & |m_{32}| & 1 & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & |m_{n2}| & |m_{n3}| & 1 \end{bmatrix}$$

Since equation (5) must be satisfied, the following equations must be satisfied:

$$\gamma_{i1} \leqq \frac{\delta \tilde{a}_{i1}}{R} \qquad (i = 1,2,\ldots,n)$$

and

$$\gamma_{ir} + \sum_{j=r+1}^{n} \gamma_{ij}|m_{jr}| \leqq \frac{\delta \tilde{a}_{ir}}{R} \qquad (n \geqq r > 1;\ n \geqq i > 1) \tag{6}$$

Equation (6) is satisfied if

$$\gamma_{ij}|m_{jr}| \leqq \frac{\delta \tilde{a}_{ir}}{R(n - r + 1)} \qquad (j = r,r+1,\ldots,n)$$

Therefore, $\gamma_{ij}$ must satisfy

$$\gamma_{ij} \leqq \min_{2 \leqq r \leqq j} \left\{ \frac{\delta \tilde{a}_{ir}}{R(n - r + 1)|m_{jr}|} \right\}$$

From this analysis it is apparent that if $\gamma_{ij}$ is defined to be

$$\left.\begin{aligned} \gamma_{ij} &\equiv \min_{2 \leqq r \leqq j} \left\{ \frac{\delta \tilde{a}_{ir}}{R(n - r + 1)|m_{jr}|} \right\} \qquad (j \geqq 2) \\ \gamma_{i1} &\equiv \frac{\delta a_{i1}}{R} \end{aligned}\right\} \tag{7}$$

then, equation (5) is satisfied and $|E||M| \leqq \frac{\delta A}{R}$ for all $E$ such that $e_{ij} \leqq \gamma_{ij}$. Thus, the $\gamma_{ij}$ can serve as bounds on the rounding errors.

## Effect of Interchanges on Rounding-Error Bounds

Now, the above assumes that the row and column interchanges that need to be made are known beforehand, but these interchanges must be determined by the algorithm; and $\gamma_{ij}$ must be calculated before $h_{ij}$ is calculated if $i \leqq j + 1$ and before $n_{i,j+1}$ is calculated if $i > j + 1$.

The following notation is used: $b = a \oplus e$ means that $b$ is set equal to a value which differs from $a$ by no more than $e$, that is, $|b - a| \leqq e$; $b = a \oplus e$ means that $0 < b - a < e$, and $b = a \ominus e$ means that $0 < a - b < e$; $b = a(1 \oplus e)$ is defined to be equivalent to $b = a \oplus (|a|e)$ with analogous meanings for $b = a(1 \oplus e)$ and $b = a(1 \ominus e)$.

Now, suppose the rounding-error bounds $b_{ij}$ are determined as in algorithm C. Note that it is important that the computed $b_{ij}$ be no larger than the $b_{ij}$ that would be calculated if exact arithmetic were used. However, it is not necessary that the $b_{ij}$ be calculated extremely accurately since a small change in the rounding-error bound is not likely to change significantly the amount of precision required in the corresponding computation. Thus, for example, it is required that

$$b_{i1} = \frac{\delta a_{i1}}{R}(1 \ominus 0.01)$$

that is, $b_{i1}$ is required to be less than the exact result of the computation $\frac{\delta a_{i1}}{R}$, but the use of a large amount of precision is not required in the computation.

Algorithm C

For $i = 1,2,\ldots,n$

$$b_{i1} = \frac{\delta a_{i1}}{R}(1 \ominus 0.01)$$

For $r = 1,2,\ldots,n$

For $i = 1,2,\ldots,r$

$$h_{ir} = \left(a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=2}^{i-1} n_{ik}h_{kr}\right) \oplus b_{ir}$$

If $r < n$, then

For $i = r+1, r+2, \ldots, n$

$$\sigma_i = a_{ir} + \sum_{k=r+1}^{n} a_{ik} n_{kr} - \sum_{k=2}^{r} n_{ik} h_{kr}$$

Find $j$ such that

$$|\sigma_j| = \max\left\{|\sigma_{r+1}|, |\sigma_{r+2}|, \ldots, |\sigma_n|\right\}$$

Interchange $(\sigma_{r+1}, \sigma_j)$

If $\sigma_{r+1} \neq 0$, then

For $i = r+1, r+2, \ldots, n$

Interchange $(a_{ji}, a_{r+1,i})$

For $i = 1, 2, \ldots, n$

Interchange $(a_{ij}, a_{i,r+1})$

Interchange $(\delta a_{ij}, \delta a_{i,r+1})$

Interchange $(\delta a_{ji}, \delta a_{r+1,i})$

For $i = 1, 2, \ldots, r$

Interchange $(n_{ji}, n_{r+1,i})$

For $i = r+2, r+3, \ldots, n$

$$n_{i,r+1} = \sigma_i / \sigma_{r+1}$$

$$n_{r+1,r} = \sigma_{r+1}$$

If $\sigma_{r+1} = 0$, then

For $i = r+2, r+3, \ldots, n$

$$n_{i,r+1} = 0$$

For $i = 2,3,\ldots,r$

$$m_{r+1,i} = -\left(n_{r+1,i} + \sum_{j=i+1}^{r} n_{r+1,j} m_{ji}\right)$$

$$m_{r+1,r+1} = 1$$

For $i = 1,2,\ldots,n$

$$b_{i,r+1} = \min_{2 \leq j \leq r+1} \left\{ \left[ \frac{\delta a_{ij}}{R(n - j + 1)\left|m_{r+1,j}\right|} \right] (1 \ominus 0.01) \right\}$$

Now it must be determined whether or not the bounds on the rounding errors as computed in algorithm C are larger than those given by equation (7). Suppose that during execution of algorithm C all calculations for $r \leq r'$ have been completed. Any remaining interchanges affect only rows $r'+2,r'+3,\ldots,n$ of N, A, and $\delta A$, and columns $r'+2,r'+3,\ldots,n$ of A and $\delta A$.

The only elements of M required for the calculation of $b_{ir'}$ $(i = 1,2,\ldots,n)$ are those in row $r'$. These elements are not affected by the remaining interchanges. Therefore, row $r'$ of M is a row of the inverse of the N obtained after the execution of algorithm C is finished.

For $i \leq r' + 1$, $\delta a_{ir'} = \delta \tilde{a}_{ir'}$. Therefore, for $i \leq r' + 1$, $b_{ir'} \leq \gamma_{ir'}$.

Now, for $i > r' + 1$, the element in position $n_{i,r'+1}$ after all interchanges are completed may be the element which was in position $n_{i',r'+1}$ when $r = r'$ during the execution of the algorithm. Thus, $b_{i',r'}$ was used as a bound for the rounding error made in the computation of $n_{i,r'+1}$. This implies that $\delta \tilde{a}_{ir'}$ is equal to the element which was in position $\delta a_{i',r'}$ when $r = r'$. Therefore, $b_{i',r'}$ computed during the execution of algorithm C satisfies $b_{i',r'} \leq \gamma_{ir'}$; that is, the bound on the rounding error which was made in the calculation of $n_{i,r'+1}$ is no larger than the one given by equation (7).

Therefore, if algorithm C is used to compute N and H,

$$\left|E\right|\left|N^{-1}\right| \leq \frac{\delta \tilde{A}}{R} \tag{8}$$

where E is the matrix containing the actual rounding errors.

## An Upper Bound on $|M|$

By using variable-precision arithmetic such as SPAR (ref. 8) provides, the elements of $M$ can be calculated exactly. However, the $b_{ir}$ need not be extremely accurate. It is required only that

$$b_{ir} \leqq \min_{2 \leqq j \leqq r} \left\{ \frac{\delta a_{ij}}{R(n - j + 1)m_{rj}} \right\} \tag{9}$$

to guarantee that equation (8) is satisfied. Therefore, rather than compute $M$ exactly, it is more efficient to calculate an upper bound on $|M|$ by using single-precision arithmetic.

One way of finding an upper bound on $|M|$ is as follows: Let $M^*$ be that approximation to $M$ which is calculated by

$$m^*_{ip} = -fl\left( n_{ip} + \sum_{j=p+1}^{i-1} n_{ij} m^*_{jp} \right) \tag{10}$$

where $fl$ denotes that the operations are done in single-precision floating-point arithmetic. Let $S_1 = n_{ip}$ and $S_{k+1} = \left[ S_k + n_{i,k+p}(1 + \rho_k) m^*_{k+p,p}(1 + \eta_k) \right](1 + \theta_k)$. Then, $m^*_{ip} = -S_{i-p}$ for some $\rho_k$, $\eta_k$, and $\theta_k$ $(k = 1,2,\ldots,i-p)$ satisfying $|\rho_k|$, $|\eta_k|$, $|\theta_k| < \beta_{sp}$, and $\beta_{sp}$ is a bound on the rounding errors made in single-precision calculations (ref. 9, p. 7). Therefore,

$$S_{i-p} = S_1 \prod_{i=1}^{i-p-1} (1 + \theta_i) + \sum_{j=p+1}^{i-1} n_{ij} m^*_{jp}(1 + \rho_{j-p})(1 + \eta_{j-p}) \prod_{r=j-p}^{i-p-1} (1 + \theta_r)$$

and

$$m^*_{ip} = -S_{i-p}$$

Therefore

$$0 = m^*_{ip} + n_{ip} \prod_{i=1}^{i-p-1} (1 + \theta_i) + \sum_{j=p+1}^{i-1} n_{ij}(1 + \rho_{j-p}) m^*_{jp}(1 + \eta_{j-p}) \prod_{r=j-p}^{i-p-1} (1 + \theta_r) \tag{11}$$

Let $F = NM^* - I$ where $F = [f_{ij}]$. If $i < j$, $f_{ij} = 0$. If $i = j$, $f_{ij} = 1 - 1 = 0$. If $i = j + 1$, $f_{ij} = n_{j+1,j} + m_{j+1,j} = 0$, since the operation $m_{j+1,j} = -n_{j+1,j}$ can be performed exactly. If $i > p + 1$

$$f_{ip} = n_{ip} + \sum_{j=p+1}^{i-1} n_{ij} m^*_{jp} + m^*_{ip} \tag{12}$$

Subtracting equation (11) from equation (12) yields

$$f_{ip} = n_{ip}\left[1 - \prod_{i=1}^{i-p-1}(1 + \theta_i)\right] + \sum_{j=p+1}^{i-1} n_{ij} m^*_{jp}\left[1 - (1 + \rho_{j-p})(1 + \eta_{j-p}) \prod_{r=j-p}^{i-p-1}(1 + \theta_r)\right] \tag{13}$$

The following lemma is proved on page 65 in reference 10:

Lemma – If $|\epsilon_i| < \beta < 1$ and $0 \leq r \leq k \leq n$ and $n\beta' = \exp\left(\frac{n\beta}{1 - \beta}\right) - 1$ then

$$\left| \prod_{i=1}^{r}(1 + \epsilon_i) \prod_{i=r+1}^{k}(1 + \epsilon_i)^{-1} - 1 \right| < k\beta'$$

Applying this lemma to equation (13) yields

$$|f_{ip}| \leq |n_{ip}|(i - p - 1)\beta' + \beta' \sum_{j=p+1}^{i-1} |n_{ij}||m^*_{jp}|(i - j + 2)$$

where

$$\beta' = \frac{1}{n}\left[\exp\left(\frac{n\beta_{sp}}{1 - \beta_{sp}}\right) - 1\right]$$

Let

$$\epsilon^*_{ip} \equiv |n_{ip}|(i - p - 1)\beta' + \beta' \sum_{j=p+1}^{i-1} |n_{ij}||m^*_{jp}|(i - j + 2) \tag{14}$$

Then

$$\left|f_{ip}\right| \leqq \epsilon^*_{ip}$$

Now

$$F = NM^* - I = N(M^* - M)$$

Therefore, for example,

$$f_{42} = m^*_{42} - m_{42}$$

Since $\left|f_{42}\right| < \epsilon^*_{42}$, $\left|m^*_{42} - m_{42}\right| < \epsilon^*_{42}$.

Define $\epsilon'_{42} \equiv \epsilon^*_{42}$. In general, for $p \geqq q + 2$,

$$f_{pq} = n_{pq} \cdot 0 + n_{p,q+1} \cdot 0 + \sum_{i=q+2}^{p-1} n_{pi}\left(m^*_{iq} - m_{iq}\right) + m^*_{pq} - m_{pq}$$

Therefore

$$m^*_{pq} - m_{pq} = f_{pq} - \sum_{i=q+2}^{p-1} n_{pi}\left(m^*_{iq} - m_{iq}\right)$$

Therefore

$$\left|m^*_{pq} - m_{pq}\right| < \epsilon^*_{pq} + \sum_{i=q+2}^{p-1} \left|n_{pi}\right| \epsilon'_{iq}$$

Define

$$\epsilon'_{pq} \equiv \epsilon^*_{pq} + \sum_{i=q+2}^{p-1} \left|n_{pi}\right| \epsilon'_{iq} \tag{15}$$

Thus, for $p + 1 > q$,

$$\left|m_{pq}\right| < \left|m^*_{pq}\right| + \epsilon'_{pq}$$

Therefore, equation (8) is still satisfied if substitution is made for the part of algorithm C that governs the calculation of $m_{r+1,i}$ and $b_{i,r+1}$ as follows:

For $i = 2,3,\ldots,r$

$$m^*_{r+1,i} = -fl\left(n_{r+1,i} + \sum_{j=i+1}^{r} n_{r+1,j}m^*_{ji}\right)$$

$$\epsilon^*_{r+1,i} = \left\{\beta'\left[\left|n_{r+1,i}\right|(r - i) + \sum_{j=i+1}^{r}\left|n_{r+1,j}m^*_{ji}\right|(r - j + 1)\right]\right\}(1 \oplus 0.01)$$

$$\epsilon'_{r+1,i} = \left(\epsilon^*_{r+1,i} + \sum_{j=i+2}^{r}\left|n_{r+1,j}\right|\epsilon'_{ji}\right)(1 \oplus 0.01)$$

$$m^*_{r+1,r+1} = 1$$

$$\epsilon'_{r+1,r+1} = 0$$

For $i = 1,2,\ldots,n$

$$b_{i,r+1} = \min_{2\leq j\leq r+1}\left\{\left[\frac{\delta a_{ij}}{R(n - j + 1)\left(\left|m^*_{r+1,j}\right| + \epsilon'_{r+1,j}\right)}\right](1 \ominus 0.01)\right\}$$

The calculation of $\left|m^*_{r+1,j}\right| + \epsilon'_{r+1,j}$ requires $3(r - i)$ multiplications and $4(r - i) + 1$ additions.

An upper bound on $|M|$ can also be calculated by using expanding-interval arithmetic. With this method an interval $\left[m^*_{ip} \pm \delta m_{ip}\right]$ is calculated by using the rules of expanding-interval arithmetic as described on page 390 in reference 1 or under the heading "Rounded Interval Arithmetic" on page 11 in reference 11 such that $m_{ip} \in \left[m^*_{ip} \pm \delta m_{ip}\right]$. To find an upper bound on $m_{r+1,j}$ by using this method requires $4(r - i)$ multiplications and $4(r - i) + 1$ additions. The $\delta m_{ip}$ calculated by using expanding-interval arithmetic satisfies the following inequality:

$$\delta m_{ip} \leq \epsilon'_{ip}\left[1 + \beta_{sp}(i - p - 1)\right]$$

where $\epsilon'_{ip}$ is defined by equation (15).

## Storage of Arrays

Although in the above development $N$, $H$, and $A$ are thought of as three separate matrices, there is no need to store them in the computer as separate matrices. Notice in algorithm C that after $a_{ir}$ is used in the calculation of $h_{ir}$ it is never used again. Therefore, $h_{ir}$ can be stored in the same location as $a_{ir}$. Also, after $a_{ir}$ is used to calculate $n_{i,r+1}$, $a_{ir}$ is not used again. Thus, $n_{i,r+1}$ can be stored in the same location as $a_{ir}$. If this scheme is used to store $N$, $H$, and $A$, only one set of $n \times n$ locations is needed to store all three matrices.

Single-precision arithmetic can be used in the calculation of $m^*_{ip}$ and $\epsilon'_{pq}$. Since $N$ is calculated by using variable-precision arithmetic, each $n_{ij}$ must be rounded to single precision before either $m^*_{ip}$ or $\epsilon'_{pq}$ can be calculated. Since the locations above the diagonal in the matrix $M^* = [m^*_{ij}]$ are not used, these locations can be used to store the elements of $N$ rounded to single precision.

Let $m^*_{ji} = n_{ij}(1 \oplus \beta_{sp})$ for $i > j$. With these changes algorithm C becomes

Algorithm D

For $i = 1,2,\ldots,n$

$$b_i = \frac{\delta a_{i1}}{R}(1 \ominus 0.01)$$

For $r = 1,2,\ldots,n$

If $r > 1$, then

For $i = 1,2,\ldots,r$

$$a_{ir} = \left(a_{ir} + \sum_{k=r+1}^{n} a_{ik}a_{k,r-1} - \sum_{k=2}^{i-1} a_{i,k-1}a_{kr}\right) \oplus b_i$$

If $n > r > 1$, then

For $i = r+1, r+2, \ldots, n$

$$a_{ir} = a_{ir} + \sum_{k=r+1}^{n} a_{ik}a_{k,r-1} - \sum_{k=2}^{r} a_{i,k-1}a_{kr}$$

If $r < n$, then

Find $j$ such that $|a_{jr}| \neq \max\left\{|a_{r+1,r}|, |a_{r+2,r}|, \ldots, |a_{nr}|\right\}$

If $a_{jr} \neq 0$, then

For $i = 1,2,\ldots,n$

Interchange $(a_{ji}, a_{r+1,i})$

Interchange $(a_{ij}, a_{i,r+1})$

Interchange $(\delta a_{ji}, \delta a_{r+1,i})$

Interchange $(\delta a_{ij}, \delta a_{i,r+1})$

$$a_{r+1,r} = a_{r+1,r} \oplus b_r$$

For $i = r+2, r+3, \ldots, n$

$$a_{ir} = \left(\frac{a_{ir}}{a_{r+1,r}}\right) \oplus b_i$$

For $i = r+2, r+3, \ldots, n$

$$m^*_{r+1,i} = a_{ir}(1 \oplus \beta_{sp})$$

For $i = 2,3,\ldots,r$

$$m^*_{r+1,i} = -fl\left(m^*_{i,r+1} + \sum_{j=i+1}^{r} m^*_{j,r+1} m^*_{ji}\right)$$

$$\epsilon'_{r+1,i} = \left\{\beta'\left[(r - i)\left|m^*_{i,r+1}\right| + \sum_{j=i+1}^{r}\left|m^*_{j,r+1} m^*_{ji}\right|(r - j + 1)\right]\right\}(1 \oplus 0.01)$$

$$\epsilon'_{r+1,i} = \left(\epsilon'_{r+1,i} + \sum_{j=i+2}^{r}\left|m^*_{j,r+1}\right|\epsilon'_{ji}\right)(1 \oplus 0.01)$$

$$m^*_{r+1,r+1} = 1$$

$$\epsilon'_{r+1,r+1} = 0$$

For $i = 1,2,\ldots,n$

$$b_i = \min_{2 \leq j \leq r+1} \left\{ \left[ \frac{\delta a_{ij}}{R(n - j + 1)\left( \left| m^*_{r+1,j} \right| + \epsilon'_{r+1,j} \right)} \right] \left(1 \ominus 0.01\right) \right\}$$

Determination of $\delta H$

Now that an algorithm for determining $N$ and $H$ has been developed, the interval half-width $\delta H$ must be determined so that equation (3) is satisfied. Let

$$m'_{pq} = \left| m^*_{pq} \right| + \epsilon'_{pq}$$

Let

$$M' = \left[ m'_{pq} \right]$$

Then

$$|M| < M'$$

The problem of finding $\delta H$ is solved in two steps. First, a $\delta Q$ is found such that

$$|N| \, \delta Q \leq \frac{R - 1}{R} \, \delta \tilde{A} \tag{16}$$

Then $\delta H$ is found such that

$$\delta H \, M' \leq \delta Q \tag{17}$$

Equations (16) and (17) imply that

$$|N| \, \delta H \, |M| \leq |N| \, \delta H \, M' \leq |N| \, \delta Q \leq \frac{R - 1}{R} \, \delta \tilde{A}$$

Therefore, if equations (16) and (17) are satisfied, then equation (3) is satisfied.

Let $\delta Q = \left[ \delta q_{ij} \right]$. Because of the form of $\delta H \, M'$, $\delta q_{i1} = 0$ for $i \geq 3$. Equation (16) is satisfied if

$$\delta q_{1i} \leq \frac{R - 1}{R} \, \delta a_{1i}$$

$$\delta q_{21} \, n_{i2} \leq \frac{R - 1}{R} \, \delta a_{i1} \qquad (i = 2,3,\ldots,n)$$

and

$$\sum_{k=2}^{i} |n_{ik}| \, \delta q_{kj} \leq \frac{R - 1}{R} \, \delta a_{ij} \qquad (i,j > 1)$$

This last equation is satisfied if

$$|n_{ik}| \, \delta q_{kj} \leq \frac{R - 1}{R} \frac{\delta a_{ij}}{i - 1} \qquad (2 \leq k \leq i)$$

Therefore, if $j > 1$ and $k > 1$, define

$$\delta q_{kj} \equiv \frac{R - 1}{R} \min_{i \geq k} \left\{ \frac{\delta a_{ij}}{|n_{ik}|(i - 1)} \right\} \tag{18}$$

Define

$$\delta q_{1i} \equiv \frac{R - 1}{R} \, \delta a_{1i} \qquad (i = 1,2,\ldots,n) \tag{19}$$

and

$$\delta q_{21} \equiv \frac{R - 1}{R} \min_{i \geq 2} \left\{ \frac{\delta a_{i1}}{|n_{i2}|} \right\} \tag{20}$$

With this definition of $\delta Q$, equation (16) is satisfied.

Notice in equation (18) the quotients $\frac{\delta a_{ij}}{i - 1}$ can be calculated beforehand and stored in the locations reserved for $\delta a_{ij}$.

If $\delta H$ is to satisfy equation (17) then the following equations must be satisfied:

$$\delta h_{11} \leq \delta q_{11}$$

$$\delta h_{21} \leq \delta q_{21}$$

and

$$\sum_{k=\max\{j,i-1\}}^{n} \delta h_{ik} \, m'_{kj} \leq \delta q_{ij} \qquad (j > 1)$$

This last equation is satisfied if

$$\delta h_{ik} \leqq \frac{\delta q_{ij}}{\left(n - \max\{j,i-1\} + 1\right) m'_{kj}} \qquad \left(\max\{j,i-1\} \leqq k \leqq n\right)$$

Therefore, define

$$\delta h_{ik} \equiv \min_{1<j\leqq k} \left\{ \frac{\delta q_{ij}}{\left(n - \max\{j,i-1\} + 1\right) m'_{kj}} \right\} \qquad (k > 1) \qquad (21)$$

and

$$\delta h_{i1} \equiv \delta q_{i1} \qquad (i = 1 \text{ or } 2)$$

With this definition of $\delta H$, equation (17) is satisfied.

The quotients $\dfrac{\delta q_{ij}}{n - \max\{j,i-1\} + 1}$ can be calculated beforehand and stored in the location reserved for $\delta a_{ij}$. Once $\delta q_{ij}$ is determined, it can be stored in the same location as $\delta a_{ij}$ if the elements of $\delta Q$ are determined in the sequence shown in the following algorithm:

For $j = 2,3,\ldots,n$

  For $k = 2,3,\ldots,n$

    $$\delta q_{kj} = \frac{R-1}{R} \min_{i\geqq k} \left\{ \frac{\delta a_{ij}}{n_{ik}(i-1)} \right\}$$

The $\delta h_{ik}$ can be stored in the same location as $\delta q_{ik}$ if the elements of $\delta H$ are determined in the sequence shown in the following algorithm:

For $k = n,n-1,\ldots,2$

  For $i = 1,2,\ldots,\min\{k+1,n\}$

    $$\delta h_{ik} = \min_{1<j\leqq k} \left\{ \frac{\delta q_{ij}}{\left(n - \max\{j,i-1\} + 1\right) m'_{kj}} \right\}$$

In calculating $\delta H$ and $\delta Q$, it is important that the calculated values be no larger than the exact values. Putting the above ideas together gives the following algorithm for obtaining $\delta H$:

Algorithm E

$$R' = \frac{R - 1}{R}$$

For $i = 2,3,\ldots,n$

$$i' = i - 1$$

For $j = 2,3,\ldots,n$

$$\delta a_{ij} = \left(\frac{\delta a_{ij}}{i'}\right)(1 \ominus 0.001)$$

$$\delta a_{1i} = \left(\frac{R' \, \delta a_{1i}}{(n - i + 1)}\right)(1 \ominus 0.001)$$

$$\delta a_{11} = (R' \, \delta a_{11})(1 \ominus 0.001)$$

$$\delta a_{21} = \left(R' \min_{i \geq 2}\left\{\frac{\delta a_{i1}}{|n_{i2}|}\right\}\right)(1 \ominus 0.001)$$

For $j = 2,3,\ldots,n$

For $k = 2,3,\ldots,n$

$$\delta a_{kj} = \left[\frac{R' \min_{i \geq k}\left\{\frac{\delta a_{ij}}{|n_{ik}|}\right\}}{\left(n - \max\{j, k - 1\} + 1\right)}\right](1 \ominus 0.001)$$

For $k = n,n-1,\ldots,2$

For $i = 1,2,\ldots,\min\{k+1,n\}$

$$\delta a_{ik} = \left(\min_{1 < j \leq k}\left\{\frac{\delta a_{ij}}{m'_{kj}}\right\}\right)(1 \ominus 0.001)$$

After execution of algorithm E, $\delta h_{ik}$ is stored in the location initially reserved for $\delta a_{ik}$.

This completes the discussion of how to obtain $H$ and $\delta H$.

## CHAPTER III

## REDUCTION OF AN INTERVAL OF HESSENBERG MATRICES TO AN INTERVAL OF COLLEAGUE MATRICES

Chapter III is devoted to the reduction of an interval of Hessenberg matrices to an interval of colleague matrices by a similarity transformation. Where a colleague matrix is a matrix of the form

$$F = \begin{bmatrix} \beta_1 & \alpha_1 & & & -a_n \\ 1 & \beta_2 & \alpha_2 & & -a_{n-1} \\ & 1 & \beta_3 & & -a_{n-2} \\ & & \ddots & \ddots & \vdots \\ & & & \alpha_{n-1} & -a_2 \\ & & & 1 & \beta_n - a_1 \end{bmatrix}$$

The interval of colleague matrices is of the form $[F \pm \delta F]$ where

$$\delta F = \begin{bmatrix} 0 & \cdots & 0 & \delta a_n \\ 0 & \cdots & 0 & \delta a_{n-1} \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & \delta a_1 \end{bmatrix}$$

### A Property of F

The matrix F has the property that

$$\det(\lambda I - F) = \sum_{i=0}^{n-1} a_{n-i} P_i(\lambda) + P_n(\lambda) \tag{22}$$

where the $P_i(\lambda)$ are defined by $P_i(\lambda) = \det(\lambda I - B_i)$ and

$$B_i = \begin{bmatrix} \beta_1 & \alpha_1 & & & & \\ 1 & \beta_2 & \alpha_2 & & & \\ & 1 & \beta_3 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \alpha_{i-1} \\ & & & & 1 & \beta_i \end{bmatrix}$$

The proof of this uses the following theorem:

Theorem 2 – The $P_i(\lambda)$ satisfy the recurrence relationship

$$P_{i+1}(\lambda) = (\lambda - \beta_{i+1})\, P_i(\lambda) - \alpha_i\, P_{i-1}(\lambda) \tag{23}$$

with $P_0 = 1$ and $P_1 = \lambda - B_1$.

Proof – $P_r(\lambda) = \det(\lambda I - B_r)$

$$= \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & & \\ -1 & \lambda - \beta_2 & -\alpha_2 & & & \\ & -1 & \lambda - \beta_3 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -\alpha_{r-1} \\ & & & & -1 & \lambda - \beta_r \end{bmatrix}$$

$$= (\lambda - \beta_r)\ \det(\lambda I - B_{r-1})$$

$$+ \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & & & \\ -1 & \lambda - \beta_2 & -\alpha_2 & & & & \\ & -1 & \ddots & \ddots & & & \\ & & \ddots & \ddots & -\alpha_{r-3} & & \\ & & & -1 & \lambda - \beta_{r-2} & 0 & \\ & & & & -1 & -\alpha_{r-1} & \end{bmatrix}$$

$$= (\lambda - \beta_r)\ \det(\lambda I - B_{r-1}) - \alpha_{r-1}\ \det(\lambda I - B_{r-2})$$

$$+ \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & & & \\ -1 & \lambda - \beta_2 & -\alpha_2 & & & & \\ & -1 & \ddots & \ddots & & & \\ & & \ddots & \ddots & -\alpha_{r-4} & & \\ & & & -1 & \lambda - \beta_{r-3} & 0 & \\ & & & & -1 & 0 & \\ & & & & & -1 & 0 \end{bmatrix}$$

$$= (\lambda - \beta_r)\ P_{r-1}(\lambda) - \alpha_{r-1}\ P_{r-2}(\lambda)$$

since the last term is zero.

Now it may be proved that equation (22) is valid.

Theorem 3 – $\det(\lambda I - F) = \sum_{i=0}^{n-1} a_{n-i}\ P_i(\lambda) + P_n(\lambda)$

Proof –

$$\det(\lambda I - F) = \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & a_n \\ -1 & \lambda - \beta_2 & -\alpha_2 & & a_{n-1} \\ & -1 & \ddots & \ddots & a_{n-2} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \vdots \\ & & & -1 & \lambda - \beta_{n-1} \quad a_2 - \alpha_{n-1} \\ & & & & -1 \quad \lambda - \beta_n + a_1 \end{bmatrix}$$

$$= (-1)^{n+1} a_n \det \begin{bmatrix} -1 & \lambda - \beta_2 & -\alpha_2 & & \\ & -1 & \lambda - \beta_3 & -\alpha_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & -\alpha_{n-2} \\ & & & & \lambda - \beta_{n-1} \\ & & & & -1 \end{bmatrix}$$

$$+ (-1)^{n+2} a_{n-1} \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & \\ & -1 & \lambda - \beta_3 & -\alpha_3 & \\ & & -1 & \lambda - \beta_4 & \ddots \\ & & & \ddots & \ddots \\ & & & & -\alpha_{n-2} \\ & & & & \lambda - \beta_{n-1} \\ & & & & -1 \end{bmatrix}$$

$$+ \ldots + (-1)(-\alpha_{n-1} + a_2) \det \begin{bmatrix} \lambda - \beta_1 & -\alpha_1 & & & \\ -1 & \lambda - \beta_2 & -\alpha_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & & -1 & \lambda - \beta_{n-2} \quad -\alpha_{n-2} \\ & & & & 0 \quad -1 \end{bmatrix}$$

$$+ (\lambda - \beta_n + a_1)\, P_{n-1}(\lambda)$$

Therefore

$$\begin{aligned}\det(\lambda I - F) &= a_n + a_{n-1}P_1 + a_{n-2}P_2 + \cdots \\ &\quad + (-\alpha_{n-1} + a_2)P_{n-2} + (\lambda - \beta_n + a_1)P_{n-1} \\ &= \sum_{i=0}^{n-1} a_{n-i}P_i - \alpha_{n-1}P_{n-2} + (\lambda - \beta_n)P_{n-1} \\ &= \sum_{i=0}^{n-1} a_{n-i}P_i + P_n\end{aligned}$$

Because of the recurrence relation in equation (23), the polynomials $P_i$ can be made to be the first $n$ polynomials of any desired orthogonal set of polynomials by properly selecting the $\alpha_i$ and the $\beta_i$.

## Algorithm for Reducing Hessenberg Matrix to Colleague Form by Similarity Transformation

Now in returning to the problem of reducing the interval $[H \pm \delta H]$ to an interval of colleague matrices, it is desirable to determine $V$ and $a_1, a_2, \ldots, a_n$ such that

$$VH = FV \tag{24}$$

where $V$ is of the form

$$V = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ & V_{22} & & V_{2n} \\ & & \ddots & \vdots \\ & & & V_{nn} \end{bmatrix}$$

For $j < n$ the ijth element of equation (24) is

$$\sum_{k=i}^{j+1} V_{ik}h_{kj} = V_{i-1,j} + \beta_i V_{ij} + \alpha_i V_{i+1,j} \tag{25}$$

where

$$V_{0j} = V_{n+1,n} = 0$$

and

$$V_{ij} = 0 \qquad (i > j)$$

For $j = n$, equation (24) gives

$$\sum_{k=i}^{n} V_{ik} h_{kn} = V_{i-1,n} + \beta_i V_{in} + \alpha_i V_{i+1,n} - a_{n-i+1} V_{nn} \tag{26}$$

Define $V_{i,n+1} \equiv a_{n-i+1} V_{nn}$ $(i = 1,2,\ldots,n)$ and $V_{n+1,n+1} \equiv V_{nn}$. The characteristic polynomial of $F$ can then be written

$$\begin{aligned} \det(\lambda I - F) &= P_n(\lambda) + \sum_{i=0}^{n-1} a_{n-i} P_i(\lambda) \\ &= \frac{1}{V_{nn}} \sum_{i=0}^{n} V_{i+1,n+1} P_i(\lambda) \end{aligned} \tag{27}$$

Thus, the problem of finding $a_1, a_2, \ldots, a_n$ is equivalent to finding an additional column for the matrix $V$.

Equation (25) yields

$$V_{i,j+1} = \frac{V_{i-1,j} + \beta_i V_{ij} + \alpha_i V_{i+1,j} - \sum_{k=i}^{j} V_{ik} h_{kj}}{h_{j+1,j}} \tag{28}$$

Equation (26) yields

$$V_{i,n+1} = V_{i-1,n} + \beta_i V_{in} + \alpha_i V_{i+1,n} - \sum_{k=i}^{n} V_{ik} h_{kj} \tag{29}$$

It can be assumed that no $h_{j+1,j}$ is zero since if it is the problem can be treated as two smaller problems.

Equations (28) and (29) yield the following algorithm for the calculation of $V$:

Algorithm F

$h_{n+1,n} = 1$

$V_{11} = 1$

For $j = 2,3,\ldots,n+1$

  For $i = 1,2,\ldots,\min\{j,n\}$

$$V_{ij} = \frac{-\sum_{k=i}^{j-1} V_{ik}h_{k,j-1} + V_{i-1,j-1} + \beta_i V_{i,j-1} + \alpha_i V_{i+1,j-1}}{h_{j,j-1}} + b_{i,j-1}$$

$V_{n+1,n+1} = V_{nn}$

The $b_{ij}$ are the rounding errors incurred in the execution of the algorithm. Let

$$K = \begin{bmatrix} h_{21} & & & \\ & h_{32} & & \\ & & \ddots & \\ & & & h_{n+1,n} \end{bmatrix}$$

Then

$$VH = FV + BK \tag{30}$$

Obtaining Midpoint of Interval of Colleague Matrices

Since $H$ is the midpoint of the interval $[H \pm \delta H]$, it is desirable that $F$ be the midpoint of the interval of colleague matrices. How small $BK$ must be and how large $\delta F$ can be must be determined.

Let $F'$ be a colleague matrix; then $H' = V^{-1}F'V$ is a Hessenberg matrix, where $V$ is the same as in equation (30). Subtracting this from equation (30) gives

$$V(H - H') = (F - F')V + BK$$

or

$$H - H' = V^{-1}(F - F')V + V^{-1}BK$$

It is required that $|H - H'| < \delta H$, which is true if $|V^{-1}BK| < \frac{\delta H}{R}$ and $|V^{-1}(F - F')V| < \frac{R - 1}{R}\delta H$.

Define $W \equiv V^{-1}$. Then it is necessary that $|WBK| < \frac{\delta H}{R}$, which is true if

$$|b_{ij}| < \frac{1}{R} \min_{k \leq i} \left\{ \left| \frac{\delta h_{kj}}{(j - k + 2) w_{ki} h_{j+1,j}} \right| \right\} \tag{31}$$

The proof of this is on page 18 in reference 6.

A bound on $|b_{i,j-1}|$ must be known before $V_{ij}$ can be calculated. Therefore, from equation (31) it can be seen that the ith column of $W$ must be known before the ith row of $V$ can be calculated. There is no direct method of calculating $W$ from $V$ so that the ith column of $W$ is available for the calculation of $V_{ii}$. However, $W$ can be computed independently of $V$ by the following method:

If $VH = FV$ then $HW = WF$. This means that for $j \leq n - 1$ and $i \leq j + 1$

$$\sum_{k=\max\{1,i-1\}}^{j} h_{ik} w_{kj} = w_{i,j-1}\alpha_{j-1} + w_{ij}\beta_j + w_{i,j+1}$$

where $w_{ij} = 0$ for $i > j$ and $w_{i0} = 0$. This equation can be solved for $w_{i,j+1}$. Thus, $W$ can be determined by the following algorithm:

$w_{11} = 1$

For $j = 2,3,\ldots,n$

  For $i = 1,2,\ldots,j$

$$w_{ij} = \sum_{k=\max\{1,i-1\}}^{j-1} h_{ik} w_{k,j-1} - w_{i,j-2}\alpha_{j-2} - w_{i,j-1}\beta_{j-1}$$

To use the $W$ determined by this algorithm in equation (31) it is required that $|W| \geq |V^{-1}|$ but, since the computation of $W$ is done independently of the computation of $V$, this is not necessarily true. If interval arithmetic is used in the calculation of $W$

then it can be guaranteed that $|W|$ exceeds $|W'|$ for any $W'$ corresponding exactly to any $H' \in [H \pm \delta H]$; that is, $H'W' = W'F'$.

However, if $V$ is calculated by using algorithm A and if the rounding errors are required to satisfy equation (31) with the computed $W$ how can it be guaranteed that this $V$ corresponds exactly to some $H' \in [H \pm \delta H]$?

The answer to this question is contained in the following theorem which is taken from reference 6, page 23:

Theorem – If $W$ is an upper triangular matrix such that $|W| \geqq |W'|$ for every $W'$ corresponding to an $H'$ in the interval $[H \pm \delta H]$ and if equation (31) is observed in the computation of $V$, that is, if $|W||B||K| < \frac{\delta H}{R}$, then the $H'$ which corresponds to the computed $V$ lies in $[H \pm \delta H]$.

Thus, $V^{-1}$ corresponds exactly to $H' \in [H \pm \delta H]$. Therefore, $|W| \geqq |V^{-1}|$. Now, since a bound on $|V^{-1}|$ is known, a bound on $|b_{ij}|$ can be calculated by using equation (31). Thus the following algorithm is obtained for the determination of $V$:

Algorithm G

$V_{11} = 1$

$w_{11} = 1$

$h_{n+1,n} = 1$

For $j = 2,3,\ldots,n+1$

For $i = 1,2,\ldots,\min\{j,n\}$

If $j \leqq n$

$$\left[w_{ij} \pm \delta w_{ij}\right] = \sum_{k=\max\{1,i-1\}}^{j-1} h_{ik}\left[w_{k,j-1} \pm \delta w_{k,j-1}\right] - \left[w_{i,j-2} \pm \delta w_{i,j-2}\right]\alpha_{j-2} - \left[w_{i,j-1} \pm \delta w_{i,j-1}\right]\beta_{j-1}$$

$$\xi = \left[\frac{1}{R} \min_{k \leqq i}\left\{\frac{\delta h_{k,j-1}}{(j-k+1)\left(|w_{ki}| + \delta w_{ki}\right)|h_{i,j-1}|}\right\}\right](1 \ominus 0.01)$$

$$V_{ij} = \left( \frac{-\sum_{k=1}^{j-1} V_{ik} h_{k,j-1} + V_{i-1,j-1} + \beta_i V_{i,j-1} + \alpha_i V_{i+1,j-1}}{h_{j,j-1}} \right) \oplus \xi$$

$V_{n+1,n+1} = V_{nn}$

Interval arithmetic is used in the calculation of $\left[w_{ij} \pm \delta w_{ij}\right]$.

Obtaining Interval Half-Width

In the problem of determining how large to make $\delta F$, the constraint on $\delta F$ is that

$$\left|V^{-1}(F - F')V\right| < \frac{R - 1}{R} \delta H$$

if $\left|F - F'\right| < \delta F$. This is true if

$$\left|W\right| \left|F - F'\right| \left|V\right| < \frac{R - 1}{R} \delta H \tag{32}$$

$$\left|F - F'\right| = \begin{bmatrix} 0 & 0 & \cdots & 0 & \left|a_n - a'_n\right| \\ 0 & 0 & \cdots & 0 & \left|a_{n-1} - a'_{n-1}\right| \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \left|a_1 - a'_1\right| \end{bmatrix}$$

Therefore

$$\left|F - F'\right| \left|V\right| = \begin{bmatrix} 0 & 0 & \cdots & 0 & \left|V_{nn}\right| \left|a_n - a'_n\right| \\ 0 & 0 & \cdots & 0 & \left|V_{nn}\right| \left|a_{n-1} - a'_{n-1}\right| \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \left|V_{nn}\right| \left|a_1 - a'_1\right| \end{bmatrix} \tag{33}$$

Define $V'_{i,n+1} \equiv a'_{n-i+1} V_{nn}$ and $\delta V_i \equiv \left|V_{i,n+1} - V_{i,n+1}\right|$.

Then

$$\begin{aligned} |V_{nn}|\,|a_{n-i+1} - a'_{n-i+1}| &= |V_{nn}a_{n-i+1} - V_{nn}a'_{n-i+1}| \\ &= |V_{i,n+1} - V'_{i,n+1}| \\ &= \delta V_i \end{aligned} \tag{34}$$

Up to this point it has been assumed that an interval $[F \pm \delta F]$ would be obtained, but the algorithm determines $V_{i,n+1}$ instead of $a_{n-i+1}$ and it is easier to obtain a bound on $\delta V_i$ than on $|a_{n-i+1} - a'_{n-i+1}|$. Thus, a bound $\overline{\delta V_i}$ is found such that if $|\delta V_i| \leqq \overline{\delta V_i}$ then equation (32) is satisfied. Therefore, n intervals $\left[V_{i,n+1} \pm \overline{\delta V_i}\right]$ $(i = 1,2,\ldots,n)$ are found such that if $V'_{i,n+1} \in \left[V_{i,n+1} \pm \overline{\delta V_i}\right]$ $(i = 1,2,\ldots,n)$ then the polynomial $V_{n+1,n+1}\,P_n(\lambda) + \sum_{i=1}^{n} V'_{i,n+1}\,P_{i-1}(\lambda)$ is the characteristic polynomial of some matrix $H' \in [H \pm \delta H]$. This is true because, since equations (32) and (31) are satisfied, it is implied that $P_n(\lambda) + \sum_{i=0}^{n} a'_{n-i}\,P_i(\lambda)$ is the characteristic polynomial of some $H' \in [H \pm \delta H]$, but

$$V_{n+1,n+1}\,P_n(\lambda) + \sum_{i=1}^{n} V'_{i,n+1}\,P_{i-1}(\lambda) = V_{nn}\left[P_n(\lambda) + \sum_{i=0}^{n-1} a'_{n-i}\,P_i(\lambda)\right]$$

From equations (33) and (34) it can be seen that

$$|F - F'|\,|V| = \begin{bmatrix} 0 & \cdots & 0 & \delta V_1 \\ 0 & \cdots & 0 & \delta V_2 \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \delta V_n \end{bmatrix}$$

Therefore

$$|W|\,|F - F'|\,|V| = \begin{bmatrix} 0 & \cdots & 0 & \sum_{i=1}^{n} \delta V_i\,|w_{1i}| \\ 0 & \cdots & 0 & \sum_{i=2}^{n} \delta V_i\,|w_{2i}| \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & |w_{nn}|\,\delta V_n \end{bmatrix}$$

Thus, if equation (32) is to be satisfied, it must follow that

$$\sum_{i=j}^{n} \delta V_i\,|w_{ji}| \leqq \frac{R-1}{R}\,\delta h_{jn} \qquad (j = 1,2,\ldots,n)$$

This is true if

$$\delta V_i\,|w_{ji}| \leqq \frac{R-1}{R}\,\frac{\delta h_{jn}}{n-j+1} \qquad (i = j,j+1,\ldots,n;\ j = 1,2,\ldots,n)$$

that is, if

$$\delta V_i \leqq \frac{R-1}{R}\,\frac{\delta h_{jn}}{(n-j+1)|w_{ji}|} \qquad (j = 1,2,\ldots,i)$$

Let

$$\overline{\delta V_i} \equiv \frac{R-1}{R}\,\min_{j \leqq i}\left\{\frac{\delta h_{jn}}{(n-j+1)|w_{ji}|}\right\} \tag{35}$$

Then $\delta V_i \leqq \overline{\delta V_i}$ $(i = 1,2,\ldots,n)$ implies that equation (32) is satisfied.

Of course, if $F$ and $\delta F$ are needed they can very easily be found from $V_{i,n+1}$ and $\overline{\delta V_i}$ $(i = 1,2,\ldots,n)$.

## CHAPTER IV

## FACTORING THE CHARACTERISTIC POLYNOMIAL

In chapter III $V_{n+1,n+1}$ and intervals $\left[V_{i,n+1} \pm \overline{\delta V_i}\right]$ $(i = 1,2,\ldots,n)$ were obtained such that if $V_i' \in \left[V_{i,n+1} \pm \overline{\delta V_i}\right]$ $(i = 1,2,\ldots,n)$ then

$$V_{n+1,n+1}\, P_n(\lambda) + \sum_{i=1}^{n} V_i'\, P_{i-1}(\lambda)$$

is the characteristic polynomial of some matrix $H' \in [H \pm \delta H]$, where the $P_i$ satisfy the recurrence relation in equation (23) of chapter III.

In chapter IV a method is developed for factoring an interval polynomial into a product of interval quadratic polynomials. The notation may be simplified by assuming that the interval polynomial to be factored is $[P \pm \delta P]$ where

$$P(x) = a_n P_0 + a_{n-1} P_1 + \ldots + a_1 P_{n-1} + a_0 P_n$$

and

$$\delta P(x) = \delta a_n P_0 + \delta a_{n-1} P_1 + \ldots + \delta a_1 P_{n-1}$$

Also, it is assumed that the $\alpha_i$ are all equal and the $\beta_i$ are all equal; that is,

$$\alpha_i = \alpha \qquad (i = 1,2,\ldots,n-1)$$

and

$$\beta_i = \beta \qquad (i = 1,2,\ldots,n)$$

Given two intervals $[C \pm \delta C]$ and $[D \pm \delta D]$, define the product $[C \pm \delta C][D \pm \delta D] = \{C'D' | C' \in [C \pm \delta C]$ and $D' \in [D \pm \delta D]\}$.

In obtaining a quadratic factor of $[P \pm \delta P]$, first

$$Q = P_2 - sP_1 - tP_0$$

and

$$T = b_0 P_{n-2} + \cdots + b_{n-2} P_0$$

are found such that

$$QT \in \left[P \pm \frac{\delta P}{R}\right]$$

Then $\delta Q$ and $\delta T$ are found such that

$$[Q \pm \delta Q][T \pm \delta T] \subseteq \left[QT \pm \frac{R - 1}{R}\, \delta P\right]$$

that is,

$$[Q \pm \delta Q][T \pm \delta T] \subseteq [P \pm \delta P]$$

The same method applied to $[P \pm \delta P]$ can then be applied to $[T \pm \delta T]$ to obtain another quadratic factor.

Obtaining Interval Midpoints

Given estimates for $s$ and $t$, $P(x)$ can be factored in the following manner:

$$P(x) = \left(P_2 - sP_1 - tP_0\right)\left(b_0 P_{n-2} + \cdots + b_{n-3} P_1 + b_{n-2} P_0\right) + b_{n-1}\left(P_1 - sP_0\right) + b_n P_0 \tag{36}$$

To develop a method for factoring $P(x)$ as shown in equation (36) the following lemmas are needed:

Lemma 1 – If $i \geq j \geq 0$ then

$$P_j P_i = \sum_{k=0}^{j} \alpha^{j-k} P_{i+2k-j}$$

Proof – For $j = 0$ and $i \geq 0$, $P_0 P_i = P_i$. Thus the lemma is true for $j = 0$. Now, assume

$$P_j P_i = \sum_{k=0}^{j} \alpha^{j-k} P_{i+2k-j}$$

for $i \geq j$ if $j = 1,2,\ldots,m-1$. For $i \geq m$

$$P_m P_i = \left[(x - \beta)P_{m-1} - \alpha P_{m-2}\right]P_i$$

$$= (x - \beta) \sum_{k=0}^{m-1} \alpha^{m-1-k} P_{i+2k-m+1} - \alpha \sum_{k=0}^{m-2} \alpha^{m-2-k} P_{i+2k-m+2}$$

$$= (x - \beta) \sum_{k=1}^{m-1} \alpha^{m-1-k} P_{i+2k-m+1} - \alpha \sum_{k=1}^{m-1} \alpha^{m-1-k} P_{i+2k-m}$$

$$+ (x - \beta)\alpha^{m-1} P_{i-m+1}$$

$$= \sum_{k=1}^{m-1} \alpha^{m-1-k}\left[(x - \beta)P_{i+2k-m+1} - \alpha P_{i+2k-m}\right] + (x - \beta)\alpha^{m-1} P_{i-m+1}$$

$$= \sum_{k=1}^{m-1} \alpha^{m-1-k} P_{i+2k-m+2} + (x - \beta)\alpha^{m-1} P_{i-m+1}$$

$$= \sum_{k=2}^{m} \alpha^{m-k} P_{i+2k-m} + \alpha^{m-1}\left(P_{i-m+2} + \alpha P_{i-m}\right)$$

Therefore

$$P_m P_i = \sum_{k=0}^{m} \alpha^{m-k} P_{i+2k-m}$$

Lemma 2 – $P(x)$ can be factored as shown in equation (36) by using algorithm H.

Proof – Equation (36) yields

$$P(x) = b_0 P_2 P_{n-2} + b_1 P_2 P_{n-3} + b_2 P_2 P_{n-4} + \cdots + b_{n-2} P_0 P_2 + b_{n-1} P_1 + b_n P_0$$

$$- sb_0 P_1 P_{n-2} - sb_1 P_1 P_{n-3} - \cdots - sb_{n-3} P_1 P_1 - sb_{n-2} P_1 - sb_{n-1} P_0$$

$$- tb_0 P_0 P_{n-2} - \cdots - tb_{n-4} P_0 P_2 - tb_{n-3} P_0 P_1 - tb_{n-2} P_0$$

From lemma 1

$$P_2P_i = P_{i+2} + \alpha P_i + \alpha^2 P_{i-2}$$

and

$$P_1P_i = P_{i+1} + \alpha P_{i-1}$$

Therefore

$$\begin{aligned} P(x) = {} & P_n(b_0) + P_{n-1}(b_1 - sb_0) + P_{n-2}(b_0\alpha + b_2 - sb_1 - tb_0) \\ & + P_{n-3}\left[\alpha(b_1 - sb_0) + b_3 - sb_2 - tb_1\right] \\ & + P_{n-4}\left[b_0\alpha^2 + \alpha(b_2 - sb_1) + b_4 - sb_3 - tb_2\right] \\ & + P_{n-5}\left[b_1\alpha^2 + \alpha(b_3 - sb_2) + b_5 - sb_4 - tb_3\right] \\ & + \ldots + P_1\left[b_{n-5}\alpha^2 + \alpha(b_{n-3} - sb_{n-4}) + b_{n-1} - sb_{n-2} - tb_{n-3}\right] \\ & + P_0\left(\alpha^2 b_{n-4} - \alpha s b_{n-3} + b_n - sb_{n-1} - tb_{n-2}\right) \end{aligned}$$

Since the polynomials $P_i$ are linearly independent, their coefficients can be equated in the above equation. Thus

$$a_0 = b_0$$

$$a_1 = b_1 - sb_0$$

$$a_2 = b_0\alpha + b_2 - sb_1 - tb_0$$

$$a_3 = \alpha(b_1 - sb_0) + b_3 - sb_2 - tb_1$$

and for $n - 1 \geqq i \geqq 4$

$$a_i = \alpha^2 b_{i-4} + \alpha(b_{i-2} - sb_{i-3}) + b_i - sb_{i-1} - tb_{i-2}$$

and

$$a_n = \alpha^2 b_{n-4} - s\alpha b_{n-3} + b_n - sb_{n-1} - tb_{n-2}$$

Thus the $b_i$ can be calculated by the following algorithm:

Algorithm H

$b_0 = a_0$

$b_1 = a_1 + sb_0$

$b_2 = a_2 + sb_1 + tb_0 - \alpha b_0$

$b_3 = a_3 + sb_2 + tb_1 - \alpha(b_1 - sb_0)$

For $i = 4,5,\ldots,n-1$

$$b_i = a_i + sb_{i-1} + tb_{i-2} - \alpha(b_{i-2} - sb_{i-3}) - \alpha^2 b_{i-4}$$

$b_n = a_n + sb_{n-1} + tb_{n-2} + \alpha sb_{n-3} - \alpha^2 b_{n-4}$

This completes the proof of lemma 2.

Let $u(s,t) = b_{n-1}$ and $v(s,t) = b_n$. If $P_2 - sP_1 - tP_0$ is to be a quadratic factor of $P(x)$ then it is necessary that $u(s,t) = 0$ and $v(s,t) = 0$. The problem of determining $s$ and $t$ such that this is true is complicated by the fact that it is not practical to execute algorithm H exactly. Thus, actually, approximations $u^*$ and $v^*$ to $u$ and $v$ are calculated as in the following algorithm:

Algorithm I

$b_0^* = a_0$

$b_1^* = a_1 + sb_0^* + \psi_1$

$b_2^* = a_2 + sb_1^* + tb_0^* - \alpha b_0^* + \psi_2$

$b_3^* = a_3 + sb_2^* + tb_1^* - \alpha(b_1^* - sb_0^*) + \psi_3$

For $i = 4,5,\ldots,n-1$

$$b_i^* = a_i + sb_{i-1}^* + tb_{i-2}^* - \alpha\left(b_{i-2}^* - sb_{i-3}^*\right) - \alpha^2 b_{i-4}^* + \psi_i$$

$$b_n^* = a_n + sb_{n-1}^* + tb_{n-2}^* + \alpha sb_{n-3}^* - \alpha^2 b_{n-4}^* + \psi_n$$

$$u^*(s,t) = b_{n-1}^*$$

$$v^*(s,t) = b_n^*$$

where the $\psi_i$ are rounding errors. Let $a_i' = a_i + \psi_i$. Then

$$a_0P_n + a_1'P_{n-1} + \ldots + a_n'P_0 = \left(P_2 - sP_1 - tP_0\right)\left(b_0^*P_{n-2} + \ldots + b_{n-3}^*P_1 + b_{n-2}^*P_0\right) + b_{n-1}^*\left(P_1 - sP_0\right) + b_n^*P_0$$

This can be rewritten as

$$P'(x) = \left(P_2 - sP_1 - tP_0\right)\left(b_0^*P_{n-2} + \ldots + b_{n-3}^*P_1 + b_{n-2}^*P_0\right)$$

where

$$P'(x) = a_0P_n + a_1'P_{n-1} + \ldots + a_{n-2}'P_2 + P_1\left(a_{n-1}' - b_{n-1}^*\right) + P_0\left(a_n' - b_n^* + sb_{n-1}^*\right)$$

Then

$$P' \in \left[P \pm \frac{\delta P}{R}\right]$$

if

$$\left|a_i' - a_i\right| \leq \frac{\delta a_i}{R} \qquad (i = 1,2,\ldots,n-2)$$

$$\left|a_{n-1}' - b_{n-1}^* - a_{n-1}\right| \leq \frac{\delta a_{n-1}}{R}$$

and

$$\left|a_n' - b_n^* + sb_{n-1}^* - a_n\right| \leq \frac{\delta a_n}{R}$$

These equations are satisfied if

$$|\psi_i| \leqq \frac{\delta a_i}{R} \qquad (i = 1,2,\ldots,n-2)$$

$$|\psi_{n-1}| \leqq \frac{\delta a_{n-1}}{2R}$$

$$|\psi_n| \leqq \frac{\delta a_n}{2R}$$

$$|b^*_{n-1}| \leqq \frac{\delta a_{n-1}}{2R}$$

and

$$|b^*_n - sb^*_{n-1}| \leqq \frac{\delta a_n}{2R}$$

The above equations can be checked to determine if improved estimates $s$ and $t$ are needed. New estimates $s_N$ and $t_N$ can be calculated by using Newton's method; that is,

$$\begin{bmatrix} s_N \\ t_N \end{bmatrix} = \begin{bmatrix} s \\ t \end{bmatrix} + \begin{bmatrix} \Delta s \\ \Delta t \end{bmatrix}$$

where

$$\begin{bmatrix} \frac{\partial u}{\partial s} & \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial s} & \frac{\partial v}{\partial t} \end{bmatrix}_{s,t} \begin{bmatrix} \Delta s \\ \Delta t \end{bmatrix} = -\begin{bmatrix} u \\ t \end{bmatrix}_{s,t}$$

Approximations $\frac{\partial u^*}{\partial s}$ and $\frac{\partial v^*}{\partial s}$ to $\frac{\partial u}{\partial s}$ and $\frac{\partial v}{\partial s}$, respectively, can be computed by the following algorithm:

Algorithm J

$d^*_{-1} = 0$

$d^*_0 = b^*_0$

$d^*_1 = b^*_1 + sd^*_0 + \eta_1$

$d^*_2 = b^*_2 + sd^*_1 + td^*_0 - \alpha(d^*_0 - b^*_0) + \eta_2$

For $i = 3,4,\ldots,n-2$

$$d_i^* = b_i^* + sd_{i-1}^* + td_{i-2}^* + \alpha\left(b_{i-2}^* - d_{i-2}^* + sd_{i-3}^*\right) - \alpha^2 d_{i-4}^* + \eta_i$$

$$d_{n-1}^* = b_{n-1}^* + sd_{n-2}^* + td_{n-3}^* + \alpha b_{n-3}^* + \alpha sd_{n-4}^* - \alpha^2 d_{n-5}^*$$

$$\frac{\partial u^*}{\partial s} = d_{n-2}^*$$

$$\frac{\partial v^*}{\partial s} = d_{n-1}^*$$

The $\eta_i$ are rounding errors. If each $\eta_i = 0$ and if $b_i^* = b_i$ then $\frac{\partial u^*}{\partial s} = \frac{\partial u}{\partial s}$ and $\frac{\partial v^*}{\partial s} = \frac{\partial v}{\partial s}$.

Algorithm K can be used to compute approximations to $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$ as follows:

Algorithm K

$$C_{-2}^* = 0$$

$$C_{-1}^* = 0$$

$$C_0^* = b_0^*$$

$$C_1^* = b_1^* + sC_0^* + \zeta_1$$

For $i = 2,3,\ldots,n-3$

$$C_i^* = b_i^* + sC_{i-1}^* + tC_{i-2}^* - \alpha\left(C_{i-2}^* - sC_{i-3}^*\right) - \alpha^2 C_{i-4}^* + \zeta_i$$

$$C_{n-2}^* = b_{n-2}^* + sC_{n-3}^* + tC_{n-4}^* + \alpha sC_{n-5}^* - \alpha^2 C_{n-6}^*$$

$$\frac{\partial u^*}{\partial t} = C_{n-3}^*$$

$$\frac{\partial v^*}{\partial t} = C_{n-2}^*$$

The $\zeta_i$ are the rounding errors and $\frac{\partial u^*}{\partial t}$ and $\frac{\partial v^*}{\partial t}$ are approximations to $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$.

Then the $\Delta s$ and $\Delta t$ satisfy

$$\begin{bmatrix} \frac{\partial u^*}{\partial s} & \frac{\partial u^*}{\partial t} \\ \frac{\partial v^*}{\partial s} & \frac{\partial v^*}{\partial t} \end{bmatrix}_{s,t} \begin{bmatrix} \Delta s - \rho_1 \\ \Delta t - \rho_2 \end{bmatrix} = - \begin{bmatrix} u^* \\ v^* \end{bmatrix}_{s,t}$$

where $\rho_1$ and $\rho_2$ take into account the rounding errors made in determining $\Delta s$ and $\Delta t$. Thus, the new estimates for $s$ and $t$ are $s_N = s + \Delta s$ and $t_N = t + \Delta t$.

Successive estimates for $s$ and $t$ can be calculated in this manner, but the process does not cause convergence unless the rounding errors are controlled. The following method is chosen for controlling rounding errors: As the $(i + 1)$th estimate $\begin{bmatrix} s_{i+1} \\ t_{i+1} \end{bmatrix}$ is being calculated, each of the rounding errors $\left[\psi_r \ (r = 1,2,\ldots,n),\right.$ $\eta_r \ (r = 1,2,\ldots,n-1)$, $\zeta_r \ (r = 1,2,\ldots,n-2)$, and $\left.\rho_1,\rho_2\right]$ is required to be in absolute value less than $\lambda \left\| \begin{matrix} s_i - s_{i-1} \\ t_i - t_{i-1} \end{matrix} \right\|_\infty^2$, where $\lambda = \dfrac{1}{\left\| \begin{bmatrix} s_0 \\ t_0 \end{bmatrix} \right\|_\infty}$ and $\begin{bmatrix} s_0 \\ t_0 \end{bmatrix}$ is the initial estimate.

Thus the algorithm for obtaining the interval midpoints is:

Algorithm L

$$\lambda = \frac{1}{\max\{|s_0|, |t_0|\}} (1 \ominus 0.01)$$

For $j = 0,1,\ldots,$large

$s = s_j$

$t = t_j$

$\xi = \xi_j$

$b_0^* = a_0$

$b_1^* = \left(a_1 + s b_0^*\right) \oplus \xi$

$b_2^* = \left(a_2 + s b_1^* + t b_0^* - \alpha b_0^*\right) \oplus \xi$

$b_3^* = \left[a_3 + s b_2^* + t b_1^* - \alpha\left(b_1^* - s b_0^*\right)\right] \oplus \xi$

For $i = 4,5,\ldots,n-1$

$$b_i^* = \left[a_i + sb_{i-1}^* + tb_{i-2}^* - \alpha\left(b_{i-2}^* - sb_{i-3}^*\right) - \alpha^2 b_{i-4}^*\right] \oplus \xi$$

$$b_n^* = \left(a_n + sb_{n-1}^* + tb_{n-2}^* + \alpha sb_{n-3}^* - \alpha^2 b_{n-4}^*\right) \oplus \xi$$

If $\left|b_{n-1}^*\right| \leqq \frac{\delta a_{n-1}}{2R}$, $\left|b_n^* - sb_{n-1}^*\right| \leqq \frac{\delta a_n}{2R}$, and

$$\xi \leqq \min\left\{\frac{\delta a_1}{R}, \frac{\delta a_2}{R}, \ldots, \frac{\delta a_{n-2}}{R}, \frac{\delta a_{n-1}}{2R}, \frac{\delta a_n}{2R}\right\} \quad \text{then}$$

convergence has been obtained. Proceed to the determination of the interval half-widths.

$$d_{-1}^* = 0$$

$$d_0^* = b_0^*$$

$$d_1^* = \left(b_1^* + sd_0^*\right) \oplus \xi$$

$$d_2^* = \left[b_2^* + sd_1^* + td_0^* - \alpha\left(d_0^* - b_0^*\right)\right] \oplus \xi$$

For $i = 3,4,\ldots,n-2$

$$d_i^* = \left[b_i^* + sd_{i-1}^* + td_{i-2}^* + \alpha\left(b_{i-2}^* - d_{i-2}^* + sd_{i-3}^*\right) - \alpha^2 d_{i-4}^*\right] \oplus \xi$$

$$d_{n-1}^* = \left(b_{n-1}^* + sd_{n-2}^* + td_{n-3}^* + \alpha b_{n-3}^* + \alpha sd_{n-4}^* - \alpha^2 d_{n-5}^*\right) \oplus \xi$$

$$C_{-2}^* = 0$$

$$C_{-1}^* = 0$$

$$C_0^* = b_0^*$$

$$C_1^* = \left(b_1^* + sC_0^*\right) \oplus \xi$$

For $i = 2,3,\ldots,n-3$

$$C_i^* = \left[b_i^* + sC_{i-1}^* + tC_{i-2}^* - \alpha\left(C_{i-2}^* - sC_{i-3}^*\right) - \alpha^2 C_{i-4}^*\right] \circledast \xi$$

$$C_{n-2}^* = \left(b_{n-2}^* + sC_{n-3}^* + tC_{n-4}^* + \alpha sC_{n-5}^* - \alpha^2 C_{n-6}^*\right) \circledast \xi$$

$$\Delta s = \left(\frac{b_{n-1}^* C_{n-2}^* - b_n^* C_{n-3}^*}{C_{n-2}^* d_{n-2}^* - d_{n-1}^* C_{n-3}^*}\right) \circledast \xi$$

$$\Delta t = \left(\frac{b_n^* d_{n-2}^* - b_{n-1}^* d_{n-1}^*}{C_{n-2}^* d_{n-2}^* - d_{n-1}^* C_{n-3}^*}\right) \circledast \xi$$

$$\xi_i = \lambda\left(\min\left\{|\Delta s|, |\Delta t|\right\}\right)^2$$

$$s_{j+1} = s_j + \Delta s$$

$$t_{j+1} = t_j + \Delta t$$

Let

$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} = \begin{bmatrix} s \\ t \end{bmatrix}$$

$$\xi_i = \lambda \left\| x_i - x_{i-1} \right\|^2$$

$$f(x) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

and

$$f^*(x) = \begin{bmatrix} u^*(s,t) \\ v^*(s,t) \end{bmatrix}$$

Let

$$J = \begin{bmatrix} \frac{\partial u}{\partial s} & \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial s} & \frac{\partial v}{\partial t} \end{bmatrix}$$

and

$$J^* = \begin{bmatrix} \frac{\partial u^*}{\partial s} & \frac{\partial u^*}{\partial t} \\ \frac{\partial v^*}{\partial s} & \frac{\partial v^*}{\partial t} \end{bmatrix}$$

Then, theorem 4 may be stated as follows:

Theorem 4 – The convergence criteria of algorithm L are satisfied in a finite number of iterations if the following assumptions are satisfied:

(1) $\left\| J^{-1}(x_0) \right\| \leqq a$

(2) $\left\| x_1 - x_0 \right\| \leqq b$

(3) $\sum_{k=1}^{n} \left| \frac{\partial^2 f_i(x)}{\partial x^j \partial x^k} \right| \leqq \frac{c}{2}$

for all $x$ in $\left\| x - x_0 \right\| \leqq 2b$ where $x^1 = s$, $x^2 = t$, $f_1 = u$, and $f_2 = v$

(4) $h < \frac{1}{2}$ where $h = abc$

(5) $\lambda abK < \frac{1}{2} \frac{\frac{1}{2} - h}{5 + 8h}$

where

$$K = \sup_{\|x - x_0\| \leqq 2b} k(x)$$

and

$$k(x) = \frac{1}{a} + \delta f(x) + \frac{2aF}{1 - 2h} \delta J(x)$$

where $\delta f$ and $\delta J$ are positive continuous functions of $x$ with the property that

$$\left\| J^*(x_i) - J(x_i) \right\| \leqq \xi_i \, \delta J(x_i)$$

and, for each i,

$$\left\|f^*(x_i) - f(x_i)\right\| \leqq \xi\ \delta f(x_i)$$

$$(6) \qquad \lambda a b^2 \left[\sup_{\|x-x_0\| \leqq 2b} \delta J(x)\right] < \frac{1}{8}$$

$$(7) \qquad \xi_0 \leqq 4\lambda b^2$$

Proof – See appendix A.

## Obtaining Interval Half-Widths

From now on the $b_0^*, b_1^*, \ldots, b_{n-2}^*$ calculated by algorithm L are denoted by $b_0, b_1, \ldots, b_{n-2}$.

The interval midpoints s and t $(b_0, b_1, \ldots, b_{n-2})$ satisfy

$$a_0 P_n + a_1' P_{n-1} + \ldots + a_{n-2}' P_2 + P_1\left(a_{n-1}' - b_{n-1}\right) + P_0\left(a_n' - b_n + s b_{n-1}\right)$$

$$= \left(P_2 - s P_1 - t P_0\right)\left(b_0 P_{n-2} + \ldots + b_{n-3} P_1 + b_{n-2} P_0\right)$$

Let

$$a_i'' = a_i' \qquad (i = 1,2,\ldots,n-2)$$

$$a_{n-1}'' = a_{n-1}' - b_{n-1}$$

and

$$a_n'' = a_n' - b_n + s b_{n-1}$$

Let

$$P'' = a_0 P_n + \sum_{i=1}^{n} a_i'' P_i$$

Then

$$P'' \in \left[P \pm \frac{\delta P}{R}\right]$$

Consider

$$\overline{P}(x) = \left[P_2 - (s + \delta s)P_1 - (t + \delta t)\right]\left[b_{n-2} + \delta b_{n-2} + \left(b_{n-3} + \delta b_{n-3}\right)P_1 \right.$$

$$\left. + \ldots + \left(b_1 + \delta b_1\right)P_{n-3} + b_0 P_{n-2}\right]$$

Now, if

$$\overline{P}(x) = a_n'' + \delta a_n'' + \left(a_{n-1}'' + \delta a_{n-1}''\right)P_1 + \ldots + \left(a_1'' + \delta a_1''\right)P_{n-1} + a_0 P_n$$

then expanding the first equation for $\overline{P}(x)$ and equating coefficients of the $P_i(x)$ as was done in lemma 2 yield

$$a_0 = b_0$$

$$a_1'' + \delta a_1'' = b_1 + \delta b_1 - (s + \delta s)b_0$$

$$a_2'' + \delta a_2'' = b_0\alpha + b_2 + \delta b_2 - (s + \delta s)\left(b_1 + \delta b_1\right) - (t + \delta t)b_0$$

$$a_3'' + \delta a_3'' = \alpha\left[b_1 + \delta b_1 - (s + \delta s)b_0\right] + b_3 + \delta b_3$$

$$- (s + \delta s)\left(b_2 + \delta b_2\right) - (t + \delta t)\left(b_1 + \delta b_1\right)$$

and for $n - 2 \geq i \geq 4$

$$a_i'' + \delta a_i'' = \alpha^2\left(b_{i-4} + \delta b_{i-4}\right) + \alpha\left[b_{i-2} + \delta b_{i-2} - (s + \delta s)\left(b_{i-3} + \delta b_{i-3}\right)\right]$$

$$+ \left(b_i + \delta b_i\right) - (s + \delta s)\left(b_{i-1} + \delta b_{i-1}\right) - (t + \delta t)\left(b_{i-2} + \delta b_{i-2}\right)$$

$$a_{n-1}'' + \delta a_{n-1}'' = \left(b_{n-5} + \delta b_{n-5}\right)\alpha^2 + \alpha\left[b_{n-3} + \delta b_{n-3} - (s + \delta s)\left(b_{n-4} + \delta b_{n-4}\right)\right]$$

$$- (s + \delta s)\left(b_{n-2} + \delta b_{n-2}\right) - (t + \delta t)\left(b_{n-3} + \delta b_{n-3}\right)$$

$$a_n'' + \delta a_n'' = \alpha^2\left(b_{n-4} + \delta b_{n-4}\right) - \alpha(s + \delta s)\left(b_{n-3} + \delta b_{n-3}\right) - (t + \delta t)\left(b_{n-2} + \delta b_{n-2}\right)$$

Using the fact that

$$P''(x) = \left(P_2 - sP_1 - tP_0\right)\left(b_0 P_{n-2} + \ldots + b_{n-3}P_1 + b_{n-3}P_0\right)$$

yields

$$\left.\begin{aligned}
\delta a_1'' &= \delta b_1 - \delta s\, b_0 \\
\delta a_2'' &= \delta b_2 - s\,\delta b_1 - \delta s(b_1 + \delta b_1) - \delta t\, b_0 \\
\delta a_3'' &= \alpha(\delta b_1 - \delta s\, b_0) + \delta b_3 - s\,\delta b_2 - \delta s(b_2 + \delta b_2) \\
&\quad - t\,\delta b_1 - \delta t(b_1 + \delta b_1)
\end{aligned}\right\} \tag{37}$$

and for $n - 2 \geqq i \geqq 4$

$$\delta a_i'' = \alpha^2 \delta b_{i-4} + \alpha\left[\delta b_{i-2} - s\,\delta b_{i-3} - \delta s(b_{i-3} + \delta b_{i-3})\right] + \delta b_i - s\,\delta b_{i-1}$$
$$- \delta s(b_{i-1} + \delta b_{i-1}) - t\,\delta b_{i-2} - \delta t(b_{i-2} + \delta b_{i-2})$$

$$\delta a_{n-1}'' = \alpha^2\,\delta b_{n-5} + \alpha\left[\delta b_{n-3} - s\,\delta b_{n-4} - \delta s(b_{n-4} + \delta b_{n-4})\right]$$
$$- s\,\delta b_{n-2} - \delta s(b_{n-2} + \delta b_{n-2}) - t\,\delta b_{n-3} - \delta t(b_{n-3} + \delta b_{n-3})$$

$$\delta a_n'' = \alpha^2\,\delta b_{n-4} - \alpha\,\delta s(b_{n-3} + \delta b_{n-3}) - s\alpha\,\delta b_{n-3} - t\,\delta b_{n-2} - \delta t(b_{n-3} + \delta b_{n-2})$$

Now it is desirable to find bounds $\overline{\delta s}$ on $\delta s$, $\overline{\delta t}$ on $\delta t$, and $\overline{\delta b_i}$ on $\delta b_i$ such that if $|\delta s| \leqq \overline{\delta s}$, $|\delta t| \leqq \overline{\delta t}$, and $|\delta b_i| \leqq |\overline{\delta b_i}|$ $(i = 1,2,\ldots,n-2)$ then $\overline{P}(x) \in \left[P'' \pm \frac{R-1}{R}\,\delta P\right]$. This is true if

$$|\delta a_i''| \leqq \frac{R-1}{R}\,\delta a_i \qquad (i = 1,2,\ldots,n) \tag{38}$$

For $n - 2 \leqq i \leqq 4$, where $M > 1$, equations (38) are satisfied if

$$|\alpha^2\,\delta b_{i-4}| \leqq \frac{M-1}{5M}\,\frac{R-1}{R}\,\delta a_i$$

$$|s\,\delta b_{i-3}| \leqq \frac{M-1}{5M}\,\frac{R-1}{R}\,\delta a_i$$

$$(|\alpha| + |t|)|\delta b_{i-2}| \leqq \frac{M-1}{5M}\,\frac{R-1}{R}\,\delta a_i$$

$$\left|s\ \delta b_{i-1}\right| \leq \frac{M-1}{5M}\ \frac{R-1}{R}\ \delta a_i$$

$$\left|\delta b_i\right| \leq \frac{M-1}{5M}\ \frac{R-1}{R}\ \delta a_i$$

$$\left|\alpha\right|\left|\delta s\right|\left|b_{i-3} + \delta b_{i-3}\right| \leq \frac{1}{3M}\ \frac{R-1}{R}\ \delta a_i$$

$$\left|\delta s\right|\left|b_{i-1} + \delta b_{i-1}\right| \leq \frac{1}{3M}\ \frac{R-1}{R}\ \delta a_i$$

$$\left|\delta t\right|\left|b_{i-2} + \delta b_{i-2}\right| \leq \frac{1}{3M}\ \frac{R-1}{R}\ \delta a_i$$

Treating the equations for $\delta a_1''$, $\delta a_2''$, $\delta a_3''$, $\delta a_{n-1}''$, and $\delta a_n''$ in a similar manner shows that equations (38) are valid for $\left|\delta b_i\right| \leq \overline{\delta b}_i$, $\left|\delta s\right| \leq \overline{\delta s}$, and $\left|\delta t\right| \leq \overline{\delta t}$ with $\overline{\delta b}_1, \ldots, \overline{\delta b}_{n-2}$, $\overline{\delta s}$, and $\overline{\delta t}$ defined in the following manner:

$$\delta b_0 \equiv 0 \qquad (i \geq 1)$$

$$\overline{\delta b}_i \equiv \frac{M-1}{5M}\ \frac{R-1}{R}\ \min\left\{\delta a_i,\ \frac{\delta a_{i+1}}{|s|},\ \frac{\delta a_{i+2}}{|t| + |\alpha|},\ \frac{\delta a_{i+3}}{|s|},\ \frac{\delta a_{i+4}}{\alpha^2}\right\} \tag{39}$$

where $\delta a_j \equiv 0$ for $j > n$.

$$\overline{\delta t} \equiv \frac{1}{3M}\ \frac{R-1}{R}\ \min_{n \geq i \geq 2}\left\{\frac{\delta a_i}{\left|b_{i-2}\right| + \overline{\delta b}_{i-2}}\right\} \tag{40}$$

$$\overline{\delta s} \equiv \frac{1}{3M}\ \frac{R-1}{R}\ \min\left\{\min_{n \geq i \geq 1}\left\{\frac{\delta a_i}{\left|b_{i-1}\right| + \overline{\delta b}_{i-1}}\right\},\ \min_{n \geq i \geq 3}\left\{\frac{\delta a_i}{|\alpha|\left(\left|b_{i-3}\right| + \overline{\delta b}_{i-3}\right)}\right\}\right\} \tag{41}$$

Let

$$Q_1 = P_2 - sP_1 - tP_0$$

$$\delta Q_1 = \overline{\delta s}\ P_1 + \overline{\delta t}\ P_0$$

$$R_1 = b_0 P_{n-2} + b_1 P_{n-3} + \cdots + b_{n-2} P_0$$

and

$$\delta R_1 = \overline{\delta b}_1 P_{n-3} + \overline{\delta b}_2 P_{n-4} + \ldots + \overline{\delta b}_{n-2} P_0$$

Then

$$\left[Q_1 \pm \delta Q_1\right]\left[R_1 \pm \delta R_1\right] \subseteq \left[P \pm \delta P\right]$$

The same methods can be applied to $\left[R_1 \pm \delta R_1\right]$ to calculate $\left[Q_2 \pm \delta Q_2\right]$ and $\left[R_2 \pm \delta R_2\right]$ such that

$$\left[Q_2 \pm \delta Q_2\right]\left[R_2 \pm \delta R_2\right] \subseteq \left[R_1 \pm \delta R_1\right]$$

This process can be continued until

$$\left[Q_1 \pm \delta Q_1\right]\left[Q_2 \pm \delta Q_2\right] \ldots \left[Q_{\left[\frac{n}{2}\right]} \pm \delta Q_{\left[\frac{n}{2}\right]}\right]\left[R_{\left[\frac{n}{2}\right]} \pm \delta R_{\left[\frac{n}{2}\right]}\right] \subseteq \left[P \pm \delta P\right]$$

where $R_{\left[\frac{n}{2}\right]}$ is either a constant or a polynomial of degree one depending on whether $n$ is even or odd.

The roots of any $Q_i' \in \left[Q_i \pm \delta Q_i\right]$ are eigenvalues of some matrix $A' \in \left[A \pm \delta A\right]$.

# CHAPTER V

## CONCLUDING REMARKS

In chapter II algorithms are given for reducing an interval $[A \pm \delta A]$ of real matrices to an interval $[H \pm \delta H]$ of Hessenberg matrices such that each element of $[H \pm \delta H]$ is similar to some matrix belonging to the interval $[A \pm \delta A]$. The algorithms of chapter II can be used in developing contracting-interval programs for other processes which require reduction of a matrix to Hessenberg form.

In chapter III algorithms are given for reducing an interval of Hessenberg matrices to an interval of colleague matrices by a similarity transformation. As is shown in chapter III this interval of colleague matrices is equivalent to an interval of characteristic polynomials $[P \pm \delta P]$, each element of which is the characteristic polynomial of some $H' \in [H \pm \delta H]$.

In chapter IV algorithms are given for obtaining interval quadratic polynomials $\left[Q_1 \pm \delta Q_1\right]$, $\left[Q_2 \pm \delta Q_2\right]$, . . ., $\left[Q_{\left[\frac{n}{2}\right]} \pm \delta Q_{\left[\frac{n}{2}\right]}\right]$ such that

$$\left[Q_1 \pm \delta Q_1\right]\left[Q_2 \pm \delta Q_2\right] \cdots \left[Q_{\left[\frac{n}{2}\right]} \pm \delta Q_{\left[\frac{n}{2}\right]}\right]\left[R_{\left[\frac{n}{2}\right]} \pm \delta R_{\left[\frac{n}{2}\right]}\right] \subseteq [P \pm \delta P]$$

where $R_{\left[\frac{n}{2}\right]}$ is a constant if $n$ is even or $R$ is a polynomial of degree one if $n$ is odd.

Now if the Hessenberg matrix $H$ that is obtained from chapter II has any zeros along the subdiagonal then it is partitioned into Hessenberg matrices $H_1, H_2, \ldots, H_\ell$ with $n = \sum_{i=1}^{\ell} \text{order}\,(H_i)$ and such that no $H_i$ has a zero along the subdiagonal. Each $H_i$ must be reduced to colleague form by the methods of chapter III, and then the characteristic polynomial factored by the methods of chapter IV.

Combining the methods of chapters II to IV yields quadratic interval polynomials $\left[Q_{ij} \pm \delta Q_{ij}\right]$ $\left(j = 1,2,\ldots,\left[\frac{i}{2}\right];\ i = 1,2,\ldots,\ell\right)$ and interval polynomials $\left[R_i \pm \delta R_i\right]$ $(i = 1,2,\ldots,\ell)$, where $R_i$ is a polynomial of degree one if order $(H_i)$ is odd and $R_i$ is a constant if order $(H_i)$ is even. Then if

$$Q'_{ij} \in \left[Q_{ij} \pm \delta Q_{ij}\right] \qquad \left(j = 1,2,\ldots,\left[\frac{i}{2}\right];\ i = 1,2,\ldots,\ell\right)$$

and

$$R_i' \in \left[R_i \pm \delta R_i\right] \qquad (i = 1,2,\ldots,l)$$

then

$$\prod_{i=1}^{l}\left(R_i' \prod_{j=1}^{\left[\frac{i}{2}\right]} Q'_{ij}\right) \tag{42}$$

is the characteristic polynomial of some matrix $A' \in [A \pm \delta A]$.

Thus, any root of equation (42) is an eigenvalue of some matrix $A' \in [A \pm \delta A]$. Therefore, since the roots of any $Q_{ij}$ are trivial to determine, a method is developed for obtaining eigenvalues which are exact for a matrix differing by less than a specified amount from the matrix $A$.

Appendix B contains an example of this method applied to a simple $3 \times 3$ matrix.

It is essential that variable-precision arithmetic be used when programing the algorithms given in this report. It cannot be assured that the program is a contracting-interval program unless the required amount of precision is used, and this required amount of precision may be more than is obtained with either single- or double-precision arithmetic.

Variable-precision arithmetic is presently not available at very many computing installations since the hardware on most present-generation computers was not designed in a way which would facilitate implementation of variable-precision arithmetic. However, software implementations of variable-precision arithmetic such as SPAR (ref. 8) have been developed. As more efficient implementations of variable-precision arithmetic are developed, the advantages of contracting-interval programs will be more discernible. Even with efficient implementations of variable-precision arithmetic it will probably take more computer time to execute an algorithm by using variable-precision arithmetic than by using single-precision arithmetic. However, this should not lead to the dismissal of the concept of contracting-interval programs. The accuracy of computed results must be determined by some means if these results are to be of any use, and it may be more efficient to use a few more minutes of computer time to calculate results of known accuracy than to estimate the accuracy by some other means. Conventional methods of determining accuracy may require a comparison of the results of several computer runs or a study of the problem by a numerical analyst.

It has been the accepted pattern to have computers assume more and more duties previously done by people, so the concept of computing numbers of known accuracy should be the natural thing to do when the computers are designed with this in mind and when the methods become available.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., July 29, 1971.

## APPENDIX A

## PROOF OF THEOREM 4

The proof of theorem 4 makes use of a theorem which gives sufficient conditions for the convergence of Newton's method in $R^n$, where the function and its Jacobian are known only approximately. This theorem is a modification of a theorem given on page 115 in reference 12.

Suppose it is desired to find a root of $f(x) = 0$, where

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{bmatrix}$$

By using Newton's method successive approximations would be generated by solving $J(x_i)\,(x_{i+1} - x_i) = -f(x_i)$ for $x_{i+1}$, where $J(x_i)$ is the Jacobian evaluated at $x_i$. In most cases neither $J$ nor $f$ can be evaluated exactly at the point $x_i$. Let $J_{\xi_i}(x_i)$ and $f_{\xi_i}(x_i)$ be the approximations to $J(x_i)$ and $f(x_i)$ that are calculated.

Therefore, actually,

$$J_{\xi_i}(x_i)\,(x_{i+1} - x_i) = -f_{\xi_i}(x_i)$$

is solved for $x_{i+1}$ except that this system of equations cannot be solved exactly for $x_{i+1}$. Therefore, $x_{i+1}$ satisfies

$$J_{\xi_i}(x_i)\,(x_{i+1} - x_i - \rho_i) = -f_{\xi_i}(x_i) \tag{A1}$$

where $\rho_i$ is the error made in the calculation.

Now, if the approximations $J_{\xi_i}(x_i)$ and $f_{\xi_i}(x_i)$ do not approach $J(x_i)$ and $f(x_i)$ or if $\rho_i$ does not approach 0 as $i$ approaches $\infty$ then the sequence of iterates cannot be expected to converge to a root of $f(x) = 0$.

However, suppose it can be guaranteed that

$$\left\| J_{\xi_i}(x_i) - J(x_i) \right\|_\infty \leqq \xi_i \, \delta J(x_i) \tag{A2}$$

$$\left\| f_{\xi_i}(x_i) - f(x_i) \right\|_\infty \leqq \xi_i \, \delta f(x_i) \tag{A3}$$

and

$$\left\| \rho_i \right\|_\infty \leqq \xi_i \tag{A4}$$

where $\delta J(x)$ and $\delta f(x)$ are positive continuous functions of $x$,

$$\xi_i = \lambda \left\| x_i - x_{i-1} \right\|_\infty^2 \qquad (i = 1,2,\ldots)$$

$\xi_0$ is a prescribed value, and $\lambda$ is a constant having the same dimensions as $1/x$.

The theorem gives sufficient conditions for the convergence of the iterates determined by equation (A1) where equations (A2) to (A4) are satisfied at each step of the procedure.

Theorem 5 – Let successive approximations to a root of $f(x) = 0$ be generated by solving equation (A1) for $x_{i+1}$ at each step, where equations (A2) to (A4) are satisfied and $\xi_0 \leqq 4\lambda b^2$.

All norms used are $\infty$-norms. The following assumptions are made:

(1) If $x_0$ is the initial iterate, then $\left\| J^{-1}(x_0) \right\| \leqq a$

(2) $\left\| x_1 - x_0 \right\| \leqq b$

(3) Let the components of $f(x)$ have continuous second derivatives which satisfy

$$\sum_{k=1}^{n} \left| \frac{\partial^2 f_i(x)}{\partial x^j \partial x^k} \right| \leqq \frac{c}{h} \quad \text{for all} \quad x \quad \text{in} \quad \left\| x - x_0 \right\| \leqq 2b$$

(4) Let $h < \frac{1}{2}$ where $h = abc$

(5) Let $f_\xi(x)$ be bounded for $\left\| x - x_0 \right\| \leqq 2b$ and $\xi \leqq \xi_0$, and let

$$F \equiv \sup_{\substack{\|x-x_0\| \leqq 2b \\ \xi \leqq \xi_0}} \left\| f_\xi(x) \right\|$$

Let

$$k(x) \equiv \frac{1}{a} + \delta f(x) + \frac{2aF}{1 - 2h}\,\delta J(x)$$

and

$$K \equiv \sup_{\|x-x_0\| \leqq 2b} k(x)$$

Then it is required that

$$\lambda abK < \frac{1}{2}\,\frac{\frac{1}{2} - h}{5 + 8h}$$

$$(6) \qquad \lambda ab^2 \left[ \sup_{\|x-x_0\| \leqq 2b} \delta J(x) \right] < \frac{1}{8}$$

If all these assumptions are satisfied then the iterates are uniquely defined and $\|x - x_0\| \leqq 2b$ for each $v$; and the iterates converge to some vector, say $\lim_{v \to \infty} x_v = \alpha$, for which

$$f(\alpha) = 0$$

and

$$\|x_v - \alpha\| \leqq \frac{2b}{2^v}$$

Proof – See reference 7, page 11.

Lemma 3 – Let $w_i = y_i + \sum_{j=1}^{i-1} \alpha_{ij}(s,t)\, w_j$

$$w_i' = y_i' + \sum_{j=1}^{i-1} \alpha_{ij}(s,t)\, w_j'$$

and $z_i = w_i - w_i'$ $(i = 1,2,\ldots,n)$ where each $\alpha_{ij}(s,t)$ is a continuous function of $s$ and $t$. If $C_i(s,t)$ is a positive continuous function of $s$ and $t$ for each $i$ and if $|y_i - y_i'| < \xi\, C_i(s,t)$ $(i = 1,2,\ldots,n)$ then $|z_i| < \xi\, B_i(s,t)$ for some positive continuous function $B_i(s,t)$.

Proof – For $i = 1$ define $B_1 \equiv C_1$ and the result is obvious.

Assume $|z_i| < \xi B_i$ $(i = 2,3,\ldots,k)$. Then

$$|z_{k+1}| = \left| y_{k+1} - y'_{k+1} + \sum_{j=1}^{k} \alpha_{k+1,j} z_j \right| \leqq |y_{k+1} - y'_{k+1}| + \sum_{j=1}^{k} |\alpha_{k+1,j}| |z_j| < \xi C_{k+1} + \sum_{j=1}^{k} |\alpha_{k+1,j}| \xi B_j$$

Define $B_{k+1}(s,t) \equiv C_{k+1}(s,t) + \sum_{j=1}^{k} |\alpha_{k+1,j}(s,t)| B_j(s,t)$. Then $|z_{k+1}| < \xi B_{k+1}$ and $B_{k+1}$ is a continuous function of $s$ and $t$. Let $x = \begin{bmatrix} s \\ t \end{bmatrix}$. Then

$$f(x) = \begin{bmatrix} u(s,t) \\ v(s,t) \end{bmatrix}$$

and

$$f^*(x) = \begin{bmatrix} u^*(s,t) \\ v^*(s,t) \end{bmatrix}$$

Let

$$J = \begin{bmatrix} \frac{\partial u}{\partial s} & \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial s} & \frac{\partial v}{\partial t} \end{bmatrix}$$

and

$$J^* = \begin{bmatrix} \frac{\partial u^*}{\partial s} & \frac{\partial u^*}{\partial t} \\ \frac{\partial v^*}{\partial s} & \frac{\partial v^*}{\partial t} \end{bmatrix}$$

Theorem 6 – Let $f^*(x)$ be calculated by algorithm I where $|\psi_i| < \xi$ $(i = 1,2,\ldots,n)$. Then there exists a positive continuous function $\delta f(x)$ such that $\|f^*(x) - f(x)\| < \xi\, \delta f(x)$

for all $x$. Let $J^*(x)$ be calculated by algorithms J and K where $|\eta_i| < \xi$ $(i = 1,2,\ldots,n-1)$ and $|z_i| < \xi$ $(i = 1,2,\ldots,n-2)$. Then there exists a positive continuous function $\delta J(x)$ such that $\|J^*(x) - J(x)\| < \xi\, \delta J(x)$ for all $x$.

Proof – Since the multipliers of the $b_i$ and $b_i^*$ in algorithms H and I are continuous functions of $s$ and $t$ and, since

$$\left|a_i - (a_i + \psi_i)\right| = \left|\psi_i\right| < \xi \qquad (i = 1,2,\ldots,n)$$

lemma 3 can be applied to obtain $\left|b_i - b_i^*\right| < \xi\, B_i(s,t)$ for some positive continuous function $B_i(s,t)$.

Let $\delta f(x) = \left\| \begin{matrix} B_{n-1}(s,t) \\ B_n(s,t) \end{matrix} \right\|$ then $\delta f(x)$ is a continuous function of $x$ and

$$\left\|f^*(x) - f(x)\right\| = \left\| \begin{bmatrix} u^*(s,t) - u(s,t) \\ v^*(s,t) - v(s,t) \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} b_{n-1}^*(s,t) - b_{n-1}(s,t) \\ b_n^*(s,t) - b_n(s,t) \end{bmatrix} \right\| < \xi\, \delta f(x)$$

Now

$$\left|b_i - (b_i^* + \eta_i)\right| \leqq \left|b_i - b_i^*\right| + \left|\eta_i\right| \leqq \xi\left[B_i(s,t) + 1\right]$$

Let $d_i$ be the result of algorithm J when each $\eta_i = 0$. Then by lemma 3 there exist positive continuous functions $D_i(s,t)$ such that

$$\left|d_i - d_i^*\right| < \xi\, D_i(s,t)$$

Therefore

$$\left|\frac{\partial u}{\partial s} - \frac{\partial u^*}{\partial s}\right| < \xi\, D_{n-2}(s,t)$$

and

$$\left|\frac{\partial v}{\partial s} - \frac{\partial v^*}{\partial s}\right| < \xi\, D_{n-1}(s,t)$$

Likewise, if $c_i$ is the result of algorithm K when each $\zeta_i = 0$

$$\left|c_i - c_i^*\right| < \xi\, C_i(s,t) \qquad (i = 1,2,\ldots,n-2)$$

for some positive continuous functions $C_i(s,t)$ $(i = 1,2,\ldots,n-2)$. Therefore

$$\left|\frac{\partial u}{\partial t} - \frac{\partial u^*}{\partial t}\right| < \xi\, C_{n-3}(s,t)$$

and

$$\left|\frac{\partial v}{\partial t} - \frac{\partial v^*}{\partial t}\right| < \xi\, C_{n-2}(s,t)$$

Define

$$\delta J(x) \equiv \left\| \begin{bmatrix} D_{n-2}(s,t) & C_{n-3}(s,t) \\ D_{n-1}(s,t) & C_{n-2}(s,t) \end{bmatrix} \right\|$$

Then, $\delta J(x)$ is a positive continuous function of $x$ and

$$\left\|J^*(x) - J(x)\right\| < \xi\, \delta J(x)$$

Lemma 4 – In any bounded region $\|x - x_0\| < r$, if $\xi < \xi_0$ where $\xi_0$ is some positive value, then there exists a constant $F$ such that $\left\|f^*(x)\right\| < F$ for each $x$ in this region.

Proof – $\left\|f^*(x)\right\| < \left\|f(x)\right\| + \xi_0\, \delta f(x)$. The result follows from the fact that both $f(x)$ and $\delta f(x)$ are continuous functions.

Theorem 7 – The convergence criteria of algorithm L are satisfied in a finite number of iterations if the following assumptions are satisfied:

(1) $\left\|J^{-1}(x_0)\right\| \leqq a$

(2) $\left\|x_1 - x_0\right\| \leqq b$

(3) $\displaystyle\sum_{k=1}^{2} \left| \frac{\partial^2 f_i(x)}{\partial x^j \partial x^k} \right| \leqq \frac{c}{2}$

for all $x$ in $\left\|x - x_0\right\| \leqq 2b$ where $x^1 = s$, $x^2 = t$, $f_1 = u$, and $f_2 = v$.

(4) $h < \frac{1}{2}$ where $h = abc$

(5) $\lambda abK < \frac{1}{2} \frac{\frac{1}{2} - h}{5 + 8h}$

where

$$K = \sup_{\|x-x_0\| \leqq 2b} k(x)$$

and

$$k(x) = \frac{1}{a} + \delta f(x) + \frac{2aF}{1 - 2h} \delta J(x)$$

(6) $\lambda ab^2 \left( \sup_{\|x-x_0\| \leqq 2b} \delta J(x) \right) < \frac{1}{8}$

(7) $\xi_0 \leqq 4\lambda b^2$

Proof – In iteration $k$ of algorithm L the rounding errors made in calculating the $b_i^*$, $C_i^*$, and $d_i^*$ are all less than $\xi_k$. Thus, theorem 6 can be applied to get

$$\left\| f^*(x_k) - f(x_k) \right\| < \xi_k \, \delta f(x_k)$$

and

$$\left\| J^*(x_k) - J(x_k) \right\| < \xi_k \, \delta J(x_k)$$

The rounding errors made in calculating $\Delta s$ and $\Delta t$ are less than $\xi_k$. Thus, all the requirements of theorem 5 are met. Therefore, the successive iterates $x_j$ converge to some $\alpha$ for which $f(\alpha) = 0$. Also, since $x_j \to \alpha$, $\xi_j \to 0$.

Let $\chi = \min\left\{ \frac{\delta a_1}{R}, \frac{\delta a_2}{R}, \ldots, \frac{\delta a_{n-2}}{R}, \frac{\delta a_{n-1}}{2R}, \frac{\delta a_n}{2R} \right\}$. There exists $N_1$ such that $j > N_1$ implies that $\xi_j < \chi$.

Now $0 = f(\alpha) = f\left( \lim_{j \to \infty} x_j \right) = \lim_{j \to \infty} f(x_j)$. Therefore, there exists $N_2$ such that for $j > N_2$

$$\left|v(s_j,t_j)\right| < \frac{\delta a_n}{4R}$$

and

$$\left|u(s_j,t_j)\right| < \min\left\{\frac{\delta a_{n-1}}{2R}, \frac{\delta a_n}{|s|\,4R}\right\}$$

These two equations imply that for $j > N_2$

$$\left|u(s_j,t_j)\right| < \frac{\delta a_{n-1}}{2R}$$

and

$$\left|v(s_j,t_j) - s\,u(s_j,t_j)\right| < \frac{\delta a_n}{2R}$$

Therefore, for $j > \max\{N_1,N_2\}$ the convergence criteria of algorithm L are met.

## APPENDIX B

## EXAMPLE

In appendix B, the algorithms of this report are applied to a $3 \times 3$ matrix. It is hoped that this will aid the reader in understanding which algorithms to use, the sequence in which they are applied, and the result of each algorithm. Let

$$A = \begin{bmatrix} 5 & -6 & -6 \\ -1 & 4 & 2 \\ 3 & -6 & -4 \end{bmatrix} \qquad \delta A = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}$$

$R = 10$ and each $\alpha_i, \beta_i = 0$. Use of algorithm D yields

$$A = \begin{bmatrix} 5 & -4.02 & -6 \\ 0.3 & -2.02 & -6 \\ 0.33 & 0.01 & 2.02 \end{bmatrix}$$

and use of algorithm E yields

$$\delta A = \begin{bmatrix} 0.09 & 0.045 & 0.09 \\ 0.09 & 0.045 & 0.09 \\ & 0.0225 & 0.045 \end{bmatrix}$$

where $n_{32} = a_{31}$. Now let the upper Hessenberg portion of $A$ be renamed

$$H = \begin{bmatrix} 5 & -4.02 & -6 \\ 3 & -2.02 & -6 \\ & 0.01 & 2.02 \end{bmatrix}$$

and let $\delta H = \delta A$. Then, algorithm G yields

$$V = \begin{bmatrix} 1 & -1.667 & 65.266 \\ & 0.3333 & -99.3734 \\ & & 33.33 \end{bmatrix}$$

The fourth column of V is

$$\begin{bmatrix} -135.8393 \\ 268.0000 \\ -166.7000 \\ 33.33 \end{bmatrix}$$

Equation (35) yields

$$\begin{bmatrix} \delta V_1 = 0.027 \\ \delta V_2 = 0.0054 \\ \delta V_3 = 0.0015 \end{bmatrix}$$

Thus, the interval polynomial is

$$33.33X^2 + [-166.7 \oplus 0.0015]X^2 + [268.0 \oplus 0.0054]X + [-135.8393 \oplus 0.027]$$

Now algorithm L is used to factor this polynomial. If $s_0 = 3$, $t_0 = -2$, and $\xi_0 = 10^{-4}$, then after the first iteration $s_1 = 3.0022$, $t_1 = -2.0341$, and $\xi_1 = 10^{-5}$. After two iterations $s_2 = 3.00211$, $t_2 = -2.03841$, and $\xi_2 = 10^{-8}$. During the third iteration there are computed

$$b_0^* = 33.33$$

$$b_1^* = -66.63967370$$

$$b_2^* = 0.00016389$$

$$b_3^* = 0.00016928$$

The convergence criteria are met.

Now, by using equations (39) to (41) and with $M = 3$, $\overline{\delta b}_1$, $\overline{\delta t}$, and $\overline{\delta s}$ are computed as

$$\overline{\delta b}_1 = 0.00018$$

$$\overline{\delta t} = 0.00004051$$

and

$$\overline{\delta s} = 0.0000081$$

Thus

$$\left(X^2 - [3.00211 \oplus 0.0000081]X - [-2.03841 \oplus 0.00004051]\right)\left(33.33X + [-66.6396737 \oplus 0.00018]\right)$$

$$\subset 33.33X^3 + [-166.7 \oplus 0.0015]X^2 + [268.0 \oplus 0.0054]X + [-135.8393 \oplus 0.027]$$

Note that, since

$$H = \begin{bmatrix} 5 & -4.02 & -6 \\ 3 & -2.02 & -6 \\ & 0.01 & 2.02 \end{bmatrix}$$

and

$$\delta H = \begin{bmatrix} 0.09 & 0.045 & 0.09 \\ 0.09 & 0.045 & 0.09 \\ & 0.0225 & 0.045 \end{bmatrix}$$

if the 0.01 is replaced with 0 in $H$ and the 0.0225 replaced with 0.0125 in $\delta H$, since $[0 \oplus 0.0125] \subset [0.01 \oplus 0.0225]$, then

$$H = \begin{bmatrix} 5 & -4.02 & -6 \\ 3 & -2.02 & -6 \\ & & 2.02 \end{bmatrix}$$

and

$$\delta H = \begin{bmatrix} 0.09 & 0.045 & 0.09 \\ 0.09 & 0.045 & 0.09 \\ & 0.0125 & 0.045 \end{bmatrix}$$

Then $H$ can be partitioned into submatrices which can be treated separately. Let

$$H_1 = \begin{bmatrix} 5 & -4.02 \\ 3 & -2.02 \end{bmatrix} \qquad \delta H_1 = \begin{bmatrix} 0.09 & 0.045 \\ 0.09 & 0.045 \end{bmatrix}$$

$H_2 = 2.02$ and $\delta H_2 = 0.045$. Thus, for $\lambda \in [2.02 \pm 0.045]$, $\lambda$ is an eigenvalue of some matrix in $[A \pm \delta A]$.

Using algorithm G on $H_1$ yields

$$V = \begin{bmatrix} 1 & 1.667 \\ & 0.3333 \end{bmatrix}$$

The third column of $V$ is

$$\begin{bmatrix} 0.652 \\ -0.9937 \\ 0.3333 \end{bmatrix}$$

$$\overline{\delta V_1} = 0.02025$$

and

$$\overline{\delta V_2} = 0.0081$$

Therefore, the characteristic polynomial of $[H_1 \pm \delta H_1]$ is

$$0.3333X^2 + [-0.9937 \pm 0.0081]X + [0.652 \pm 0.02025]$$

Thus

$$\left(X - [2.02 \pm 0.045]\right)\left(0.3333X^2 + [-0.9937 \pm 0.0081]X + [0.652 \pm 0.02025]\right)$$

is an interval polynomial, each element of which is the characteristic polynomial of some $A' \in [A \pm \delta A]$.

## REFERENCES


1. Chartres, Bruce A.: Automatic Controlled Precision Calculations. J. Assoc. Comput. Mach., vol. 13, no. 3, July 1966, pp. 386-403.
2. Hansen, Eldon R.: On the Danilewski Method. J. Assoc. Comput. Mach., vol. 10, 1963, pp. 102-109.
3. Good, I. J.: The Colleague Matrix, A Chebyshev Analogue of the Companion Matrix. Quart. J. Math. (Oxford), vol. 2, no. 12, 1966, pp. 61-68.
4. Wilkinson, J. H.: The Algebraic Eigenvalue Problem. Clarendon Press (Oxford), 1965.
5. Herron, S. Russel: Zeros of Approximate Polynomials. Master of Computer Sci. Thesis, Univ. of Virginia, Jan. 1969.
6. Chartres, B. A.: Controlled Precision Calculations and the Danilewski Method. IBM-E-148/8, NSF-E-782, Div. Appl. Math., Brown Univ., Mar. 1964.
7. Harris, James Dean: A Contracting Interval Program for the Danilewski Method. Ph. D. Diss., Univ. of Virginia, June 1970.
8. Chartres, Bruce A.; and Wood, Thomas E.: SPAR Manual. Computer Science Center, Univ. of Virginia, [1970].
9. Wilkinson, J. H.: Rounding Errors in Algebraic Processes. Prentice-Hall, Inc., c.1963.
10. Chartres, Bruce A.; and Geuder, James C.: Computable Error Bounds for Direct Solution of Linear Equations. J. Assoc. Comput. Mach., vol. 14, no. 1, Jan. 1967, pp. 63-71.
11. Moore, Ramon E.: Interval Analysis. Prentice-Hall, Inc., c.1966.
12. Isaacson, Eugene; and Keller, Herbert Bishop: Analysis of Numerical Methods. John Wiley & Sons, Inc., c.1966.


NASA-Langley, 1971 —— 19 L-7725

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

# NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

**Washington, D.C. 20546**