Diss. Nr. 4624

# NUMERISCHE BESTIMMUNG DER EIGENWERTE EINER POSITIVEN QD-ZEILE UND DER RESIDUEN DER ZUGEORDNETEN ERZEUGENDEN FUNKTION MIT HILFE EINES STATIONÄREN ALGORITHMUS

ABHANDLUNG

zur Erlangung der Würde eines Doktors der Mathematik
der
EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE
ZÜRICH

vorgelegt von

TALAAT HILAL
B. Sc. El. Eng. Univ. Alexandria
dipl. Mathematiker ETH Zürich
geboren am 22. Dezember 1938
ägyptischer Staatsangehöriger

Angenommen auf Antrag von
Prof. Dr. E. Stiefel, Referent
Prof. Dr. P. Henrici, Korreferent

Juris Druck+Verlag Zürich
1971

$q_1' := e_1$ ;

auch 0 wird, wenn $e_1 = 0$ ist. In dem Fall wird ebenfalls sein Logarithmus $lq_1' = le_1$ aufgespeichert. Es ist hier zu merken, dass das zu 0 gewordene q-Element nach dem Deflations-Nullschritt wieder einen Wert erhält.

## § 2. GRUNDELEMENTE DES PROGRAMMS

integer nn, ne, n, k, km, m, p, i, j, jl, l, t;

nn = Ordnung der Anfangszeile
ne = Anzahl der verlangten Eigenwerte,
n = Ordnung der Zwischenzeilen, die immer durch jede Deflation um 1 kleiner wird,
k = Index der q-, e-Elemente,
km = Index des kleinsten $d_k$ (IV.6),
m = Index der Komponenten der Vektoren r, rl,
p = Index des letzten unveränderten $q_k$-Elements bei einem stationären Schritt,
i = Anzahl der ausgeführten progressiven Schritte bei jedem Eigenwert,
j = Anzahl der ausgeführten stationären Schritte bei jedem Eigenwert,
jl = Anzahl der ausgeführten logarithmischen stationären Schritte bei jedem Eigenwert,
l = Anzahl der ausgeführten Nullschritte bei jedem Eigenwert,
t = Anzahl der gesamten ausgeführten Schritte.

real v, v1, d, d1, sup, sup1, inf, f, f1, ff, s, a, b, gen, theta;

v = die Verschiebung, bzw. ihr Logarithmus bei den logarithmischen Schritten,

v1 = v beim vorigen Schritt beim stationären Algorithmus

d = die in den Algorithmen (IV.8), (V.7) auftretende Grösse $d_k$ bzw. $t_k$. Bei der logarithmischen Form des stationären Algorithmus ist $d = \ln(G_k^*)$ (V.8),

d1 = das dem letzten nicht geänderten q-Element $q_p$ entsprechende d beim stationären Algorithmus,

sup = (II.6),

sup1 = obere Grenze für den kleinsten Eigenwert nach jedem progressiven Schritt,

inf = (II.8),

f = ein Faktor kleiner als 1, der mit sup multipliziert ein Versuchswert für v gibt,

f1 = f beim vorigen Schritt,

ff = f des letzten Schrittes vor dem ersten Versagen beim vorigen Eigenwert, oder des ersten gelungenen Schrittes, falls der erste schon versagt hatte,

s = der Logarithmus des Zählers von (VI.2),

a,b = für verschiedene Zwecke benötigte Grössen,

gen = die gewünschte Rechengenauigkeit der Residuen, wobei für die best erreichbare Genauigkeit gen = 1.0 zu setzen ist,

theta = 0 (III.4).

boolean resid, missl, verklg, aendg;

resid true wenn die Residuen verlangt sind, sonst false,
missl true bei jedem Misslingen eines Versuchsschrittes
verklg true bei der ersten Verkleinerung von f bei jedem Eigenwert,
aendg true bei der ersten Aenderung eines q-Elementes innerhalb jedes stationären Schrittes.

array q, e, qq, ee, lq, le, lqq, lee, w, res [1:nn],
r, lr[-1:50*nn];

q = q-Elemente,
e = e-Elemente,
qq = die erzeugten q-Elemente aus einem Versuchsschritt,
ee = die erzeugten e-Elemente aus einem Versuchsschritt,
lq = ln(q),
le = ln(e),
lqq = ln(qq),
lee = ln(ee),
w = Summe der Verschiebungen, bzw. die Eigenwerte,
res = die Residuen,
r = die aufgespeicherten Grössen aller Verschiebungen der ausgeführten Schritte ($v_i$ in (VI.2)),
lr = ln(r).