KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING NUMERIEKE ANALYSE EN
TOEGEPASTE WISKUNDE
Celestijnenlaan 200A – 3001 Heverlee

On Computing Zeros of Analytic Functions and Related Problems in Structured Numerical Linear Algebra

Jury:
Prof. Dr. ir. E. Aernoudt, voorzitter
Prof. Dr. ir. A. Haegemans, promotor
Prof. Dr. ir. M. Van Barel, promotor
Prof. Dr. D. A. Bini
(Università degli Studi di Pisa)
Prof. Dr. ir. R. Cools
Prof. Dr. J. Quaegebeur
Prof. Dr. ir. S. Van Huffel
Prof. Dr. M. N. Vrahatis
(University of Patras)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Peter KRAVANJA

U.D.C. 519.6, 681.3*G13, 681.3*G15, 517.53

maart 1999

Computers are useless, they can only give you answers.

—Pablo Picasso

On Computing Zeros of Analytic Functions and Related Problems in Structured Numerical Linear Algebra

Peter Kravanja

ABSTRACT. This thesis is a blend of computational complex analysis and numerical linear algebra. We study the problem of computing all the zeros of an analytic function that lie inside a Jordan curve. The algorithm that we present computes not only approximations for the zeros but also their respective multiplicities. It does not require initial approximations for the zeros and we have found that it gives accurate results. A Fortran 90 implementation is available (the package ZEAL). Our approach is based on numerical integration and the theory of formal orthogonal polynomials. We show how it can be used to locate clusters of zeros of analytic functions. In this context we also present an alternative approach, based on rational interpolation at roots of unity. Next we consider the related problem of computing all the zeros and poles of a meromorphic function that lie inside a Jordan curve and that of computing all the zeros of an analytic mapping (in other words, all the roots of a system of analytic equations) that lie in a polydisk. We also consider analytic functions whose zeros are known to be simple, in particular Bessel functions (the package ZEBEC) and certain combinations of Bessel functions. Next we propose a modification of Newton's method for computing multiple zeros of analytic mappings. Under mild assumptions our iteration converges quadratically. It involves certain constants whose product is a lower bound for the multiplicity of the zero. As these constants are usually not known in advance, we devise an iteration in which not only an approximation for the zero is refined, but also approximations for these constants. In the last part of this thesis we develop stabilized fast and superfast algorithms for rational interpolation at roots of unity. These algorithms lead to fast and superfast solvers for (indefinite) linear systems of equations that have Hankel or Toeplitz structure.

D/1999/7515/01

ISBN 90-5682-171-7

Contents

Acknowledgements

Complex analysis has intrigued me from the moment that I first heard about it. I can still see myself watching BBC Open University programs about line integrals in the complex plane, residues and Cauchy's Theorem on early Sunday mornings ...

In 1989 I came to Leuven to study engineering and I remember how the calculus courses taught by Robert Piessens in his characteristic lively way stimulated me. In those days, however, an introduction to complex analysis was not part of the first two years of the engineering curriculum as it is today. Instead I had to wait for the introductory course taught by Ann Haegemans to a small group of fourth year students—those who had chosen to specialize in applied mathematics. Meanwhile I read several books on my own. However, by the time I arrived in my fourth year, the course given by Ann Haegemans had been abolished and replaced by a course taught by Robert Piessens to the second year students! During my fifth and final year of engineering, two important things happened. First, Adhemar Bultheel guided me towards my master's thesis, in which I presented a matrix generalization of his Szegő theory for rational functions. And secondly, I followed a course given by Ann Haegemans on applications of real and complex analysis. This turned out to be a very rewarding experience. I learned a lot in this course, especially about fractional calculus and Mellin and Hankel transforms. When by the end of that year the time came to choose a topic for a PhD thesis, I knew in which direction I wanted to do research. I chose the problem of computing zeros of analytic functions and I was lucky enough to find some clues in an early paper by Delves and Lyness and also in a book written by the Russian mathematicians Aĭzenberg and Yuzhakov, in which they formulated a multidimensional version of the logarithmic residue formula used by Delves and Lyness. The first part of this thesis has grown out of these two references. Finally, during my first PhD year I took the complex analysis course given by Alfons Van Daele at the Department of Mathematics. I remember it as a very interesting, clear and didactic course.

The previous account of "how I came to write a thesis on computational complex analysis" is of course only a partial and very teleological reconstruction. Besides, this thesis contains other ingredients as well, notably structured numerical linear algebra, numerical analysis and some informatics. But I have always had the inner conviction that I want to do research on computational complex analysis and one may read this thesis with that in mind.

I sincerely thank Ann Haegemans for accepting to be my promotor. She gave me the freedom to explore my research subject on my own while stimulating me via her kind and intelligent remarks. The results on Newton's method are joint work.

Marc Van Barel has a strong numerical intuition from which I have learned a lot. I thank him for guiding me through the world of formal orthogonal polynomials and for introducing me to structured matrices. The last part of this thesis was written in close collaboration with him.

I spent two months at the Department of Mathematics of the University of Patras (Greece). I sincerely thank Michael Vrahatis, Omiros Ragos and Filareti Zafiropoulos for their kind hospitality. They introduced me to problems concerning zeros of Bessel functions. The package ZEBEC was written in close collaboration

with them. Michael Vrahatis mentioned certain important combinations of Bessel functions to me. However, I didn't know how to tackle this particular problem. Back in Leuven, I mentioned it casually to Pierre Verlinden who, the next day, came back to my office telling me that he had an idea that might lead us somewhere. One thing indeed led to another and within less than two weeks we found all the proofs that we were looking for. These results are presented in the second part of this thesis.

I enjoy attending conferences. This thesis brought me to Luminy (France), Cortona (Italy), Hamburg (Germany), Nicosia (Cyprus), Kuwait City (Kuwait), Madrid (Spain) and Athens (Greece). There is something unusual about meeting more or less the same group of people about once a year, each time in a different city. I thank Georg Heinig, Dario Bini, Stefano Serra, Plamen Yalamov, Bernard Mourrain, Jean-Paul Cardinal and Ioannis Emiris for many cordial and stimulating conversations.

Many other people contributed to this thesis. Ronald Cools arranged my first visit to Patras and gave me advice on various compiler and Fortran 90 issues. I thank Johan Quaegebeur for joining the thesis committee and for the very enthusiastic way in which he showed interest in my work. Jan Verschelde introduced me to Dario Bini (during the SEA 95 workshop) and to Michael Vrahatis (during the ICCAM 96 conference). I thank Annie Cuyt for her friendly encouragements and for introducing me to Tetsuya Sakurai. Ria Vanhove is a very competent librarian. I sincerely thank her for the efficient way in which she obtained photocopies of papers for me from other libraries. I thank Philip Miller for reassuring me about my English. I shared offices with Frank Piessens, Marc Van Barel, Bart Maerten and Tom Michiels. The atmosphere was always very pleasant and creative.

This research was financed via a four year grant (1 October 1994 – 30 September 1998) from the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). Currently, I'm being paid as a research assistant at the Division of Numerical Analysis and Applied Mathematics. I thank the IWT and the K.U.Leuven for their financial support. Vlaamse Leergangen Leuven generously supported my research stays in Patras and the Fund for Scientific Research-Flanders (FWO-Vlaanderen) paid my ticket to Madrid. I thank Dario Bini for the support from the European Community that he was able to obtain for me to cover my expenses in Cortona. I thank Georg Heinig for giving me the opportunity to attend the FAA 98 conference in Kuwait. This was quite an experience. Most of my other travel and hotel expenses were financed by the FWO-Vlaanderen, projects #G.0261.96 "Counting and computing all isolated solutions of systems of nonlinear equations" and #G.0278.97 "Orthogonal Systems and their Applications."

And, last but not least, I thank my friends for the many nice moments and I thank my father for his love and support, for the possibilities, the freedom and the independence.

Peter Kravanja

—

Peter.Kravanja@na-net.ornl.gov

Nederlandse samenvatting

Dit proefschrift is een mengeling van numerieke complexe analyse en numerieke lineaire algebra. Het is onderverdeeld in vier delen.

We bestuderen eerst het probleem van het berekenen van *alle* nulpunten van een analytische functie f die binnen een positief georiënteerde Jordankromme γ liggen. Ons belangrijkste middel om informatie over de ligging der nulpunten te bekomen, bestaat uit een symmetrische, bilineaire vorm die geëvalueerd kan worden via numerieke integratie langs γ. Deze vorm bevat de logaritmische afgeleide f'/f van f. Onze aanpak kan bijgevolg een kwadratuurmethode genoemd worden die gebaseerd is op logaritmische residu's. In zekere zin zetten wij het pionierswerk van Delves en Lyness verder. We werpen een nieuw licht op hun aanpak door andere onbekenden te beschouwen en door gebruik te maken van de theorie der formele orthogonale veeltermen. Ons algoritme berekent niet alleen benaderingen voor de nulpunten maar ook hun respectievelijke multipliciteiten. Het vereist geen initiële benaderingen voor de nulpunten en we hebben vastgesteld dat het nauwkeurige resultaten geeft. Een Fortran 90 implementatie staat ter beschikking (het pakket ZEAL). We stellen ook een aanpak voor die de afgeleide f' niet vereist. Deze resultaten worden behandeld in Hoofdstuk 1.

In Hoofdstuk 2 spitsen we onze aandacht toe op het bepalen van clusters van nulpunten van analytische functies. We tonen aan hoe de aanpak die we in Hoofdstuk 1 voorgesteld hebben, gebruikt kan worden om een benadering te berekenen voor het middelpunt van een cluster en tevens het totale aantal nulpunten dat tot deze cluster behoort. We benaderen ons probleem van het berekenen van alle nulpunten van f die binnen γ liggen ook op een volledig andere manier, gebaseerd op rationale interpolatie in wortels uit één. We geven aan hoe deze nieuwe aanpak de vorige aanvult en op een efficiënte manier gebruikt kan worden indien γ de eenheidscirkel is.

In Hoofdstuk 3 tonen we aan hoe onze op logaritmische residu's gebaseerde aanpak gebruikt kan worden om alle nulpunten en polen van een meromorfe functie te berekenen die binnen een Jordankromme liggen.

In Hoofdstuk 4 beschouwen we stelsels analytische vergelijkingen. Een meerdimensionale logaritmische residuformule is beschikbaar in de literatuur. Deze formule bevat de integraal van een differentiaalvorm. We herschrijven deze integraal als een som van Riemann-integralen en tonen aan hoe de oplossingen en hun respectievelijke multipliciteiten uit deze integralen berekend kunnen worden door een veralgemeend eigenwaardenprobleem met Hankel-structuur op te lossen en een aantal Vandermonde-stelsels.

Dit vormt het einde van Deel 1. In Deel 2 beschouwen we analytische functies die alleen maar enkelvoudige nulpunten hebben, in het bijzonder Besselfuncties. In Hoofdstuk 5 spitsen we onze aandacht toe op de Besselfuncties $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ en $H_\nu^{(2)}(z)$, en hun eerste afgeleiden, indien de veranderlijke $z \in \mathbb{C} \setminus (-\infty, 0]$ en de orde $\nu \in \mathbb{R}$. We stellen ons softwarepakket ZEBEC voor, dat bedoeld is om alle nulpunten van deze functies te berekenen die binnen een rechthoek liggen waarvan de zijden evenwijdig zijn met de coördinaatassen. In Hoofdstuk 6 beschouwen we de functies $J_n(z) \pm iJ_{n+1}(z)$ en $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ indien $n \in \mathbb{N}$. De nulpunten van deze functies spelen een belangrijke rol in bepaalde natuurkundige toepassingen. We tonen aan dat alle nulpunten in $\mathbb{C}_0$ enkelvoudig zijn en geven aan hoe ZEBEC op een eenvoudige manier aangepast kan worden om deze nulpunten te berekenen.

In Deel 3 beschouwen we de methode van Newton. In Hoofdstuk 7 stellen we een aanpassing van de methode van Newton voor om meervoudige oplossingen van stelsels analytische vergelijkingen te berekenen. Onder milde voorwaarden convergeert onze iteratie kwadratisch. Ze bevat bepaalde constanten waarvan het product een benedengrens is voor de multipliciteit van de oplossing. Vermits deze constanten meestal niet op voorhand gekend zijn, leiden we een iteratie af waarin niet alleen een benadering voor de oplossing verfijnd wordt, maar ook benaderingen voor deze constanten.

In Deel 4 beschouwen we vraagstukken uit de gestructureerde numerieke lineaire algebra. We hebben rationale interpolatie al ontmoet op het einde van Hoofdstuk 2 waar we ervan gebruik gemaakt hebben om clusters van nulpunten van analytische functies te bepalen. In Hoofdstuk 8 beschouwen we rationale interpolatie in meer detail. We stellen gestabiliseerde snelle en supersnelle algoritmes voor rationale interpolatie in wortels uit één voor. In Hoofdstuk 9 gebruiken we deze algoritmes om gestabiliseerde snelle en supersnelle algoritmes te construeren voor het oplossen van (indefiniete) stelsels lineaire vergelijkingen met Hankel- of Toeplitz-structuur. Zo'n stelsels komen in heel wat toepassingen voor, bijv. in signaalverwerking of bij Markov-ketens. Ze spelen ook een centrale rol in de theorie der orthogonale veeltermen en bij Padé-benadering. Een Fortran 90 implementatie van onze algoritmes staat ter beschikking.

Dit proefschrift is een mengeling van theoretische resultaten (sommige daarvan zijn vrij technisch, bijv. de resultaten omtrent stelsels analytische vergelijkingen, rationale interpolatie of de methode van Newton), numerieke analyse en algoritmische aspecten, implementatieheuristieken en afgewerkte software (ZEAL, ZEBEC en onze software voor Hankel- en Toeplitz-stelsels).

Deel 1

1. Nulpunten van analytische functies

In dit hoofdstuk spitsen we onze aandacht toe op het volgende probleem. We wensen *alle* nulpunten van een analytische functie f te berekenen die binnen een Jordankromme γ liggen. Ons algoritme berekent niet alleen benaderingen voor de

nulpunten maar ook hun respectievelijke multipliciteiten. Het vereist geen initiële benaderingen voor de nulpunten en geeft nauwkeurige resultaten. Het is gebaseerd op de theorie der formele orthogonale veeltermen. Om informatie over de ligging der nulpunten te bekomen, maken we gebruik van een symmetrische bilineaire vorm die geëvalueerd kan worden via numerieke integratie langs γ. Deze vorm bevat de logaritmische afgeleide f'/f van f. Onze aanpak kan dus als volgt omschreven worden: het is een kwadratuurmethode gebaseerd op logaritmische residu's.

In de volgende hoofdstukken van het eerste deel van dit proefschrift zullen we aantonen hoe een gelijkaardige aanpak gebruikt kan worden om clusters van nulpunten van analytische functies te bepalen, om nulpunten en polen van meromorfe functies te berekenen, en om stelsels analytische vergelijkingen op te lossen.

Dit hoofdstuk stemt gedeeltelijk overeen met onze artikels [**186**], [**188**], [**190**] en [**192**].

Inleiding

Zij W een enkelvoudig samenhangend gebied in $\mathbb{C}$, $f : W \to \mathbb{C}$ analytisch in W en γ een positief georiënteerde Jordankromme in W die door geen enkel nulpunt van f gaat. We wensen *alle* nulpunten van f te berekenen die binnen γ liggen, samen met hun respectievelijke multipliciteiten.

Onze aanpak is een voortzetting van het pionierswerk geleverd door Delves en Lyness [**70**]. Zij N het totale aantal nulpunten van f die in het inwendige van γ liggen. Veronderstel vanaf nu dat $N > 0$. Delves en Lyness beschouwden de rij $Z_1, \ldots, Z_N$ bestaande uit alle nulpunten van f die binnen γ liggen. Elk nulpunt wordt hierbij zo vaak herhaald als zijn multipliciteit. Men kan eenvoudig nagaan dat de logaritmische afgeleide f'/f in elk nulpunt van f een enkelvoudige pool heeft met residu gelijk aan de multipliciteit van het nulpunt. De stelling van Cauchy impliceert dan dat

$$N = \frac{1}{2\pi i} \int_\gamma \frac{f'(z)}{f(z)}\, dz.$$

Deze formule laat ons toe N te berekenen via numerieke integratie. Methodes voor het berekenen van nulpunten van analytische functies die gebaseerd zijn op het numeriek evalueren van integralen worden *kwadratuurmethodes* genoemd. Een overzicht van dergelijke methodes werd gegeven door Ioakimidis [**158**]. Delves en Lyness beschouwden de integralen

$$s_p := \frac{1}{2\pi i} \int_\gamma z^p \frac{f'(z)}{f(z)}\, dz, \qquad p = 0, 1, 2, \ldots .$$

De residuenstelling impliceert dat de s_p's gelijk zijn aan de *Newtonsommen* van de onbekende nulpunten,

$$s_p = Z_1^p + \cdots + Z_N^p, \qquad p = 0, 1, 2, \ldots .$$

In wat volgt zullen we veronderstellen dat alle vereiste s_p's reeds berekend werden. In het bijzonder geldt dit voor $N = s_0$.

Delves en Lyness beschouwden de monische veelterm van graad N met nulpunten $Z_1, \ldots, Z_N$,

$$P_N(z) := \prod_{k=1}^{N} (z - Z_k) =: z^N + \sigma_1 z^{N-1} + \cdots + \sigma_N.$$

De coëfficiënten van $P_N(z)$ kunnen berekend worden via de identiteiten van Newton. Op deze manier herleidden ze het probleem tot het eenvoudigere probleem van het berekenen van de nulpunten van een veelterm. Helaas is de afbeelding van de Newtonsommen $s_1, \ldots, s_N$ naar de coëfficiënten $\sigma_1, \ldots, \sigma_N$ meestal slecht geconditioneerd. Daarom moeten de integralen zeer nauwkeurig berekend worden indien N groot is. Vandaar dat Delves en Lyness de veelterm $P_N(z)$ slechts opstelden indien de graad kleiner is dan een zeker getal M. Anders wordt het inwendige van γ onderverdeeld en worden de kleinere gebieden één na één beschouwd. Delves en Lyness kozen M gelijk aan 5.

Botten, Craig en McPhedran [**41**] implementeerden de methode van Delves en Lyness in Fortran 77.

Het probleem met deze aanpak is volgens ons dat men de verkeerde onbekenden beschouwt. Men dient de onderling verschillende nulpunten en hun respectievelijke multipliciteiten *afzonderlijk* te beschouwen. Dit is de aanpak die wij zullen volgen. Zij n het aantal onderling verschillende nulpunten van f die binnen γ liggen. Zij $z_1, \ldots, z_n$ deze nulpunten en $\nu_1, \ldots, \nu_n$ hun respectievelijke multipliciteiten. Onze kwadratuurmethode is een veralgemening van de methode van Delves en Lyness. We zullen aantonen hoe de onderling verschillende nulpunten berekend kunnen worden door veralgemeende eigenwaardenproblemen op te lossen. De waarde van n zullen we op een indirecte manier bepalen. Zodra n en $z_1, \ldots, z_n$ gekend zijn, kunnen de multipliciteiten $\nu_1, \ldots, \nu_n$ berekend worden door een Vandermonde-stelsel op te lossen.

Formele orthogonale veeltermen

Zij $\mathcal{P}$ de vectorruimte der veeltermen met complexe coëfficiënten. We definiëren de symmetrische bilineaire vorm

$$\langle \cdot, \cdot \rangle \; : \; \mathcal{P} \times \mathcal{P} \to \mathbb{C}$$

als volgt. Zij

$$\langle \phi, \psi \rangle := \frac{1}{2\pi i} \int_\gamma \phi(z)\psi(z) \frac{f'(z)}{f(z)}\, dz = \sum_{k=1}^{n} \nu_k \phi(z_k)\psi(z_k) \tag{1}$$

voor elke twee veeltermen $\phi, \psi \in \mathcal{P}$. De laatste gelijkheid volgt uit het feit dat f'/f een enkelvoudige pool heeft in z_k met residu ν_k voor $k = 1, \ldots, n$. Noteer dat $\langle \cdot, \cdot \rangle$ geëvalueerd kan worden via numerieke integratie langs γ. We veronderstellen vanaf nu dat alle vereiste "inwendige producten" $\langle \phi, \psi \rangle$ reeds berekend werden. Zij $s_p := \langle 1, z^p \rangle$ voor $p = 0, 1, 2, \ldots$. Deze gewone momenten zijn gelijk aan de *Newtonsommen* van de onbekende nulpunten,

$$s_p = \sum_{k=1}^{n} \nu_k z_k^p, \qquad p = 0, 1, 2, \ldots .$$

In het bijzonder geldt dat $s_0 = \nu_1 + \cdots + \nu_n = N$, het totale aantal nulpunten. We mogen dus veronderstellen dat de waarde van N gekend is. Zij H_k de $k \times k$ Hankel-matrix

$$H_k := \left[s_{p+q}\right]_{p,q=0}^{k-1} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{k-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{k-1} & \cdots & \cdots & s_{2k-2} \end{bmatrix}$$

voor $k = 1, 2, \ldots$. Een monische veelterm φ_t van graad $t \geq 0$ die voldoet aan

$$\langle z^k, \varphi_t(z) \rangle = 0, \qquad k = 0, 1, \ldots, t-1, \tag{2}$$

wordt een *formele orthogonale veelterm* (FOP) van graad t genoemd. Het adjectief *formeel* legt de nadruk op het feit dat in het algemeen de vorm $\langle \cdot, \cdot \rangle$ geen (positief definiet) inwendig product definieert. Een belangrijk gevolg hiervan is dat formele orthogonale veeltermen niet noodzakelijk voor elke graad t bestaan of enig zijn. (Voor meer details verwijzen we naar Draux [**71, 72**], Gutknecht [**128, 130**] of Gragg en Gutknecht [**113**].) Indien aan (2) voldaan is en φ_t enig is, dan wordt φ_t een *reguliere* FOP genoemd en t een *reguliere index*. Definieer

$$\varphi_t(z) =: u_{0,t} + u_{1,t} z + \cdots + u_{t-1,t} z^{t-1} + z^t.$$

Dan kan voorwaarde (2) herschreven worden als

$$\begin{bmatrix} s_0 & s_1 & \cdots & s_{t-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{t-1} & \cdots & \cdots & s_{2t-2} \end{bmatrix} \begin{bmatrix} u_{0,t} \\ u_{1,t} \\ \vdots \\ u_{t-1,t} \end{bmatrix} = - \begin{bmatrix} s_t \\ s_{t+1} \\ \vdots \\ s_{2t-1} \end{bmatrix}. \tag{3}$$

De reguliere FOP van graad $t \geq 1$ bestaat bijgevolg als en slechts als de matrix H_t regulier is.

De volgende stelling laat ons toe om, in theorie althans, n te berekenen als de rang van H_N.

STELLING. $n = \operatorname{rang} H_{n+p}$ voor elk niet-negatief geheel getal p. In het bijzonder is n gelijk aan de rang van H_N.

Bijgevolg is H_n regulier terwijl H_t singulier is voor $t > n$. Noteer dat $H_1 = [s_0]$ regulier is bij veronderstelling. De reguliere FOP van graad 1 bestaat en is gelijk aan $\varphi_1(z) = z - \mu$ waarbij

$$\mu := \frac{s_1}{s_0} = \frac{\sum_{k=1}^n \nu_k z_k}{\sum_{k=1}^n \nu_k}$$

het rekenkundige gemiddelde van de nulpunten is. De vorige stelling impliceert dat de reguliere FOP φ_n van graad n bestaat en dat er geen reguliere FOP's zijn van graad groter dan n. Men kan eenvoudig nagaan dat φ_n gelijk is aan

$$\varphi_n(z) = (z - z_1) \cdot \cdots \cdot (z - z_n).$$

Het is de monische veelterm van graad n die $z_1, \ldots, z_n$ als enkelvoudige nulpunten heeft.

Zodra de waarde van n gekend is, kunnen de onderling verschillende nulpunten $z_1, \ldots, z_n$ berekend worden door een veralgemeend eigenwaardenprobleem op te lossen. Zij $H_n^<$ de Hankel-matrix

$$H_n^< := \begin{bmatrix} s_1 & s_2 & \cdots & s_n \\ s_2 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_n & \cdots & \cdots & s_{2n-1} \end{bmatrix}.$$

STELLING. De eigenwaarden van de pencil $H_n^< - \lambda H_n$ zijn gelijk aan $z_1, \ldots, z_n$.

Zodra n en $z_1, \ldots, z_n$ gekend zijn, kunnen de multipliciteiten $\nu_1, \ldots, \nu_n$ berekend worden door het Vandermonde-stelsel

$$\begin{bmatrix} 1 & \cdots & 1 \\ z_1 & \cdots & z_n \\ \vdots & & \vdots \\ z_1^{n-1} & \cdots & z_n^{n-1} \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{bmatrix}. \tag{4}$$

op te lossen.

De vorige stellingen suggereren dus de volgende aanpak om n en $z_1, \ldots, z_n$ te bepalen. Eerst berekenen we het totale aantal nulpunten N. Vervolgens berekenen we $s_1, \ldots, s_{2N-2}$. We hebben reeds vermeld dat dit kan gebeuren met behulp van numerieke integratie langs γ. Het aantal onderling verschillende nulpunten berekenen we als de rang van H_N. De nulpunten $z_1, \ldots, z_n$ bekomen we tenslotte door een veralgemeend eigenwaardenprobleem op te lossen. Deze aanpak heeft echter verschillende nadelen:

- In theorie zijn de $N - n$ kleinste singuliere waarden van H_N gelijk aan nul. Bij numerieke berekeningen zal dit echter niet het geval zijn en is het dus moeilijk om de rang van H_N en dus de waarde van n te bepalen indien de kloof tussen de berekende benaderingen voor de singuliere waarden die in theorie gelijk zijn aan nul en de singuliere waarden die in theorie verschillend zijn van nul te klein is.
- De bekomen benaderingen voor $z_1, \ldots, z_n$ kunnen zeer onnauwkeurig zijn. De afbeelding van de Newtonsommen naar de nulpunten en hun respectievelijke multipliciteiten

$$(s_0, s_1, \ldots, s_{2n-1}) \mapsto (z_1, \ldots, z_n, \nu_1, \ldots, \nu_n) \tag{5}$$

  is immers meestal zeer slecht geconditioneerd. We verwijzen in dit verband ook naar de artikels van Gautschi [**93, 96, 97**] die de conditie van (5) onderzocht in verband met Gauss kwadratuurformules.

Vandaar dat we op zoek gaan naar een algoritme dat tot nauwkeurigere benaderingen van $z_1, \ldots, z_n$ leidt. De achterliggende idee is de volgende. De inwendige producten in de Hankel-matrices H_n en $H_n^<$ hebben te maken met de klassieke basis van de machten van z. Waarom beschouwen we geen andere basis? Het feit dat H_n en $H_n^<$

kunnen geschreven worden als

$$H_n = \left[\langle z^p, z^q\rangle\right]_{p,q=0}^{n-1} \quad \text{en} \quad H_n^< = \left[\langle z^p, z\,z^q\rangle\right]_{p,q=0}^{n-1}$$

suggereert dat we matrices van de vorm

$$\left[\langle \psi_p, \psi_q\rangle\right]_{p,q=0}^{n-1} \quad \text{en} \quad \left[\langle \psi_p, \psi_1\psi_q\rangle\right]_{p,q=0}^{n-1}$$

beschouwen, met ψ_k een veelterm van graad k voor $k = 0, 1, \ldots, n-1$. Het is zo dat, zelfs indien we erin slagen om de vorige stelling te formuleren in termen van deze nieuwe matrices, we op die manier nog geen antwoord gevonden hebben op de vraag welke veeltermen ψ_k we best zouden gebruiken. We hebben vastgesteld dat we zeer nauwkeurige resultaten bekomen indien we de formele orthogonale veeltermen gebruiken. Met andere woorden, de nulpunten van $\varphi_n(z)$ zullen berekend worden uit inwendige producten die te maken hebben met de veeltermen $\varphi_0(z), \varphi_1(z), \ldots, \varphi_{n-1}(z)$. Vooraleer we dit in detail kunnen uitleggen, moeten we echter eerst iets dieper ingaan op de orthogonaliteitseigenschappen van FOP's.

Indien H_n sterk regulier is, i.e., indien alle leidende principale deelmatrices van H_n regulier zijn, dan bestaat de volledige verzameling $\{\varphi_0, \varphi_1, \ldots, \varphi_n\}$ van reguliere FOP's. Wat gebeurt er indien H_n niet sterk regulier is? Door de ontbrekende reguliere FOP's op de juiste manier te definiëren, kan men een rij $\{\varphi_t\}_{t=0}^{\infty}$ bekomen, met φ_t een monische veelterm van graad t, die de volgende eigenschap heeft. Indien deze veeltermen in blokken gegroepeerd worden, waarbij elk blok begint met een reguliere FOP en verder geen reguliere FOP's bevat, dan zijn de veeltermen die tot verschillende blokken behoren onderling orthogonaal met betrekking tot $\langle\cdot,\cdot\rangle$. De rij $\{\varphi_t\}_{t=0}^{\infty}$ wordt als volgt gedefinieerd. Indien t een reguliere index is, dan definiëren we φ_t als de reguliere FOP van graad t. Anders definiëren we φ_t als $\varphi_r\psi_{t,r}$ waarbij r de grootste reguliere index is die kleiner is dan t en $\psi_{t,r}$ een willekeurige monische veelterm van graad $t-r$ is. In dit geval noemen we φ_t een *inwendige veelterm*. Deze veeltermen $\{\varphi_t\}_{t=0}^{\infty}$ kunnen in blokken gegroepeerd worden. Elk blok begint met een reguliere FOP en bevat verder alleen maar inwendige veeltermen. Uit de blok orthogonaliteitseigenschap volgt dan dat de *Gram matrix* $G_n := [\langle\varphi_r, \varphi_s\rangle]_{r,s=0}^{n-1}$ blok diagonaal is. De diagonaalblokken zijn regulier, symmetrisch en nul boven de hoofdantidiagonaal. (Voor meer details verwijzen we naar Bultheel en Van Barel [**47**].)

De vorige stelling kan als volgt geïnterpreteerd worden: de nulpunten van de reguliere FOP van graad n kunnen berekend worden door een veralgemeend eigenwaardenprobleem op te lossen. De volgende stelling toont aan dat dit voor alle reguliere FOP's geldt. Dit zal ons toelaten FOP's te berekenen in hun productvorm. Definieer de matrices G_k en $G_k^{(1)}$ als

$$G_k := \left[\langle\varphi_r, \varphi_s\rangle\right]_{r,s=0}^{k-1} \qquad \text{en} \qquad G_k^{(1)} := \left[\langle\varphi_r, \varphi_1\varphi_s\rangle\right]_{r,s=0}^{k-1}$$

voor $k = 1, 2, \ldots$.

STELLING. Zij $t \geq 1$ een reguliere index en zij $z_{t,1}, \ldots, z_{t,t}$ de nulpunten van de reguliere FOP φ_t. Dan zijn de eigenwaarden van de pencil $G_t^{(1)} - \lambda G_t$ gelijk aan $\varphi_1(z_{t,1}), \ldots, \varphi_1(z_{t,t})$. Met andere woorden, ze zijn gelijk aan $z_{t,1} - \mu, \ldots, z_{t,t} - \mu$ waarbij $\mu = s_1/s_0$.

STELLING. De eigenwaarden van $G_n^{(1)} - \lambda G_n$ zijn gelijk aan $z_1 - \mu, \ldots, z_n - \mu$ waarbij $\mu = s_1/s_0$.

Reguliere FOP's worden gekenmerkt door het feit dat de determinant van een Hankel-matrix verschillend is van nul, terwijl inwendige veeltermen overeenkomen met singuliere Hankel-matrices. Om te beslissen of φ_t als een reguliere FOP of als een inwendige veelterm berekend moet worden, zou men dus de determinant van H_t kunnen berekenen en vergelijken met nul. Bij numerieke berekeningen heeft een dergelijke test "is gelijk aan nul" echter geen zin. Omwille van afrondingsfouten (zowel bij het evalueren van $\langle \cdot, \cdot \rangle$ als bij het berekenen van de determinant) zou men alleen maar reguliere FOP's ontmoeten. Strikt genomen zou men kunnen beweren dat inwendige veeltermen eigenlijk overbodig zijn bij numerieke berekeningen. Het tegendeel is waar! Laten we een reguliere FOP *goed geconditioneerd* noemen indien het overeenkomstige stelsel (3) goed geconditioneerd is. In het andere geval spreken we van een *slecht geconditioneerde* reguliere FOP. Om een numeriek stabiel algoritme te bekomen, is het cruciaal om enkel goed geconditioneerde reguliere FOP's te genereren en slecht geconditioneerde reguliere FOP's te vervangen door inwendige veeltermen.

Ons algoritme voor het berekenen van de onderling verschillende nulpunten $z_1, \ldots, z_n$ berekent achtereenvolgens de veeltermen $\varphi_0(z), \varphi_1(z), \ldots, \varphi_n(z)$ in hun productvorm, te beginnen met $\varphi_0(z) \leftarrow 1$ en $\varphi_1(z) \leftarrow z - \mu$. Het algoritme maakt gebruik van een heuristiek om te beslissen of de volgende veelterm in de rij al dan niet als een reguliere FOP berekend moet worden. We hebben een groot aantal numerieke experimenten uitgevoerd en zijn op die manier tot de conclusie gekomen dat onze heuristiek tot nauwkeurige resultaten leidt. Voor meer details verwijzen we naar de Engelse tekst.

De volgende stelling laat ons toe de waarde van n te bepalen.

STELLING. Zij $t \geq n$. Dan is $\langle z^p, \varphi_t(z) \rangle = 0$ voor alle $p \geq 0$.

Veronderstel dat ons algoritme net een (goed geconditioneerde) reguliere FOP $\varphi_r(z)$ gegenereerd heeft. Om te verifiëren of $n = r$, doorloopt het algoritme de rij

$$\left(|\langle z^\tau \varphi_r(z), \varphi_r(z) \rangle| \right)_{\tau=0}^{N-1-r}.$$

Indien alle elementen "voldoende klein" zijn, dan besluit het algoritme dat inderdaad $n = r$ en het stopt.

We hebben reeds vermeld dat zodra n en (benaderingen voor) $z_1, \ldots, z_n$ gekend zijn, de multipliciteiten $\nu_1, \ldots, \nu_n$ bepaald kunnen worden door het Vandermonde-stelsel (4) op te lossen. We hebben ons algoritme in Matlab geïmplementeerd. In de Engelse tekst illustreren we onze aanpak met behulp van een aantal numerieke voorbeelden.

Het softwarepakket ZEAL

We hebben ons algoritme geïmplementeerd voor het geval dat de kromme γ een rechthoek is waarvan de zijden evenwijdig zijn met de coördinaatassen. Ons pakket heet ZEAL ('ZEros of AnaLytic functions') en is geschreven in Fortran 90. Numerieke benaderingen voor de integralen langs γ worden berekend met behulp van het kwadratuurpakket QUADPACK [**241**].

Voor meer details omtrent de structuur van ZEAL, de in- en uitvoerparameters en een aantal numerieke voorbeelden verwijzen we naar de Engelse tekst.

Een aanpak die de afgeleide f' niet vereist

De resultaten die we tot nu toe besproken hebben, zijn gebaseerd op de bilineaire vorm (1). Deze vorm bevat de logaritmische afgeleide f'/f. Zij

$$\langle \cdot, \cdot \rangle_\star \, : \, \mathcal{P} \times \mathcal{P} \to \mathbb{C}$$

de symmetrische bilineaire vorm gedefinieerd als

$$\langle \phi, \psi \rangle_\star := \frac{1}{2\pi\, i} \int_\gamma \phi(z)\psi(z)\frac{1}{f(z)}\, dz$$

voor elke twee veeltermen $\phi, \psi \in \mathcal{P}$. We tonen aan dat met behulp van deze vorm essentieel dezelfde resultaten behaald kunnen worden als voorheen met $\langle \cdot, \cdot \rangle$. De afgeleide f' is niet vereist. Het spreekt vanzelf dat we niet de onderling verschillende nulpunten zullen kunnen berekenen maar slechts de onbekenden $Z_1, \ldots, Z_N$ (zoals gedefinieerd in de inleiding). De multipliciteiten kunnen niet expliciet berekend worden. Afgezien daarvan geeft deze nieuwe aanpak echter vergelijkbare resultaten. We illustreren dit aan de hand van een aantal numerieke voorbeelden.

2. Clusters van nulpunten van analytische functies

De technieken die we in het vorige hoofdstuk voorgesteld hebben, kunnen ook gebruikt worden om clusters van nulpunten van analytische functies te bepalen, meer bepaald het zwaartepunt van een cluster en het aantal nulpunten dat tot een cluster behoort.

Dit hoofdstuk stemt overeen met ons artikel [**186**].

Veronderstel dat de nulpunten van f die binnen γ liggen, gegroepeerd kunnen worden in m clusters. Zij $I_1, \ldots, I_m$ indexverzamelingen die deze clusters definiëren, en zij

$$\mu_j := \sum_{k \in I_j} \nu_k \qquad \text{en} \qquad c_j := \frac{1}{\mu_j} \sum_{k \in I_j} \nu_k\, z_k$$

voor $j = 1, \ldots, m$. Met andere woorden, μ_j is gelijk aan het totale aantal nulpunten die cluster j vormen (het "gewicht" van de cluster) terwijl c_j het rekenkundige gemiddelde is van de nulpunten uit cluster j (het "zwaartepunt" van de cluster). We veronderstellen dat de zwaartepunten $c_1, \ldots, c_m$ onderling verschillend zijn. Voor $k = 1, \ldots, n$ definiëren we ook $\zeta_k := z_k - c_j$ indien $k \in I_j$. We definiëren de symmetrische bilineaire vorm $\langle \cdot, \cdot \rangle_m$ als volgt:

$$\langle \phi, \psi \rangle_m := \sum_{j=1}^{m} \mu_j \phi(c_j)\psi(c_j)$$

voor elke twee veeltermen $\phi, \psi \in \mathcal{P}$. Definieer

$$\delta := \max_{1 \le k \le n} |\zeta_k|.$$

De volgende stelling leert ons dat de vormen $\langle \cdot, \cdot \rangle_m$ en $\langle \cdot, \cdot \rangle$ elkaar benaderen.

STELLING. Zij $\phi, \psi \in \mathcal{P}$. Dan geldt dat $\langle \phi, \psi \rangle = \langle \phi, \psi \rangle_m + \mathcal{O}(\delta^2)$, $\delta \to 0$.

De volgende stelling zal ons toelaten benaderingen voor de zwaartepunten te bekomen.

STELLING. Zij t een geheel getal $\geq m$. Dan geldt dat $\varphi_t(c_j) = \mathcal{O}(\delta^2)$, $\delta \to 0$ voor $j = 1, \ldots, m$. Ook geldt dat $\langle z^p, \varphi_t(z) \rangle = \mathcal{O}(\delta^2)$, $\delta \to 0$ voor alle $p \geq t$.

Met andere woorden, wellicht zullen we, tenzij de FOP $\varphi_t(z)$ een zeer vlak verloop heeft in de buurt van zijn nulpunten, goede benaderingen voor de zwaartepunten $c_1, \ldots, c_m$ aantreffen tussen de nulpunten van $\varphi_t(z)$ en dit voor alle $t \geq m$. Noteer dat

$$\begin{bmatrix} 1 & \cdots & 1 \\ c_1 & \cdots & c_m \\ \vdots & & \vdots \\ c_1^{m-1} & \cdots & c_m^{m-1} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{m-1} \end{bmatrix} + \mathcal{O}(\delta^2), \ \delta \to 0.$$

Bijgevolg kunnen we benaderingen voor de gewichten $\mu_1, \ldots, \mu_m$ bekomen door een Vandermonde-stelsel op te lossen.

In de Engelse tekst zullen we in detail illustreren hoe deze stellingen ons toelaten om met behulp van het algoritme dat we in het vorige hoofdstuk voorgesteld hebben clusters te bepalen. We zullen ook een equivalente aanpak bespreken die gebaseerd is op rationale interpolatie.

3. Nulpunten en polen van meromorfe functies

Veronderstel nu dat f niet analytisch maar meromorf is in W en dat f nulpunten noch polen heeft op γ. Zij P het totale aantal polen van f die binnen γ liggen. Veronderstel dat $N + P > 0$. Zij p het aantal onderling verschillende polen van f die binnen γ liggen. Zij $y_1, \ldots, y_p$ deze polen en $\mu_1, \ldots, \mu_p$ hun respectievelijke ordes. Men kan eenvoudig nagaan dat f'/f een enkelvoudige pool heeft in y_l met residu $-\mu_l$ voor $l = 1, \ldots, p$. Bijgevolg geldt dat

$$\langle \phi, \psi \rangle = \sum_{k=1}^{n} \nu_k \phi(z_k) \psi(z_k) - \sum_{l=1}^{p} \mu_l \phi(y_l) \psi(y_l).$$

Deze uitdrukking is van dezelfde vorm als (1). De resultaten behaald in het eerste hoofdstuk maken geen gebruik van het feit dat de multipliciteiten ν_k positieve gehele getallen zijn. Ze steunen enkel op het feit dat $\nu_k \neq 0$ voor $k = 1, \ldots, n$. We tonen aan dat, op voorwaarde dat een bovengrens voor P gekend is, ons algoritme uit het eerste hoofdstuk vrij eenvoudig aangepast kan worden om alle nulpunten en polen van f te berekenen die binnen γ liggen.

Dit hoofdstuk stemt overeen met ons artikel [**191**].

4. Stelsels analytische vergelijkingen

In dit hoofdstuk tonen we aan hoe de logaritmische residu-technieken uit het eerste hoofdstuk gebruikt kunnen worden om stelsels analytische vergelijkingen op te lossen. Dit hoofdstuk stemt overeen met ons artikel [**183**].

Zij $d \geq 1$ een positief geheel getal. Beschouw een polyschijf D in $\mathbb{C}^d$ (i.e., D is het Cartesische product van d schijven in $\mathbb{C}$) en zij $f = (f_1, \ldots, f_d) : \overline{D} \to \mathbb{C}^d$ een afbeelding die analytisch is in $\overline{D}$ en geen nulpunten heeft op de rand van D. We wensen alle nulpunten van f te berekenen die in D liggen, samen met hun respectievelijke multipliciteit. Zij $Z_f(D)$ de verzameling der nulpunten van f die in D liggen en zij $\mu_a(f)$ de multipliciteit van een nulpunt $a \in Z_f(D)$.

Indien $d = 1$, dan leert ons de klassieke formule die we in de vorige hoofdstukken gebruikt hebben dat

$$\frac{1}{2\pi\, i} \int_{\partial D} \varphi(z)\, \frac{f'(z)}{f(z)}\, dz = \sum_{a \in Z_f(D)} \mu_a(f)\, \varphi(a)$$

indien $\varphi : \overline{D} \to \mathbb{C}$ analytisch is in D en continu in $\overline{D}$.

De volgende stelling is een meerdimensionale versie van dit resultaat. Zij J_f de Jacobiaanmatrix van f en zij $J_{[k]}$ de Jacobiaanmatrix van f met de $k$de kolom vervangen door $\begin{bmatrix} f_1 & \cdots & f_d \end{bmatrix}^T$:

$$J_{[k]} := \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_{k-1}} & f_1 & \frac{\partial f_1}{\partial z_{k+1}} & \cdots & \frac{\partial f_1}{\partial z_d} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_d}{\partial z_1} & \cdots & \frac{\partial f_d}{\partial z_{k-1}} & f_d & \frac{\partial f_d}{\partial z_{k+1}} & \cdots & \frac{\partial f_d}{\partial z_d} \end{bmatrix}, \qquad k = 1, \ldots, d.$$

Veronderstel dat de polyschijf D kan geschreven worden als

$$D = D_1 \times \cdots \times D_d$$

waarbij

$$D_k = \{\, z \in \mathbb{C} : |z - C_k| < R_k \,\}, \qquad k = 1, \ldots, d,$$

met $C_1, \ldots, C_d \in \mathbb{C}$ en $R_1, \ldots, R_d > 0$.

STELLING. Zij $\varphi : \overline{D} \to \mathbb{C}$ analytisch in D en continu in $\overline{D}$. Definieer $I_k(\varphi)$ voor $k = 1, \ldots, d$ als de integraal

$$I_k(\varphi) := \rho_k \int_{[0,1]^{2d-1}} \frac{\varphi(z_1, \ldots, z_d)\, \det J_f(z_1, \ldots, z_d)\, \overline{\det J_{[k]}(z_1, \ldots, z_d)}}{\left(|f_1(z_1, \ldots, z_d)|^2 + \cdots + |f_d(z_1, \ldots, z_d)|^2\right)^d} \times e^{2\pi i\, \theta_k}\, r_1 \cdots r_{k-1}\, r_{k+1} \cdots r_d\, dr_1 \cdots dr_{k-1}\, dr_{k+1} \cdots dr_d\, d\theta_1 \cdots d\theta_d$$

met

$$\rho_k = \rho(d, R_1, \ldots, R_d; k) := 2^{d-1}(d-1)!\, R_1^2 \cdots R_{k-1}^2\, R_k\, R_{k+1}^2 \cdots R_d^2$$

en waarbij

$$z_k = z_k(\theta_k) = C_k + R_k\, e^{2\pi i\, \theta_k} \qquad 0 \leq \theta_k \leq 1$$

en

$$z_l = z_l(r_l, \theta_l) = C_l + r_l\, R_l\, e^{2\pi i\, \theta_l} \qquad 0 \leq r_l, \theta_l \leq 1$$

voor $l \in \{1,\ldots,d\} \setminus \{k\}$. Dan geldt dat

$$I(\varphi) := \sum_{k=1}^{d} I_k(\varphi) = \sum_{a \in Z_f(D)} \mu_a(f)\,\varphi(a).$$

De stelling zal ons toelaten om met behulp van numerieke integratie informatie te bekomen over de ligging van de nulpunten van f. Op basis van deze informatie kunnen deze nulpunten en hun respectievelijke multipliciteiten berekend worden door een veralgemeend eigenwaardenprobleem op te lossen en d Vandermonde-stelsels. Voor meer details verwijzen we naar de Engelse tekst.

Deel 2

5. Het berekenen van enkelvoudige nulpunten van Besselfuncties

In dit hoofdstuk vestigen we onze aandacht op het probleem van het berekenen van enkelvoudige nulpunten van analytische functies. In het bijzonder beschouwen we de Besselfuncties $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$, $H_\nu^{(2)}(z)$ en hun eerste afgeleide, waarbij de veranderlijke z tot $\mathbb{C} \setminus (-\infty, 0]$ behoort en de orde ν reëel is. We stellen de lezer ons softwarepakket ZEBEC voor, waarmee voor elk van deze functies alle nulpunten berekend kunnen worden die binnen een rechthoek liggen waarvan de zijden evenwijdig met de coördinaatassen zijn.

Dit hoofdstuk stemt overeen met ons artikel [**185**].

Inleiding

De Besselvergelijking

$$z^2u''(z) + zu'(z) + (z^2 - \nu^2)u(z) = 0, \qquad z, \nu \in \mathbb{C}, \tag{6}$$

verschijnt in heel wat vraagstukken uit de wiskundige natuurkunde. De functies $J_\nu(z)$, de Besselfunctie van de eerste soort en van orde ν, en $Y_\nu(z)$, de Besselfunctie van de tweede soort en van orde ν, zijn twee lineair onafhankelijke oplossingen van deze vergelijking. Ook van belang zijn de Besselfuncties van de derde soort,

$$H_\nu^{(1)}(z) := J_\nu(z) + i\,Y_\nu(z) \quad \text{en} \quad H_\nu^{(2)}(z) := J_\nu(z) - i\,Y_\nu(z).$$

Deze functies zijn analytisch in $\mathbb{C} \setminus (-\infty, 0]$.

De nulpunten en keerpunten (i.e, nulpunten van de eerste afgeleide) van Besselfuncties spelen een belangrijke rol in de natuurkunde en de ingenieurswetenschappen. Ze komen voor bij de studie van de trillingen van een membraan, de temperatuursverdeling in een cilinder of een bol, de diffractie van een vlakke elektromagnetische golf door een geleidende cilinder, enz. Voor een elektrostatische interpretatie van de nulpunten van Besselfuncties verwijzen we naar [**220**].

In [**240, 242**] werd een softwarepakket voorgesteld voor het berekenen van nulpunten van $J_\nu(x)$ indien $x > 0$ en $\nu > -1$ en keerpunten van $J_\nu(x)$ indien $x > 0$ en $\nu > 0$. Het pakket RFSFNS [**291**] kan gebruikt worden om nulpunten en keerpunten van $J_\nu(x)$ en $Y_\nu(x)$ te berekenen indien $x > 0$ en $\nu \geq 0$. Onlangs hebben Segura

en Gil [**253**] twee algoritmes voorgesteld om reële nulpunten van $J_\nu(x)$ te berekenen indien de orde ν reëel is (zowel positief als negatief). In dit hoofdstuk stellen we een softwarepakket voor om nulpunten of keerpunten van $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ en $H_\nu^{(2)}(z)$ te berekenen indien de veranderlijke z tot $\mathbb{C} \setminus (-\infty, 0]$ behoort en de orde ν reëel is. Ons softwarepakket heet ZEBEC: 'ZEros of BEssel functions Complex.' Het kan alle nulpunten of keerpunten berekenen die in een rechthoek liggen waarvan de zijden evenwijdig zijn met de coördinaatassen. Elke oplossing van (6) heeft alleen maar enkelvoudige nulpunten, tenzij eventueel in het punt $z = 0$ [**295**, p. 479]. Dit geldt in het bijzonder voor $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ en $H_\nu^{(2)}(z)$. De afgeleide van een oplossing van (6) heeft ook slechts enkelvoudige nulpunten, tenzij eventueel in $z = 0$ of $z = \pm\nu$ [**175, 176, 267**]. De nulpunten die we wensen te berekenen zijn dus allemaal enkelvoudig, tenzij eventueel in $z = 0$ en $z = \pm\nu$. Vermits we veronderstellen dat het argument z tot $\mathbb{C} \setminus (-\infty, 0]$ behoort en de orde ν reëel is, hoeven we $z = 0$ en één van de punten $z = \pm\nu$ niet te beschouwen. Door de functie en haar eerste afgeleide te evalueren in $z = \nu$ (indien $\nu > 0$) of $z = -\nu$ (indien $\nu < 0$), kan men eenvoudig nagaan of de functie in dit punt een meervoudig nulpunt heeft of niet. Indien dit het geval is, dan waarschuwt ZEBEC de gebruiker en vraagt om een ander rechthoekig gebied. In het andere geval beginnen de berekeningen.

Enkelvoudige nulpunten van analytische functies

De nulpunten van Besselfuncties die we wensen te berekenen, zijn dus allen enkelvoudig. Vandaar dat we eerst het volgende, iets algemenere probleem beschouwen. Zij f een analytische functie met alleen maar enkelvoudige nulpunten. Om alle nulpunten van f te berekenen die binnen een rechthoek liggen waarvan de zijden evenwijdig zijn met de coördinaatassen gaan we als volgt te werk:

(1) We berekenen het totale aantal nulpunten die binnen deze rechthoek liggen.
(2) We verdelen deze rechthoek onder in deelrechthoeken tot we alle nulpunten geïsoleerd hebben. Op die manier bekomen we een verzameling rechthoeken die elk precies één nulpunt bevatten.
(3) We berekenen de nulpunten die in elk van deze deelrechthoeken liggen.

Tijdens de eerste en de tweede fase maken we gebruik van de logaritmische residuintegraal

$$\frac{1}{2\pi i} \int_\gamma \frac{f'(z)}{f(z)}\, dz.$$

Hierbij is de kromme γ een rechthoek en we tonen aan hoe in dit geval deze integraal geschreven kan worden als een som van vier Riemann-integralen. Deze integralen benaderen we met behulp van het integratiepakket QUADPACK [**241**]. Tijdens de derde fase maken we gebruik van het pakket CHABIS [**288, 289**]. Dit pakket bevat een implementatie van een veralgemeende bisectiemethode die karakteristieke bisectie genoemd wordt. Het is gebaseerd op de theorie van de topologische graad.

Het softwarepakket ZEBEC

Ons pakket ZEBEC is gebaseerd op het vorige algemene schema voor het berekenen van enkelvoudige nulpunten van analytische functies. Om de Besselfuncties te evalueren maken we gebruik van het pakket BESSCC [**268**]. Voor een gedetailleerde

beschrijving van de structuur en de invoerparameters van het pakket verwijzen we naar de Engelse tekst. We illustreren hoe ZEBEC gebruikt kan worden om alle nulpunten van $Y_{-15.3}(z)$ te berekenen die gelegen zijn in het rechthoekige gebied

$$\{\, z \in \mathbb{C} \,:\, -22 \leq \mathrm{Re}\, z \leq 23, \quad 0.5 \leq \mathrm{Im}\, z \leq 100.5 \,\}.$$

6. De nulpunten van $J_n(z) \pm iJ_{n+1}(z)$ en $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$

In dit hoofdstuk beschouwen we de complexe nulpunten van de functies

$$z \mapsto J_n(z) \pm iJ_{n+1}(z)$$

en

$$z \mapsto [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$$

waarbij $n \in \mathbb{N}$ en $J_n(z)$ de Besselfunctie van de eerste soort en orde n voorstelt. Definieer $F_n(z) := J_n(z) - iJ_{n+1}(z)$ en $G_n(z) := [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$.

De nulpunten van deze functies spelen een belangrijke rol in bepaalde fysische toepassingen. De nulpunten van $F_0(z)$ zijn van belang bij de studie van watergolven die een hellend strand ontmoeten, cf. Synolakis [**263, 264**]. De vergelijking $F_n(z) = 0$ verschijnt bij stranden met meerdere hellingen [**266**]. MacDonald [**205**] gebruikte de nulpunten van $G_0(z)$ om representatieve stroomlijnen te tekenen voor de stationaire stroming van een viskeuze vloeistof in een lange buis met constante straal die rond haar as (de $\hat{z}$-as) draait met een hoeksnelheid die bij $\hat{z} = 0$ op een discontinue manier verandert van de ene constante waarde naar een andere constante waarde met hetzelfde teken.

MacDonald [**204, 205, 206**] leidde asymptotische formules af voor de nulpunten van $F_0(z)$, $F_n(z)$ en $G_n(z)$. Deze formules laten toe nulpunten te benaderen die een grote modulus hebben. MacDonald stelde vast dat de nauwkeurigheid van de asymptotische formules voor de nulpunten van $F_n(z)$ vermindert naarmate n toeneemt. Een ander nadeel van deze asymptotische formules is dat ze vaak zeer onnauwkeurige resultaten geven voor kleinere nulpunten. Deze moeten dan via een of andere numerieke procedure berekend worden. MacDonald [**206**] ging uit van een afgebroken machtreeks voor $F_n(z)$ en berekende de nulpunten daarvan via de methode van Newton.

We stellen voor om de nulpunten van $F_n(z)$ en $G_n(z)$ te berekenen met behulp van ons softwarepakket ZEBEC. Op de eerste plaats is dit pakket ontworpen om nulpunten van de Besselfuncties $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$, $H_\nu^{(2)}(z)$ en hun eerste afgeleide te berekenen, maar het pakket kan eenvoudig aangepast worden om nulpunten van een willekeurige analytische functie te berekenen, op voorwaarde natuurlijk dat deze nulpunten enkelvoudig zijn. In $z = 0$ hebben $F_n(z)$ en $G_n(z)$ een nulpunt met multipliciteit n (indien $n \geq 1$) en $2n + 2$, respectievelijk. We bewijzen dat alle nulpunten in $\mathbb{C}_0$ enkelvoudig zijn. Ook tonen we aan dat er geen nulpunten op de coördinaatassen liggen (behalve dan in $z = 0$). Ter illustratie gebruiken we ZEBEC om de eerste 30 nulpunten van $J_5(z) - iJ_6(z)$ te berekenen die in het vierde kwadrant gelegen zijn.

Dit hoofdstuk stemt overeen met ons artikel [**193**].

Deel 3

7. De methode van Newton voor meervoudige nulpunten

In dit hoofdstuk stellen we een aanpassing van de methode van Newton voor, specifiek voor het berekenen van meervoudige oplossingen van stelsels analytische vergelijkingen. Onze iteratie convergeert kwadratisch onder milde voorwaarden. Ze bevat bepaalde constanten waarvan het product een benedengrens vormt voor de multipliciteit van het nulpunt. Deze constanten zijn zelden op voorhand gekend. Vandaar dat we een iteratieschema opstellen waarin niet alleen een benadering voor het nulpunt verfijnd wordt, maar ook benaderingen voor deze constanten.

Dit hoofdstuk stemt overeen met ons artikel [**184**].

Inleiding

Zij $f : \mathbb{C} \to \mathbb{C}$ een zachte functie en veronderstel dat $z^\star$ een nulpunt van f is met multipliciteit μ. Indien $\mu = 1$, dan convergeert de methode van Newton kwadratisch naar $z^\star$ indien de startwaarde van de iteratie voldoende dicht in de buurt van $z^\star$ ligt. Indien $\mu > 1$, dan convergeert de iteratie slechts lineair. Indien μ gekend is, dan kan men opnieuw kwadratische convergentie bekomen via de iteratie

$$z^{(p+1)} = z^{(p)} - \mu \frac{f(z^{(p)})}{f'(z^{(p)})}, \qquad p = 0, 1, 2, \ldots . \tag{7}$$

Van de Vel [**281, 283**] ontwierp een iteratie waarin niet alleen een benadering voor het nulpunt verfijnd wordt, maar ook een schatting van de multipliciteit. King [**177**] analyseerde de methode van Van de Vel en toonde aan dat de convergentieorde gelijk is aan 1.554. Hij herschikte de volgorde van de berekeningen en bekwam op die manier een iteratie met convergentieorde gelijk aan 1.618. Zijn iteratieschema ziet er als volgt uit:

$$\left\{ \begin{array}{rcl} \mu^{(p+1)} & = & \dfrac{u(z^{(p)})}{u(z^{(p)}) - u(z^{(p+1)})} \mu^{(p)} \\ z^{(p+2)} & = & z^{(p+1)} - \mu^{(p+1)} u(z^{(p+1)}) \end{array} \right. \tag{8}$$

voor $p = 0, 1, 2, \ldots$, waarbij $z^{(0)}$ en $\mu^{(0)}$ gegeven zijn en $z^{(1)} := z^{(0)} - \mu^{(0)} u(z^{(0)})$. Hierbij is $u(z) := f(z)/f'(z)$.

We veralgemenen deze resultaten naar meerdere dimensies. We beschouwen stelsels analytische vergelijkingen, i.e., analytische afbeeldingen $f : \mathbb{C}^n \to \mathbb{C}^n$.

Een wijziging van de methode van Newton

Zij $f = f(z) : \mathbb{C}^n \to \mathbb{C}^n$ een analytische afbeelding, waarbij $z = (z_1, \ldots, z_n)$ en $f = (f_1, \ldots, f_n)$. Een punt $z^\star \in \mathbb{C}^n$ heet een nulpunt van f indien $f(z^\star) = 0$. Een geïsoleerd nulpunt $z^\star$ van f heet enkelvoudig indien de Jacobiaanmatrix van f regulier is in $z^\star$. De volgende stelling is overgenomen uit het bekende boek van Aĭzenberg en Yuzhakov [**3**].

STELLING. *Indien de sluiting van een omgeving $U_{z^\star}$ van een nulpunt $z^\star$ van f geen ander nulpunt van f bevat, dan bestaat er een $\epsilon > 0$ zodanig dat voor bijna elke $\zeta \in \mathbb{C}^n$, $\|\zeta\|_2 < \epsilon$, de afbeelding*

$$z \mapsto f(z) - \zeta \tag{9}$$

alleen maar enkelvoudige nulpunten heeft in $U_{z^\star}$. Het aantal nulpunten hangt niet af van de keuze van ζ of van de omgeving $U_{z^\star}$.

Het aantal nulpunten van de afbeelding (9) die in de omgeving $U_{z^\star}$ liggen, wordt de *multipliciteit* van $z^\star$ genoemd en genoteerd als $\mu_{z^\star}(f)$. De multipliciteit van een geïsoleerd nulpunt van een analytische afbeelding is dus gelijk aan het aantal enkelvoudige nulpunten waarin dit nulpunt uiteenvalt onder een voldoende kleine perturbatie van de afbeelding.

Zij nu $z^\star = (z_1^\star, \ldots, z_n^\star)$ een geïsoleerd nulpunt van $f = (f_1, \ldots, f_n)$ zodanig dat

$$f_j(z) = \sum_{|\alpha| \geq k_j} c_{j,\alpha}(z - z^\star)^\alpha$$

voor $j = 1, \ldots, n$ waarbij α een multi-index is, $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, $|\alpha| = \alpha_1 + \cdots + \alpha_n$ en $(z - z^\star)^\alpha = (z_1 - z_1^\star)^{\alpha_1} \cdots (z_n - z_n^\star)^{\alpha_n}$. We noemen k_j de *orde* van $z^\star$ als nulpunt van f_j. Definieer

$$P_j(z) := \sum_{|\alpha| = k_j} c_{j,\alpha}(z - z^\star)^\alpha$$

voor $j = 1, \ldots, n$. De homogene polynomiale afbeelding

$$P = P(z) := (P_1(z), \ldots, P_n(z)) \tag{10}$$

wordt het *homogene principale deel* van f in $z^\star$ genoemd. De volgende stelling van Tsikh en Yuzhakov legt een verband tussen de multipliciteit $\mu_{z^\star}(f)$ en de ordes $k_1, \ldots, k_n$.

STELLING. *De multipliciteit van een geïsoleerd nulpunt $z^\star$ van f is gelijk aan het product van de ordes van $z^\star$ als nulpunt van $f_1, \ldots, f_n$ als en slechts als $z^\star$ een geïsoleerd nulpunt is van de afbeelding* (10). *Er geldt dat $\mu_{z^\star}(f) \geq k_1 \cdots k_n$.*

De volgende stelling is onze veralgemening van iteratie (7).

STELLING. *Veronderstel dat $z^{(0)}$ zodanig is dat de iteratie*

$$z^{(p+1)} = z^{(p)} - [f'(z^{(p)})]^{-1} \operatorname{diag}(k_1, \ldots, k_n) f(z^{(p)}), \qquad p = 0, 1, 2, \ldots,$$

goed gedefinieerd is voor elke p. Indien $\det P'(z) \not\equiv 0$ en indien $z^{(0)}$ voldoende dicht in de buurt van $z^\star$ ligt, dan convergeert $z^{(p)}$ kwadratisch naar $z^\star$. Indien $\det P'(z) \equiv 0$, dan is de convergentie slechts lineair.

In de Engelse tekst geven we een aantal numerieke voorbeelden om deze stelling te illustreren.

De methode van Van de Vel

De iteratie uit de vorige stelling kan slechts toegepast worden indien de ordes $k_1, \ldots, k_n$ gekend zijn. We leiden een iteratieschema af waarbij zowel een benadering voor het nulpunt als benaderingen voor de ordes verfijnd worden. De benaderingen voor de ordes $k_1, \ldots, k_n$ in de $p$de iteratiestap noteren we als $d_1^{(p)}, \ldots, d_n^{(p)}$. Zij

$$d^{(p)} := \begin{bmatrix} d_1^{(p)} \\ \vdots \\ d_n^{(p)} \end{bmatrix}$$

en definieer de matrixfunctie

$$U(z) := [f'(z)]^{-1} \operatorname{diag}(f_1(z), \ldots, f_n(z))$$

voor elke $z \in \mathbb{C}^n$ waarvoor $f'(z)$ regulier is. Dan ziet onze veralgemening van (8) er als volgt uit:

$$\begin{aligned} d^{(p+1)} &= \left[U(z^{(p)}) - U\big(z^{(p)} - U(z^{(p)})d^{(p)}\big) \right]^{-1} U(z^{(p)})d^{(p)} \\ z^{(p+1)} &= \big(z^{(p)} - U(z^{(p)})d^{(p)}\big) - U\big(z^{(p)} - U(z^{(p)})d^{(p)}\big)d^{(p+1)} \end{aligned}$$

voor $p = 0, 1, 2, \ldots$, gegeven een schatting $z^{(0)}$ voor $z^\star$ en schattingen $d^{(0)}$ voor de ordes. Dit is onze meerdimensionale versie van de methode van Van de Vel. We illustreren deze iteratiemethode met behulp van een aantal voorbeelden. Deze suggereren dat de convergentieorde dezelfde is als in het eendimensionale geval.

Deel 4

8. Supersnelle rationale interpolatie

We hebben rationale interpolatie al ontmoet op het einde van Hoofdstuk 2 bij het bepalen van clusters van nulpunten van analytische functies. In dit hoofdstuk gaan we dieper in op de problematiek van het oplossen van rationale interpolatieproblemen. De meeste gekende algoritmes zijn numeriek onstabiel. We spitsen onze aandacht toe op het algoritme van Van Barel en Bultheel [**274**]. We tonen aan hoe dit algoritme gestabiliseerd kan worden via pivotering. Met behulp van een verdeel-en-heers strategie bekomen we ook een recursief algoritme dat we stabiliseren door zogenaamde "moeilijke interpolatiepunten" een speciale behandeling te geven, met behulp van iteratieve verfijning gebaseerd op de inverse van een gekoppelde Vandermonde-matrix en met behulp van "downdating."

Dit hoofdstuk is vrij technisch van aard. Het is bedoeld om samen met het volgende hoofdstuk gelezen te worden, waarin we onze interpolatiealgoritmes zullen gebruiken bij het oplossen van stelsels lineaire vergelijkingen die Hankel- of Toeplitzstructuur hebben. Dit hoofdstuk is gedeeltelijk terug te vinden in onze artikels [**187**], [**189**], [**278**] en [**279**].

Het algoritme RATINT

Zij n een positief geheel getal. Veronderstel dat $s_1, \ldots, s_{2n} \in \mathbb{C}$ en $f_1, \ldots, f_{2n} \in \mathbb{C}^{2\times 1}$ gegeven zijn. Beschouw het interpolatieprobleem

$$f_k^T B(s_k) = [\,0 \;\; 0\,], \qquad k = 1, \ldots, 2n, \tag{11}$$

waarbij

$$B(z) = \begin{bmatrix} n_\ell(z) & n_r(z) \\ d_\ell(z) & d_r(z) \end{bmatrix} \in \mathbb{C}[z]^{2\times 2}$$

met $\deg n_\ell(z) = n$, $\deg d_\ell(z) \leq n-1$, $\deg n_r(z) \leq n-1$ en $\deg d_r(z) = n$, en zowel $n_\ell(z)$ als $d_r(z)$ monisch, i.e., waarbij $B(z)$ een monische 2×2 matrixveelterm van graad n is. Veronderstel dat dit interpolatieprobleem precies één oplossing $B^\star(z)$ heeft. In het volgende hoofdstuk zullen we rationale interpolatieproblemen ontmoeten die eenvoudig op deze manier geformuleerd kunnen worden.

Ons algoritme RATINT berekent een 2×2 matrixveelterm $B(z)$ van graad n die aan (11) voldoet en die een reguliere hoogstegraadscoëfficiënt $A \in \mathbb{C}^{2\times 2}$ heeft. Dit impliceert dat $B(z) \equiv B^\star(z)A$. We tonen aan dat $\det A = 1$. Ons algoritme is gebaseerd op het algoritme van Van Barel en Bultheel. Het maakt gebruik van pivotering om de numerieke stabiliteit te verbeteren. Het vereist $\mathcal{O}(n^2)$ flops en behoort dus tot de categorie der snelle algoritmes. Supersnelle algoritmes vereisen slechts $\mathcal{O}(n\log^2 n)$ flops.

We geven ook een matrixversie van RATINT. In dit geval behoren de vectoren f_k tot $\mathbb{C}^{2p\times p}$ waarbij p een positief geheel getal is. Voor meer details verwijzen we naar de Engelse tekst.

Een gestabiliseerde verdeel-en-heers aanpak

Ons algoritme RATINT gaat sequentieel te werk: de interpolatiepunten worden één na één afgehandeld. (Pivotering zorgt ervoor dat de volgorde der interpolatiepunten eventueel omgewisseld wordt.) In plaats van alle interpolatiepunten in één keer te beschouwen, kan men de verzameling interpolatiepunten echter ook in twee (gelijke) delen opsplitsen. De overeenkomstige interpolatieproblemen kunnen dan afzonderlijk opgelost worden. We tonen aan hoe beide oplossingen samengevoegd kunnen worden tot de oplossing van het oorspronkelijke probleem. Dit schema kan recursief toegepast worden. De interpolatieproblemen op het allerlaagste niveau (met het kleinste aantal interpolatiepunten) worden opgelost met behulp van het snelle algoritme RATINT. Dit leidt tot een supersnel algoritme dat we stabiliseren door de technieken die we hoger reeds vermeld hebben.

9. Supersnelle algoritmes voor het oplossen van stelsels lineaire vergelijkingen met Hankel- of Toeplitz-structuur

Hankel- en Toeplitz-matrices komen in heel wat toepassingen voor, bijvoorbeeld bij signaalverwerking of Markov-ketens [**34, 213**]. Ze spelen ook een centrale rol in de theorie der formele orthogonale veeltermen en bij Padé-benadering. We hebben Hankel-matrices al ontmoet in het eerste deel van dit proefschrift. In dit hoofdstuk stellen we snelle en supersnelle algoritmes voor om stelsels lineaire vergelijkingen

met Hankel- of Toeplitz-structuur op te lossen. 'Snel' betekent dat een $n \times n$ stelsel opgelost wordt met $\mathcal{O}(n^2)$ flops. Een 'supersnel' algoritme vereist $\mathcal{O}(n \log^2 n)$ flops.

Dit hoofdstuk is gedeeltelijk terug te vinden in onze artikels [**187**], [**189**], [**278**] en [**279**].

Stelsels met Hankel-structuur

Zij n een positief geheel getal, zij $H = H_n := [h_{k+l}]_{k,l=0}^{n-1}$ een reguliere complexe $n \times n$ Hankel-matrix en zij $b \in \mathbb{C}^n$. We spitsen onze aandacht toe op het berekenen van $x := H^{-1}b$.

Transformatie naar een stelsel met Loewner-structuur

Zij $y_1, \ldots, y_n, z_1, \ldots, z_n$ onderling verschillende complexe getallen en definieer $\mathbf{y} := (y_1, \ldots, y_n)$ en $\mathbf{z} := (z_1, \ldots, z_n)$. Zij $\mathcal{L}(\mathbf{y}, \mathbf{z})$ de verzameling matrices

$$\mathcal{L}(\mathbf{y}, \mathbf{z}) := \left\{ \left[\frac{c_k - d_l}{y_k - z_l} \right]_{k,l=1}^{n} \mid c_1, \ldots, c_n, d_1, \ldots, d_n \in \mathbb{C} \right\}.$$

De elementen van $\mathcal{L}(\mathbf{y}, \mathbf{z})$ worden *Loewner-matrices* genoemd. Fiedler [**79**] toonde aan hoe Hankel-matrices in Loewner-matrices getransformeerd kunnen worden en omgekeerd. Om dit resultaat te kunnen formuleren, moeten we eerst nog enkele notaties invoeren.

Zij $t_1, \ldots, t_n$ complexe getallen en definieer $\mathbf{t} := (t_1, \ldots, t_n)$.

Zij $f_{\mathbf{t}}(z)$ de monische veelterm van graad n met nulpunten $t_1, \ldots, t_n$,

$$f_{\mathbf{t}}(z) := (z - t_1) \cdots (z - t_n),$$

en definieer

$$f_{\mathbf{t},j}(z) := \prod_{k \neq j} (z - t_k), \qquad j = 1, \ldots, n.$$

Noteer dat $f_{\mathbf{t},j}(z)$ een monische veelterm van graad $n-1$ is voor $j = 1, \ldots, n$. Definieer de $n \times n$ matrix $W(\mathbf{t})$ met behulp van de vergelijking

$$\begin{bmatrix} f_{\mathbf{t},1}(z) \\ f_{\mathbf{t},2}(z) \\ \vdots \\ f_{\mathbf{t},n}(z) \end{bmatrix} = W(\mathbf{t}) \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{n-1} \end{bmatrix}.$$

De $j$de rij van $W(\mathbf{t})$ bevat dus de coëfficiënten van $f_{\mathbf{t},j}(z)$ in termen van de basis $\{1, z, \ldots, z^{n-1}\}$.

STELLING. De matrix $L := W(\mathbf{y})\, H\, [W(\mathbf{z})]^T$ is een Loewner-matrix uit $\mathcal{L}(\mathbf{y}, \mathbf{z})$. De parameters $c_1, \ldots, c_n, d_1, \ldots, d_n$ zijn (op een willekeurige constante $\xi \in \mathbb{C}$ na) gelijk aan

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = W(\mathbf{y}, \mathbf{z}) \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{2n-2} \\ \xi \end{bmatrix}.$$

Zij $\omega := \exp(2\pi\, i/n)$ en $\zeta := \exp(\pi\, i/n)$. Veronderstel nu dat de punten $\mathbf{y}$ en $\mathbf{z}$ gedefinieerd worden als

$$\mathbf{y} = (1, \omega, \ldots, \omega^{n-1}) \qquad \text{en} \qquad \mathbf{z} = (\zeta, \zeta\omega, \ldots, \zeta\omega^{n-1}).$$

Dan kan het Hankel-stelsel $Hx = b$ in $\mathcal{O}(n \log n)$ flops getransformeerd worden naar een Loewner-stelsel $Lx' = b'$. Voor meer details verwijzen we naar de Engelse tekst. Het conditiegetal verandert hierbij niet: $\kappa_2(L) = \kappa_2(H)$.

Een inversieformule voor Loewner-matrices

Van Barel en Vavřín [**280, 284**] hebben een expliciete formule afgeleid voor de inverse van een Loewner-matrix. Deze formule bevat een aantal parameters die berekend kunnen worden door twee rationale interpolatieproblemen op te lossen. Zodra deze parameters gekend zijn, kan de oplossing van het Loewner-stelsel berekend worden als $x' = L^{-1}b'$. Dit vereist slechts $\mathcal{O}(n \log n)$ flops.

Snelle en supersnelle algoritmes voor het oplossen van Hankel-stelsels

Door deze interpolatieproblemen op te lossen met behulp van RatInt bekomen we een snel algoritme voor het oplossen van Hankel-stelsels. Deze aanpak vereist $6n^2 + \mathcal{O}(n)$ vermenigvuldigingen en evenveel optellingen. In onze numerieke voorbeelden beschouwen we verschillende types Hankel-matrices. We laten n toenemen tot 10000.

We leiden ook een matrixversie af van deze transformatie-aanpak. Met behulp van de matrixversie van RatInt bekomen we op die manier een snel algoritme voor het oplossen van blok Hankel-stelsels.

In plaats van RatInt kunnen we ook gebruik maken van ons supersnel verdeel-en-heers algoritme. Dit leidt dan tot een supersnel en gestabiliseerd algoritme voor het oplossen van Hankel-stelsels, weliswaar enkel voor stelsels waarvoor n een macht van 2 is. In onze numerieke voorbeelden laten we n toenemen tot $2^{17} = 131072$.

Stelsels met Toeplitz-structuur

In de plaats van Hankel-stelsels beschouwen we nu Toeplitz-stelsels. We maken opnieuw gebruik van een expliciete formule voor de inverse. Deze formule bevat bepaalde veeltermen die we weerom berekenen door rationale interpolatieproblemen op te lossen. Hiervoor doen we beroep op ons supersnel verdeel-en-heers algoritme. Dit leidt dan tot een supersnel en gestabiliseerd algoritme voor het oplossen van Toeplitz-stelsels. De orde n hoeft geen macht van 2 te zijn. In onze numerieke voorbeelden laten we n toenemen tot $2^{18} = 262144$.

Outline of the thesis

This thesis is a blend of computational complex analysis and numerical linear algebra. It is divided into four parts.

We start by studying the problem of computing *all* the zeros of an analytic function f that lie inside a positively oriented Jordan curve γ. Our principal means of obtaining information about the location of the zeros is a certain symmetric bilinear form that can be evaluated via numerical integration along γ. This form involves the logarithmic derivative f'/f of f. Our approach could therefore be called a logarithmic residue based quadrature method. It can be seen as a continuation of the pioneering work done by Delves and Lyness. We shed new light on their approach by considering a different set of unknowns and by using the theory of formal orthogonal polynomials. Our algorithm computes not only approximations for the zeros but also their respective multiplicities. It does not require initial approximations for the zeros and we have found that it gives accurate results. The algorithm proceeds by solving generalized eigenvalue problems and a Vandermonde system. A Fortran 90 implementation is available (the package ZEAL). We also present an approach that uses only f and not its first derivative f'. These results are presented in Chapter 1.

In Chapter 2 we focus on the problem of locating clusters of zeros of analytic functions. We show how the approach presented in Chapter 1 can be used to compute approximations for the centre of a cluster and the total number of zeros in this cluster. We also attack the problem of computing all the zeros of f that lie inside γ in an entirely different way, based on rational interpolation at roots of unity. We show how the new approach complements the previous one and how it can be used effectively in case γ is the unit circle.

In Chapter 3 we show how our logarithmic residue based approach can be used to compute all the zeros and poles of a meromorphic function that lie in the interior of a Jordan curve.

In Chapter 4 we consider systems of analytic equations. A multidimensional logarithmic residue formula is available in the literature. This formula involves the integral of a differential form. We transform it into a sum of Riemann integrals and show how the zeros and their respective multiplicities can be computed from these integrals by solving a generalized eigenvalue problem that has Hankel structure and by solving several Vandermonde systems.

This concludes Part 1. In Part 2 we consider analytic functions whose zeros are known to be simple, in particular certain Bessel functions. In Chapter 5 we focus on the Bessel functions $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ and $H_\nu^{(2)}(z)$, and their first derivative, in case the argument $z \in \mathbb{C} \setminus (-\infty, 0]$ and the order $\nu \in \mathbb{R}$. We present a software

package, called ZEBEC, for computing all the zeros of one of these functions that lie inside a rectangle whose edges are parallel to the coordinate axes. In Chapter 6 we consider the functions $J_n(z) \pm i J_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z) J_{n+2}(z)$ in case $n \in \mathbb{N}$. The zeros of these functions play an important role in certain physical applications. We prove that all the zeros that lie in $\mathbb{C}_0$ are simple and show how ZEBEC can be easily extended to compute these zeros.

In Part 3 we focus on Newton's method. In Chapter 7 we propose a modification of Newton's method for computing multiple zeros of analytic mappings (in other words, multiple roots of systems of analytic equations). Under mild assumptions our iteration converges quadratically. It involves certain constants whose product is a lower bound for the multiplicity of the zero. As these constants are usually not known in advance, we devise an iteration in which not only an approximation for the zero is refined, but also approximations for these constants.

In Part 4 we consider problems of structured numerical linear algebra. We have already encountered rational interpolation at the end of Chapter 2 where it was used to locate clusters of zeros of analytic functions. In Chapter 8 we consider rational interpolation in much more detail. We present stabilized fast and superfast algorithms for rational interpolation at roots of unity. In Chapter 9 we use these algorithms to construct stabilized fast as well as superfast solvers for (indefinite) linear systems of equations that have Hankel or Toeplitz structure. Such linear systems occur in many applications, for example in signal processing or Markov chains. They also play a central role in the theory of orthogonal polynomials and Padé approximation. A Fortran 90 implementation of our algorithms is available.

Our work is a mixture of theoretical results (some of which are quite technical, e.g., the results on systems of analytic equations, rational interpolation or Newton's method), numerical analysis and algorithmic aspects, implementation heuristics, and polished software that is publicly available (ZEAL, ZEBEC and our packages for solving Hankel or Toeplitz systems).

Part 1

CHAPTER 1

Zeros of analytic functions

In this chapter we will consider the problem of computing *all* the zeros of an analytic function f that lie in the interior of a Jordan curve γ. The algorithm that we will present computes not only approximations for the zeros but also their respective multiplicities. It doesn't require initial approximations for the zeros and gives accurate results. The algorithm is based on the theory of formal orthogonal polynomials. Its principal means of obtaining information about the location of the zeros is a certain symmetric bilinear form that can be evaluated via numerical integration along γ. This form involves the logarithmic derivative f'/f of f. Our approach could therefore be called a *logarithmic residue based quadrature method*. In the next chapters we will see how it can be used to locate clusters of zeros of analytic functions, to compute all the zeros and poles of a meromorphic function that lie in the interior of a Jordan curve, and to solve systems of analytic equations.

Part of this chapter is contained in our papers [**186**], [**188**], [**190**] and [**192**].

1. Introduction

Let W be a simply connected region in $\mathbb{C}$, $f : W \to \mathbb{C}$ analytic in W and γ a positively oriented Jordan curve in W that does not pass through any zero of f. We consider the problem of computing *all* the zeros of f that lie in the interior of γ, together with their respective multiplicities.

Our approach to this problem can be seen as a continuation of the pioneering work by Delves and Lyness [**70**]. Let N denote the total number of zeros of f that lie in the interior of γ, i.e., the number of zeros where each zero is counted according to its multiplicity. Suppose from now on that $N > 0$. Delves and Lyness considered the sequence $Z_1, \ldots, Z_N$ that consists of all the zeros of f that lie inside γ. Each zero is repeated according to its multiplicity. An easy calculation shows that the logarithmic derivative f'/f has a simple pole at each zero of f with residue equal to the multiplicity of the zero. Cauchy's Theorem implies that

$$N = \frac{1}{2\pi i} \int_\gamma \frac{f'(z)}{f(z)}\, dz. \tag{12}$$

This formula enables us to calculate N via numerical integration. Methods for the determination of zeros of analytic functions that are based on the numerical evaluation of integrals are called *quadrature methods*. A review of such methods was given by Ioakimidis [**158**]. Delves and Lyness considered the integrals

$$s_p := \frac{1}{2\pi i} \int_\gamma z^p \frac{f'(z)}{f(z)}\, dz, \qquad p = 0, 1, 2, \ldots .$$

The residue theorem implies that the s_p's are equal to the *Newton sums* of the unknown zeros,

$$s_p = Z_1^p + \cdots + Z_N^p, \qquad p = 0, 1, 2, \ldots . \tag{13}$$

In what follows we will assume that all the s_p's that are needed have been calculated. In particular, we will assume that the value of $N = s_0$ is known.

Delves and Lyness considered the monic polynomial of degree N that has zeros $Z_1, \ldots, Z_N$,

$$P_N(z) := \prod_{k=1}^{N} (z - Z_k) =: z^N + \sigma_1 z^{N-1} + \cdots + \sigma_N.$$

They called $P_N(z)$ the *associated polynomial* for the interior of γ. Its coefficients can be calculated via Newton's identities.

THEOREM 1 (Newton's identities).

$$\begin{aligned} &s_1 + \sigma_1 = 0 \\ &s_2 + s_1 \sigma_1 + 2\sigma_2 = 0 \\ &\qquad\vdots \\ &s_N + s_{N-1}\sigma_1 + \cdots + s_1 \sigma_{N-1} + N\sigma_N = 0. \end{aligned}$$

PROOF. An elegant proof was given by Carpentier and Dos Santos [**54**]. □

In this way they reduced the problem to the easier problem of computing the zeros of a polynomial. Unfortunately, the map from the Newton sums $s_1, \ldots, s_N$ to the coefficients $\sigma_1, \ldots, \sigma_N$ is usually ill-conditioned. Also, the polynomials that arise in practice may be such that small changes in the coefficients produce much larger changes in some of the zeros. This ill-conditioning of the map between the coefficients of a polynomial and its zeros was investigated by Wilkinson [**300**]. The location of the zeros determines their sensitivity to perturbations of the coefficients. Multiple zeros and very close zeros are extremely sensitive, but even a succession of moderately close zeros can result in severe ill-conditioning. Wilkinson states that ill-conditioning in polynomials cannot be overcome without, at some stage of the computation, resorting to high precision arithmetic.

If f has many zeros in the interior of γ, then the associated polynomial is of high degree and could be very ill-conditioned. Therefore, if N is large, one has to calculate the coefficients $\sigma_1, \ldots, \sigma_N$, and thus the integrals $s_1, \ldots, s_N$, very accurately. To avoid the use of high precision arithmetic and to reduce the number of integrand evaluations needed to approximate the s_p's, Delves and Lyness suggested to construct and solve the associated polynomial only if its degree is smaller than or equal to a preassigned number M. Otherwise, the interior of γ is subdivided or covered with a finite covering and the smaller regions are treated in turn. The choice of M involves a trade-off. If M is increased, then fewer regions have to be scanned. However, if M is chosen too large, then the resulting associated polynomial may be ill-conditioned. Delves and Lyness chose $M = 5$.

Botten, Craig and McPhedran [**41**] made a Fortran 77 implementation of the method of Delves and Lyness.

In some applications, the calculation of the derivative f' is more time-consuming than that of f. Delves and Lyness used an integration by parts to derive a formula for s_p that depends only on a multi-valued logarithm of f and not on f'. To apply this formula, they had to keep track of the sheet on which $\log f(z)$ lies as z runs along the curve γ. Unfortunately, in most cases it is impossible to do this in a completely reliable way, i.e., without accidentally overlooking any sheets. Carpentier and Dos Santos [**54**] and Davies [**63**] derived similar formulae. See also Ioakimidis and Anastasselou [**161**].

Instead of using Newton's identities to construct the associated polynomial, Li [**199**] considered (13) as a system of polynomial equations. He used a homotopy continuation method to solve this system.

What is wrong with these approaches, in our opinion, is that they consider the wrong set of unknowns. One should consider the mutually distinct zeros and their respective multiplicities *separately*. This is the approach that we will follow. Let n denote the number of mutually distinct zeros of f that lie inside γ. Let $z_1, \ldots, z_n$ be these zeros and $\nu_1, \ldots, \nu_n$ their respective multiplicities. The quadrature method that we will present generalizes the approach of Delves and Lyness. We will show how the mutually distinct zeros can be calculated by solving generalized eigenvalue problems. The value of n will be determined indirectly. Once n and $z_1, \ldots, z_n$ have been found, the problem becomes linear and the multiplicities $\nu_1, \ldots, \nu_n$ can be computed by solving a Vandermonde system.

The rest of this chapter is organized as follows. In the remainder of this section we will discuss how the total number of zeros can be calculated with certainty and we will give an overview of other approaches that were proposed for computing zeros of analytic functions. In Section 2 we will tackle our problem by using the theory of formal orthogonal polynomials. This section is devoted to theoretical considerations whereas our numerical algorithm will be presented in Section 3. We have found that this algorithm gives very accurate results. In Section 4 we will give numerical examples computed via a Matlab implementation whereas in Section 5 we will present a Fortran 90 implementation (the package ZEAL) and we will give more numerical examples. In Section 6 we will present a derivative-free approach that involves $1/f$ instead of f'/f and we will compare both approaches.

1.1. Computing the total number of zeros with certainty. The total number of zeros of f that lie inside γ is given by the integral

$$N = \frac{1}{2\pi i} \int_\gamma \frac{f'(z)}{f(z)}\, dz, \tag{14}$$

cf. Equation (12). By making the substitution $w := f(z)$ we obtain that

$$N = \frac{1}{2\pi i} \int_{f(\gamma)} \frac{1}{w}\, dw. \tag{15}$$

Here $f(\gamma)$ denotes the image of the curve γ under f. This is a closed curve that avoids the origin. The winding number of $f(\gamma)$ with respect to the origin is defined as the increase in the argument of $f(z)$ along γ divided by 2π,

$$n\big(f(\gamma), 0\big) := \frac{1}{2\pi} \big[\arg f(z) \big]_{z \in \gamma}.$$

Informally speaking, one can say that it is equal to the number of times that the curve $f(\gamma)$ "winds" itself around the origin. A classical theorem in complex analysis (see, e.g., Henrici [**141**, p. 233]) says that this winding number can be expressed as the integral that appears in the right-hand side of (15). Hence $N = n\big(f(\gamma), 0\big)$. This result is known as the "principle of the argument."

EXAMPLE 1. Suppose that $f(z) = e^{3z} + 20z\cos z - 1$. Figure 1 shows the curve $f(\gamma)$ in case $\gamma = \{z \in \mathbb{C} : |z| = 2\}$. Clearly $N = 4$.

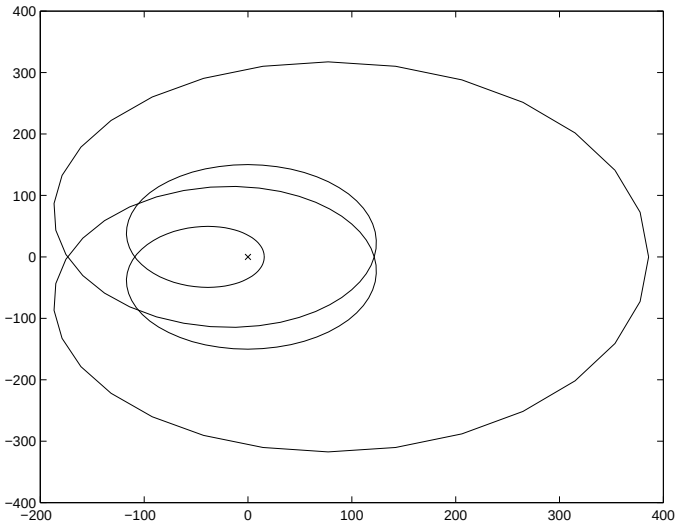


FIGURE 1. The curve $f(\gamma)$ where $f(z) = e^{3z} + 20z\cos z - 1$ and $\gamma = \{z \in \mathbb{C} : |z| = 2\}$.

If $\gamma = \{z \in \mathbb{C} : |z - 1| = 1/2\}$, then $N = 0$, as can be seen from Figure 2. ♢

Earlier in this chapter we have suggested to calculate the total number of zeros N via numerical integration, i.e., by using (14). An alternative approach can be based on an algorithm for computing winding numbers. The range of the function arg is $(-\pi, \pi]$. If the increase in argument along the straight section

$$[\alpha, \beta] := \{\, z \in \mathbb{C} : z = t\alpha + (1-t)\beta, \quad 0 \le t \le 1 \,\}, \qquad \alpha, \beta \in \mathbb{C},$$

satisfies

$$\big|\big[\arg f(z)\big]_{z\in[\alpha,\beta]}\big| \le \pi,$$

then

$$\big[\arg f(z)\big]_{z\in[\alpha,\beta]} = \arg\Big(\frac{f(\beta)}{f(\alpha)}\Big),$$

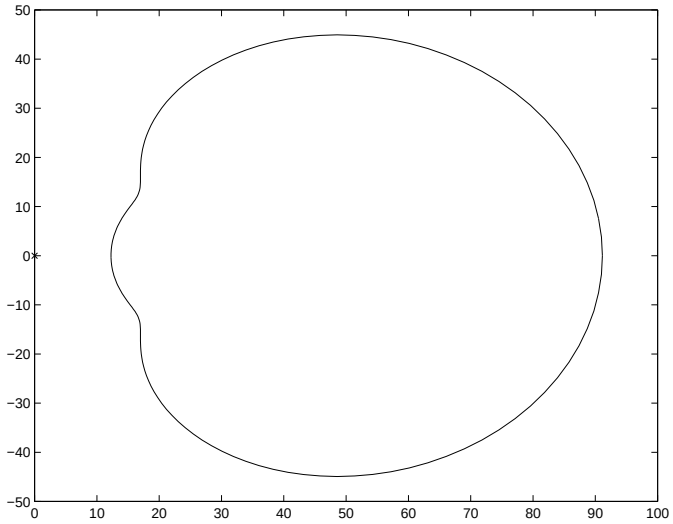


FIGURE 2. The curve $f(\gamma)$ where $f(z) = e^{3z} + 20z\cos z - 1$ and $\gamma = \{z \in \mathbb{C} : |z-1| = 1/2\}$.

as the reader may easily verify. Let us discretize the curve γ into the sequence of points $c_1, \ldots, c_G$. Define $c_{G+1} := c_1$. Then it follows that

$$N = \frac{1}{2\pi} \sum_{k=1}^{G} \arg\Big(\frac{f(c_{k+1})}{f(c_k)}\Big)$$

if

$$\Big|\big[\arg f(z)\big]_{z\in[c_k,c_{k+1}]}\Big| \leq \pi \tag{16}$$

for $k = 1, \ldots, G$. In other words, if condition (16) is satisfied, then N can be computed simply by evaluating f at the points $c_1, \ldots, c_G$. These considerations form the basis for Henrici's algorithm [**141**, pp. 239–241]. See also Ying and Katz [**307**].

Unfortunately, condition (16) may not be easy to verify for an arbitrary analytic function f. If the discretization of γ is inadequate, then the computed value of $n\big(f(\gamma), 0\big)$ and hence N may be wrong. In this sense Henrici's algorithm is unreliable and the same holds for numerical integration. Indeed, a finite number of functional or derivative values are not enough to determine the number of zeros of f, even if f is a polynomial. This was shown by Ying in his PhD thesis [**306**]. The next theorem is a slightly modified version of his result.

THEOREM 2. *Let Ω be a simply connected region in $\mathbb{C}$. Let $l \geq 1$ be a positive integer and let $\zeta_1, \ldots, \zeta_l \in \Omega$. Let $m_1, \ldots, m_l \geq 0$ be nonnegative integers and suppose that numerical values are given for*

$$p(\zeta_1), p'(\zeta_1), \ldots, p^{(m_1)}(\zeta_1), \ \ldots \ , p(\zeta_l), p'(\zeta_l), \ldots, p^{(m_l)}(\zeta_l) \tag{17}$$

where p is only known to be a polynomial in Ω that is not identically zero in Ω. Then the following holds:

1. *The information in* (17) *is insufficient to determine the location of all the zeros of* p *that lie in* Ω.

Let Γ *be a Jordan curve in* Ω *that does not pass through any zero of* p.

2. *Suppose that* N, *the total number of zeros of* p *that lie inside* Γ, *is known. Even then, the information in* (17) *is insufficient to determine the location of all the zeros of* p *that lie inside* Γ, *unless* $N = 0$ *or* $N > 0$ *and these zeros coincide with a subset of* $\{\zeta_1, \ldots, \zeta_l\}$.
3. *If the points* $\zeta_1, \ldots, \zeta_l$ *do not belong to the closure of the interior of* Γ, *then the information in* (17) *is insufficient to estimate the location of all the zeros of* p *that lie inside* Γ.

PROOF.

1. Define $h(z)$ as $(z-\zeta_1)^{m_1+1} \cdot \cdots \cdot (z-\zeta_l)^{m_l+1}$. Choose ζ_0 in $\Omega \setminus \{\zeta_1, \ldots, \zeta_l\}$ such that $p(\zeta_0) \neq 0$ and let
$$p_1(z) := p(z) - \frac{p(\zeta_0)}{h(\zeta_0)}\, h(z).$$
Then p_1 is a polynomial in Ω and $p_1^{(k)}(\zeta_i) = p^{(k)}(\zeta_i)$ for $i = 1, \ldots, l$ and $k = 0, 1, \ldots, m_i$. However, $p_1(\zeta_0) = 0$ while $p(\zeta_0) \neq 0$.
2. Define
$$\epsilon := \inf_{z \in \Gamma} |p(z)| \qquad \text{and} \qquad M := \sup_{z \in \Gamma} |h(z)|.$$
Then $\epsilon > 0$ and $M > 0$. Let
$$p_2(z) := p(z) + \frac{\epsilon}{2M}\, h(z).$$
Then p_2 is a polynomial in Ω and $p_2^{(k)}(\zeta_i) = p^{(k)}(\zeta_i)$ for $i = 1, \ldots, l$ and $k = 0, 1, \ldots, m_i$. Since $|p(z)| > |\frac{\epsilon}{2M}\, h(z)|$ for all $z \in \Gamma$, Rouché's Theorem asserts that p_2 and p have the same total number of zeros in the interior of Γ. Let ζ be a zero of p that lies in the interior of Γ. If $\zeta \notin \{\zeta_1, \ldots, \zeta_l\}$, then $p_2(\zeta) \neq 0$.
3. Define
$$\delta := \inf_{z \in \Gamma} |h(z)| \qquad \text{and} \qquad Q := \sup_{z \in \Gamma} |p(z)|.$$
Then $\delta > 0$ and $Q > 0$. Let
$$p_3(z) := p(z) + \frac{2Q}{\delta}\, h(z).$$
Then p_3 is a polynomial in Ω and $p_3^{(k)}(\zeta_i) = p^{(k)}(\zeta_i)$ for $i = 1, \ldots, l$ and $k = 0, 1, \ldots, m_i$. Since $|\frac{2Q}{\delta}\, h(z)| > |p(z)|$ for all $z \in \Gamma$, Rouché's Theorem asserts that p_3 and h have the same total number of zeros in the interior of Γ. Therefore, p_3 has no zeros in the interior of Γ.

This proves the theorem. □

There exist several reliable approaches for computing N. However, they all assume that some kind of *global* information is available, which may not be the case in practice and hence these algorithms are in fact of little use.

- Ying and Katz [**307**] developed a reliable variant of Henrici's algorithm. They assume that an upper bound for $|f''(z)|$ along an arbitrary line segment is available.
- Herlocker and Ely [**142**] experimented with a numerical integration approach based on Simpson's rule and the corresponding formula for the integration error. This formula involves the fourth derivative of the integrand evaluated at an unknown point in the integration interval. Automatic differentiation combined with interval arithmetic enabled them to bound the integration error.
- The total number of zeros can also be computed as the topological degree of the mapping

$$F(x,y) := \big(\operatorname{Re} f(x+iy), \operatorname{Im} f(x+iy)\big)$$

with respect to the interior of γ (interpreted as a subset of $\mathbb{R}^2$) and the point $(0,0)$. (We will not go into the details of degree theory. For an excellent introduction, we refer the interested reader to Lloyd's book [**201**].) Boult and Sikorski [**42**] considered the case that γ is the boundary of the unit square $[0,1] \times [0,1]$. They proved that, in case F satisfies the Lipschitz condition with constant $K > 0$ and if the infinity norm of F on γ is at least $d > 0$ where $K/(4d) \geq 1$, then at least $4\lfloor K/(4d)\rfloor$ function evaluations are needed to compute the topological degree. See also Traub, Wasilkowski and Woźniakowski [**269**, pp. 193–194].

As already mentioned, these reliable algorithms can seldom be used in practice since global information such as an upper bound for the modulus of a higher derivative of f or the Lipschitz constant of $F = (\operatorname{Re} f, \operatorname{Im} f)$ or a lower bound for the infinity norm of F on γ is usually not available. Therefore, although they may indeed give incorrect results, Henrici's algorithm and numerical integration are the only approaches that we can use to compute N. The output of Henrici's algorithm is always an integer but unfortunately one has no idea whether it's the correct integer or not. By using quadrature formulae with different degrees of accuracy, one can easily obtain an estimate of the quadrature error. Also, the size of the imaginary part of the computed approximation for N as well as the distance of the real part to the nearest integer give a clear indication of the error. (Of course, as the integral is known to be an integer, an approximation that has an error that is less than 0.5 is sufficient.) For these reasons, we prefer to use numerical integration instead of Henrici's algorithm.

1.2. An overview of other approaches. Let us conclude this section by giving a brief overview of other methods that were proposed for computing zeros of analytic functions.

Suppose that the zeros of f that lie inside γ are known to be simple. Then f can be written as

$$f(z) = \phi(z) \prod_{k=1}^{n} (z - z_k) \tag{18}$$

where $\phi : W \to \mathbb{C}$ is analytic in W and does not vanish inside γ. Let $\alpha \in \mathbb{C}$ be an arbitrary point inside γ such that $f(\alpha) \neq 0$. Then in the interior of γ the function ϕ can be written as

$$\phi(z) = \exp \psi(z), \qquad z \in \operatorname{int} \gamma, \tag{19}$$

where ψ is defined by

$$\psi(z) := \frac{1}{2\pi\, i} \int_\gamma \frac{\log[(t-\alpha)^{-n} f(t)]}{t-z}\, dt, \qquad z \in \operatorname{int} \gamma, \tag{20}$$

cf. Smirnov [**256**]. The function ψ is analytic inside γ. Note that the logarithm in (20) is well defined as the winding number of $(z-\alpha)^{-n} f(z)$ with respect to γ and the origin is equal to zero.

Starting from (18) we find that

$$z_k = z - \frac{f(z)}{\phi(z) \prod_{j=1, j\neq k}^{n} (z - z_j)}$$

for $k = 1, \ldots, n$. Assume that distinct complex numbers $\zeta_1, \ldots, \zeta_n$ are reasonably good approximations for the zeros $z_1, \ldots, z_n$ of f. Putting $z = \zeta_k$ and substituting the zeros z_j by their approximations ζ_j $(j \neq k)$ we obtain

$$\hat{\zeta}_k := \zeta_k - \frac{f(\zeta_k)}{\phi(\zeta_k) \prod_{j=1, j\neq k}^{n} (\zeta_k - \zeta_j)} \tag{21}$$

for $k = 1, \ldots, n$. The point $\hat{\zeta}_k$ appears to be a new approximation for the zero z_k. Petković, Carstensen and Trajković [**233**] presented a simultaneous iterative method that is based on (21). They proved that the iteration converges quadratically if the initial approximations $\zeta_1^{(0)}, \ldots, \zeta_n^{(0)}$ are sufficiently close to the zeros $z_1, \ldots, z_n$ of f. The function ϕ is evaluated via numerical integration.

Since (18) and (19) imply that

$$\frac{f'(z)}{f(z)} = \psi'(z) + \sum_{k=1}^{n} \frac{1}{z - z_k}, \qquad z \notin \{z_1, \ldots, z_n\},$$

it follows that

$$z_k = z - \left[\frac{f'(z)}{f(z)} - \psi'(z) - \sum_{j=1, j\neq k}^{n} \frac{1}{z - z_j} \right]^{-1}, \qquad z \notin \{z_1, \ldots, z_n\},$$

for $k = 1, \ldots, n$. Putting again $z = \zeta_k$ and substituting the zeros z_j by their approximations ζ_j $(j \neq k)$ we obtain

$$\hat{\zeta}_k := \zeta_k - \left[\frac{f'(\zeta_k)}{f(\zeta_k)} - \psi'(\zeta_k) - \sum_{j=1, j\neq k}^{n} \frac{1}{\zeta_k - \zeta_j} \right]^{-1} \tag{22}$$

for $k = 1, \ldots, n$. Petković and Herceg [**234**] analysed the corresponding iterative method and proved that it has cubic convergence. They also presented a version of

the algorithm that uses circular interval arithmetic (see also [**232**]). The derivative ψ' is given by

$$\psi'(z) = \frac{1}{2\pi i} \int_\gamma \frac{\log[(t-\alpha)^{-n} f(t)]}{(t-z)^2}\, dt, \qquad z \in \operatorname{int} \gamma,$$

and has to be evaluated via numerical integration.

Petković and Marjanović [**235**] generalized these results and gave simultaneous iterative methods that have order of convergence $p+2$ ($p = 1, 2, \dots$) if p denotes the order of the highest derivative of f that appears in the iteration formula. These algorithms can be used only if the zeros of f are known to be simple and if sufficiently accurate initial approximations are available. Atanassova [**16**] considered the case of multiple zeros but assumed that the multiplicities are known in advance.

In case f is a polynomial, the iteration that corresponds to (21) is called the Durand-Kerner (or also Weierstrass) method whereas the iteration that corresponds to (22) is known as Aberth's method (cf. Bini [**32**]). The problem of calculating zeros of polynomials received a lot of interest in the past and is still a lively area of research. McNamee [**208, 209**] compiled an extensive bibliography on computing zeros of polynomials. Although polynomials are of course a special case of analytic functions, we will not give a comprehensive overview of all the methods that were proposed. Instead we refer the interested reader to the papers by Bini and Pan [**36**], Cardinal [**52**], Carstensen and Sakurai [**55**] and the survey paper by Pan [**229**] as well as the references cited therein. Recently, Bini, Gemignani and Meini [**33**] have presented an interesting method to compute a factor of a polynomial or of an analytic function that is given as a power series. Their approach is based on a matrix version of Koenig's Theorem and on cyclic reduction. It involves infinite Toeplitz matrices in block Hessenberg form.

In a number of papers and short notes, Anastasselou and Ioakimidis [**7, 8, 9, 10, 11, 159, 160, 162, 163**] considered the problem of computing zeros of sectionally analytic functions (i.e., functions that are analytic except for a finite number of discontinuity arcs). They proposed variations and generalizations of the method of Burniston and Siewert [**48**], which is based on the theory of Riemann-Hilbert boundary value problems (cf. Gakhov [**87**]). The authors focused on the function $\alpha + \beta z - z \tanh^{-1}(1/z)$ where $\alpha, \beta \in \mathbb{C}$ are parameters. This function appears in the theory of ferromagnetism. It has the discontinuity interval $[-1, 1]$. Already for this example, the approach of Anastasselou and Ioakimidis requires a lot of specific analytical calculations. Therefore, it is unclear how their method could lead to a 'black box' algorithm that can handle arbitrary functions. Also, although multiple zeros are not a problem, their approach cannot calculate multiplicities.

Yakoubsohn [**305**] proposed an exclusion method for computing zeros of analytic functions. Unfortunately, his exclusion function is difficult to evaluate and multiple zeros require special treatment. He applied his algorithm only to polynomials. See also Ying and Katz [**308**].

2. Formal orthogonal polynomials

Let $\mathcal{P}$ be the linear space of polynomials with complex coefficients. We define a symmetric bilinear form

$$\langle \cdot, \cdot \rangle \; : \; \mathcal{P} \times \mathcal{P} \to \mathbb{C}$$

by setting

$$\langle \phi, \psi \rangle := \frac{1}{2\pi\, i} \int_\gamma \phi(z)\psi(z)\frac{f'(z)}{f(z)}\, dz = \sum_{k=1}^{n} \nu_k \phi(z_k)\psi(z_k) \tag{23}$$

for any two polynomials $\phi, \psi \in \mathcal{P}$. The latter equality follows from the fact that f'/f has a simple pole at z_k with residue ν_k for $k = 1, \ldots, n$. Note that $\langle \cdot, \cdot \rangle$ can be evaluated via numerical integration along γ. In what follows, we will assume that all the "inner products" $\langle \phi, \psi \rangle$ that are needed have been calculated. Let $s_p := \langle 1, z^p \rangle$ for $p = 0, 1, 2, \ldots$. These ordinary moments are equal to the *Newton sums* of the unknown zeros,

$$s_p = \sum_{k=1}^{n} \nu_k z_k^p, \qquad p = 0, 1, 2, \ldots .$$

In particular, $s_0 = \nu_1 + \cdots + \nu_n = N$, the total number of zeros. Hence, we may assume that the value of N is known. Let H_k be the $k \times k$ Hankel matrix

$$H_k := \left[\; s_{p+q} \; \right]_{p,q=0}^{k-1} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{k-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{k-1} & \cdots & \cdots & s_{2k-2} \end{bmatrix}$$

for $k = 1, 2, \ldots$. Let H be the infinite Hankel matrix

$$H := \left[\; s_{p+q} \; \right]_{p,q \geq 0} .$$

Observe that the form $\langle \cdot, \cdot \rangle$ is completely determined by the sequence of moments $(s_p)_{p \geq 0}$.

A monic polynomial φ_t of degree $t \geq 0$ that satisfies

$$\langle z^k, \varphi_t(z) \rangle = 0, \qquad k = 0, 1, \ldots, t-1, \tag{24}$$

is called a *formal orthogonal polynomial* (FOP) of degree t. (Observe that condition (24) is void for $t = 0$.) The adjective *formal* emphasizes the fact that, in general, the form $\langle \cdot, \cdot \rangle$ does not define a true inner product. An important consequence of this fact is that, in contrast to polynomials that are orthogonal with respect to a true inner product, formal orthogonal polynomials need not exist or need not be unique for every degree. (For details, see for example Draux [**71, 72**], Gutknecht [**128, 130**] or Gragg and Gutknecht [**113**].) If (24) is satisfied and φ_t is unique, then φ_t is called a *regular* FOP and t a *regular index*. If we set

$$\varphi_t(z) =: u_{0,t} + u_{1,t} z + \cdots + u_{t-1,t} z^{t-1} + z^t$$

then condition (24) translates into the Yule-Walker system

$$\begin{bmatrix} s_0 & s_1 & \cdots & s_{t-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{t-1} & \cdots & \cdots & s_{2t-2} \end{bmatrix} \begin{bmatrix} u_{0,t} \\ u_{1,t} \\ \vdots \\ u_{t-1,t} \end{bmatrix} = - \begin{bmatrix} s_t \\ s_{t+1} \\ \vdots \\ s_{2t-1} \end{bmatrix}. \tag{25}$$

Hence, the regular FOP of degree $t \geq 1$ exists if and only if the matrix H_t is nonsingular. Thus, the rank profile of H determines which regular FOPs exist. If t is a regular index, then

$$\varphi_t(z) = \frac{1}{\det H_t} \begin{vmatrix} s_0 & s_1 & \cdots & s_{t-1} & 1 \\ s_1 & & \iddots & \vdots & z \\ \vdots & \iddots & & \vdots & \vdots \\ s_{t-1} & \cdots & \cdots & s_{2t-2} & z^{t-1} \\ s_t & \cdots & \cdots & s_{2t-1} & z^t \end{vmatrix},$$

as one can easily verify. Note that this determinant expression implies that

$$\langle \varphi_t, \varphi_t \rangle = \frac{\det H_{t+1}}{\det H_t}. \tag{26}$$

The following theorem characterizes n, the number of mutually distinct zeros. It enables us, theoretically at least, to calculate n as $\operatorname{rank} H_N$.

THEOREM 3. *$n = \operatorname{rank} H_{n+p}$ for every nonnegative integer p. In particular, $n = \operatorname{rank} H_N$.*

PROOF. Let p be a nonnegative integer. The matrix H_{n+p} can be written as

$$\begin{aligned} H_{n+p} &= \sum_{k=1}^{n} \nu_k \begin{bmatrix} 1 & \cdots & z_k^{n+p-1} \\ \vdots & & \vdots \\ z_k^{n+p-1} & \cdots & z_k^{2(n+p)-2} \end{bmatrix} \\ &= \sum_{k=1}^{n} \nu_k \begin{bmatrix} 1 \\ \vdots \\ z_k^{n+p-1} \end{bmatrix} \begin{bmatrix} 1 & \cdots & z_k^{n+p-1} \end{bmatrix}. \end{aligned}$$

This implies that $\operatorname{rank} H_{n+p} \leq n$. However, H_n is nonsingular. Indeed, as one can easily verify, the matrix H_n can be factorized as $H_n = V_n D_n V_n^T$ where V_n is the Vandermonde matrix $V_n := [z_s^r]_{r=0,s=1}^{n-1,n}$ and D_n is the diagonal matrix $D_n := \operatorname{diag}(\nu_1, \ldots, \nu_n)$. Therefore $\operatorname{rank} H_{n+p} \geq n$. It follows that $\operatorname{rank} H_{n+p} = n$. □

Therefore H_n is nonsingular whereas H_t is singular for $t > n$. Note that $H_1 = [s_0]$ is nonsingular by assumption. The regular FOP of degree 1 exists and is given by $\varphi_1(z) = z - \mu$ where

$$\mu := \frac{s_1}{s_0} = \frac{\sum_{k=1}^{n} \nu_k z_k}{\sum_{k=1}^{n} \nu_k}$$

is the arithmetic mean of the zeros. Theorem 3 implies that the regular FOP φ_n of degree n exists and tells us also that regular FOPs of degree larger than n do not exist. The polynomial φ_n is easily seen to be

$$\varphi_n(z) = (z - z_1) \cdot \cdots \cdot (z - z_n). \tag{27}$$

It is the monic polynomial of degree n that has $z_1, \ldots, z_n$ as simple zeros. Its coefficients can be calculated by solving an $n \times n$ Yule-Walker system. This polynomial has the peculiar property that it is orthogonal to *all* polynomials (including itself),

$$\langle z^k, \varphi_n(z) \rangle = 0, \qquad k = 0, 1, 2, \ldots . \tag{28}$$

NOTE. Kronecker's Theorem [**230**, p.37] tells us that the infinite Hankel matrix H has finite rank if and only if its *symbol*, which is defined as the formal Laurent series

$$\frac{s_0}{z} + \frac{s_1}{z^2} + \frac{s_2}{z^3} + \cdots ,$$

represents a rational function of z. This is indeed the case. It is easily seen that

$$\sum_{k=1}^{n} \frac{\nu_k}{z - z_k} = \frac{s_0}{z} + \frac{s_1}{z^2} + \frac{s_2}{z^3} + \cdots \qquad \text{near } z = \infty. \tag{29}$$

In system theory the problem of reconstructing the rational function that appears in the left-hand side of (29) from the sequence of moments $(s_p)_{p \geq 0}$ is called a minimal realization problem. In that context the moments are called Markov parameters. For more details and the connection with continued fractions, Padé approximation, the Euclidean algorithm for formal Laurent series and the Berlekamp-Massey algorithm, see for example [**45, 46, 47, 71, 113, 166**]. Note that the left-hand side of (29) is a rational function of type $[n - 1/n]$ and that its denominator polynomial is given by $\varphi_n(z)$.

Once n is known, the mutually distinct zeros $z_1, \ldots, z_n$ can be calculated by solving a generalized eigenvalue problem. Indeed, let $H_n^<$ be the Hankel matrix

$$H_n^< := \begin{bmatrix} s_1 & s_2 & \cdots & s_n \\ s_2 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_n & \cdots & \cdots & s_{2n-1} \end{bmatrix}.$$

THEOREM 4. *The eigenvalues of the pencil $H_n^< - \lambda H_n$ are given by $z_1, \ldots, z_n$.*

PROOF. Suppose that $\varphi_n(z) =: u_{0,n} + u_{1,n} z + \cdots + u_{n-1,n} z^{n-1} + z^n$. Then (27) implies that the zeros $z_1, \ldots, z_n$ are given by the eigenvalues of the companion matrix

$$C_n := \begin{bmatrix} 0 & 0 & \cdots & 0 & -u_{0,n} \\ 1 & 0 & \cdots & 0 & -u_{1,n} \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & -u_{n-2,n} \\ 0 & \cdots & 0 & 1 & -u_{n-1,n} \end{bmatrix}.$$

Let $\lambda^\star$ be an eigenvalue of C_n and x a corresponding eigenvector. As H_n is regular, we may conclude that

$$C_n x = \lambda^\star x \Leftrightarrow H_n C_n x = \lambda^\star H_n x.$$

Using (25) one can easily verify that $H_n C_n = H_n^<$. This proves the theorem.

Another proof goes as follows. As in the proof of Theorem 3, let V_n be the Vandermonde matrix

$$V_n := \begin{bmatrix} 1 & \cdots & 1 \\ z_1 & \cdots & z_n \\ \vdots & & \vdots \\ z_1^{n-1} & \cdots & z_n^{n-1} \end{bmatrix}$$

and let $D_n := \operatorname{diag}(\nu_1, \ldots, \nu_n)$. Also, let $D_n^{(1)} := \operatorname{diag}(\nu_1 z_1, \ldots, \nu_n z_n)$. Then the matrices H_n and $H_n^<$ can be factorized as

$$H_n = V_n D_n V_n^T \quad \text{and} \quad H_n^< = V_n D_n^{(1)} V_n^T.$$

Let $\lambda^\star$ be an eigenvalue of the pencil $H_n^< - \lambda H_n$ and x a corresponding eigenvector. Then

$$\begin{aligned} & H_n^< x = \lambda^\star H_n x \\ \Leftrightarrow \quad & V_n D_n^{(1)} V_n^T x = \lambda^\star V_n D_n V_n^T x \\ \Leftrightarrow \quad & D_n^{(1)} y = \lambda^\star D_n y \quad \text{if } y := V_n^T x \\ \Leftrightarrow \quad & \operatorname{diag}(z_1, \ldots, z_n) y = \lambda^\star y. \end{aligned}$$

This proves the theorem. □

NOTE. Recently, Golub, Milanfar and Varah [**111**] have presented a stable numerical solution to the problem of reconstructing a polygonal shape from moments. This problem has many applications including tomographic reconstruction and geophysical inversion. In the latter application, it is of interest to reconstruct the shape and (possibly) density of a gravitational anomaly from discrete measurements of the exterior gravitational field at spatially separated points. The authors' approach is based on the matrix pencil that appears in Theorem 4. To solve the generalized eigenvalue problem, they have developed an efficient variant of the QZ algorithm that takes into account the way the matrices H_n and $H_n^<$ are related.

Once $z_1, \ldots, z_n$ have been found, the multiplicities $\nu_1, \ldots, \nu_n$ can be computed by solving the Vandermonde system

$$\begin{bmatrix} 1 & \cdots & 1 \\ z_1 & \cdots & z_n \\ \vdots & & \vdots \\ z_1^{n-1} & \cdots & z_n^{n-1} \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{bmatrix}. \tag{30}$$

This can be done via the algorithm of Gohberg and Koltracht [**108**]. This algorithm takes full account of the structure of a Vandermonde matrix and is not only faster but also more accurate than general purpose algorithms such as Gaussian elimination with partial pivoting. It has arithmetic complexity $\mathcal{O}(n^2)$.

NOTE. Vandermonde matrices are often very ill-conditioned. Gautschi wrote many papers on this subject, see for example [**61, 91, 92, 94, 95, 98, 99**]. In our case, however, the components of the solution vector of (30) are known to be integers, and therefore there is no problem, even if the linear system (30) happens to be ill-conditioned, as long as the computed approximations for the components of the solution vector have an absolute error that is less than 0.5. We remark that Hankel matrices also have the reputation of being ill-conditioned [**271**].

Theorem 3 and 4 suggest the following approach to compute n and $z_1, \ldots, z_n$. Start by computing the total number of zeros N. Next, compute $s_1, \ldots, s_{2N-2}$. As already mentioned, this can be done via numerical integration along γ. The number of mutually distinct zeros is then calculated as the rank of H_N, $n = \operatorname{rank} H_N$. Finally, the zeros $z_1, \ldots, z_n$ are obtained by solving a generalized eigenvalue problem. Unfortunately, this approach has several disadvantages:

- Theoretically the $N - n$ smallest singular values of H_N are equal to zero. In practice, this will not be the case, and it may be difficult to determine the rank of H_N and hence the value of n in case the gap between the computed approximations for the zero singular values and the nonzero singular values is too small.
- The approximations for $z_1, \ldots, z_n$ obtained via Theorem 4 may not be very accurate. Indeed, the mapping from the Newton sums to the zeros and their respective multiplicities,

$$(s_0, s_1, \ldots, s_{2n-1}) \mapsto (z_1, \ldots, z_n, \nu_1, \ldots, \nu_n), \tag{31}$$

  is usually very ill-conditioned. (See for example the papers by Gautschi [**93, 96, 97**] who studied the conditioning of (31) in the context of Gauss quadrature formulae. For a recent paper on this subject, we refer to Beckermann and Bourreau [**19**].) Indeed, a classical adage in numerical analysis says that one should avoid the use of ordinary moments.

In Section 3 we will present an algorithm that gives more accurate approximations for $z_1, \ldots, z_n$. The idea is the following. The inner products that appear in the Hankel matrices H_n and $H_n^<$ are related to the standard monomial basis. Why not consider a different basis? In other words, let us try to use modified moments instead of ordinary moments. The fact that

$$H_n = \left[\langle z^p, z^q \rangle\right]_{p,q=0}^{n-1} \quad \text{and} \quad H_n^< = \left[\langle z^p, z z^q \rangle\right]_{p,q=0}^{n-1}$$

suggests that we should consider the matrices

$$\left[\langle \psi_p, \psi_q \rangle\right]_{p,q=0}^{n-1} \quad \text{and} \quad \left[\langle \psi_p, \psi_1 \psi_q \rangle\right]_{p,q=0}^{n-1} \tag{32}$$

where ψ_k is a polynomial of degree k for $k = 0, 1, \ldots, n-1$. Of course, even if we succeed in writing Theorem 4 in terms of (32), the question remains which polynomials ψ_k to choose. We have found that very accurate results are obtained if we use the formal orthogonal polynomials. In other words, the zeros of $\varphi_n(z)$ will be computed from inner products that involve $\varphi_0(z), \varphi_1(z), \ldots, \varphi_{n-1}(z)$. The value of n will be determined indirectly.

All this will be explained in more detail in Section 3. Let us conclude this section by discussing the orthogonality properties of FOPs. This will enable us to define the matrices (32) and to examine their structure.

If H_n is strongly nonsingular, i.e., if all its leading principal submatrices are nonsingular, then we have a full set $\{\varphi_0, \varphi_1, \ldots, \varphi_n\}$ of regular FOPs.

What happens if H_n is not strongly nonsingular and thus there is no full set of regular FOPs? Let $\{k_j\}_{j=0}^{J}$ be the set of all regular indices, with

$$k_0 < k_1 < \cdots < k_J.$$

Then $k_0 = 0$, $k_1 = 1$ and $k_J = n$. By filling up the gaps in the sequence of existing regular FOPs it is possible to define a sequence $\{\varphi_t\}_{t=0}^{\infty}$, with φ_t a monic polynomial of degree t, such that if these polynomials are grouped into blocks according to the sequence of regular indices, then polynomials belonging to different blocks are orthogonal with respect to (23). More precisely, define $\{\varphi_t\}_{t=0}^{\infty}$ as follows. If t is a regular index, then let φ_t be the regular FOP of degree t. Else define φ_t as $\varphi_r \psi_{t,r}$ where r is the largest regular index less than t and $\psi_{t,r}$ is an arbitrary monic polynomial of degree $t - r$. In the latter case φ_t is called an *inner polynomial*. If $\psi_{t,r}(z) = z^{t-r}$ then we say that φ_t is defined *by using the standard monomial basis.* These polynomials $\{\varphi_t\}_{t=0}^{\infty}$ can be grouped into $J+1$ blocks

$$\begin{aligned}
\Phi^{(0)} &:= \begin{bmatrix} \varphi_0 \end{bmatrix} \\
\Phi^{(1)} &:= \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_{k_2-1} \end{bmatrix} \\
\Phi^{(2)} &:= \begin{bmatrix} \varphi_{k_2} & \varphi_{k_2+1} & \cdots & \varphi_{k_3-1} \end{bmatrix} \\
&\vdots \qquad\qquad \vdots \\
\Phi^{(J-1)} &:= \begin{bmatrix} \varphi_{k_{J-1}} & \varphi_{k_{J-1}+1} & \cdots & \varphi_{k_J-1} \end{bmatrix} \\
\Phi^{(J)} &:= \begin{bmatrix} \varphi_n & \varphi_{n+1} & \cdots & \end{bmatrix}.
\end{aligned}$$

Note that each block starts with a regular FOP and that the remaining polynomials in each block are inner polynomials. The pth block has length $l_p := k_{p+1} - k_p$ for $p = 0, 1, \ldots, J-1$. Let

$$\langle \Psi, \Phi \rangle := \begin{bmatrix} \langle \psi_0, \phi_0 \rangle & \cdots & \langle \psi_0, \phi_q \rangle \\ \vdots & & \vdots \\ \langle \psi_p, \phi_0 \rangle & \cdots & \langle \psi_p, \phi_q \rangle \end{bmatrix} \in \mathbb{C}^{(p+1)\times(q+1)}$$

for any two row vectors

$$\Psi := \begin{bmatrix} \psi_0 & \psi_1 & \cdots & \psi_p \end{bmatrix} \qquad \text{and} \qquad \Phi := \begin{bmatrix} \phi_0 & \phi_1 & \cdots & \phi_q \end{bmatrix}$$

of polynomials in $\mathcal{P}$.

THEOREM 5. *The following block orthogonality relations hold:*

$$\langle \Phi^{(p)}, \Phi^{(q)} \rangle = \begin{cases} 0_{l_p \times l_q} & \text{if } p \neq q \\ \delta_p & \text{if } p = q \end{cases} \quad \text{for } p, q = 0, 1, \ldots, J-1.$$

The matrix $\delta_p \in \mathbb{C}^{l_p \times l_p}$ is nonsingular and symmetric for $p = 0, 1, \ldots, J-1$. Its entries are equal to zero above the main antidiagonal and equal to $\langle z^{k_p + l_p - 1}, \varphi_{k_p} \rangle$

along the main antidiagonal. Also, if all the inner polynomials of the block $\Phi^{(p)}$ *where* $p \in \{1, \ldots, J-1\}$ *are defined by using the standard monomial basis, then* δ_p *is a Hankel matrix.*

PROOF. This result is well-known in the theory of FOPs. However, as readers who are less familiar with formal orthogonal polynomials than with the literature concerning the computation of zeros may find themselves slightly overwhelmed by the amount of information given in, for example, the survey papers by Gutknecht [**128, 130**] or the book by Bultheel and Van Barel [**47**], we prefer to include a proof.

The proof is by induction. Obviously, $\langle \Phi^{(0)}, \Phi^{(0)} \rangle = [\langle 1, 1 \rangle] = [s_0]$ is nonsingular. Now suppose that the theorem holds for $p, q = 0, 1, \ldots, k-1$ where $k \in \{1, \ldots, J-1\}$. Consider the block $\Phi^{(k)}$. Let us call the first polynomial of this block φ_r and let l be the length of this block,

$$\Phi^{(k)} = \begin{bmatrix} \varphi_r & \varphi_{r+1} & \cdots & \varphi_{r+l-1} \end{bmatrix}.$$

Then the matrices $H_{r+1}, \ldots, H_{r+l-1}$ are singular while H_r and H_{r+l} are nonsingular. By symmetry considerations it suffices to prove that

$$\hat{K} := \begin{bmatrix} \langle \varphi_0, \varphi_r \rangle & \cdots & \langle \varphi_0, \varphi_{r+l-1} \rangle \\ \langle \varphi_1, \varphi_r \rangle & \cdots & \langle \varphi_1, \varphi_{r+l-1} \rangle \\ \vdots & & \vdots \\ \langle \varphi_{r-1}, \varphi_r \rangle & \cdots & \langle \varphi_{r-1}, \varphi_{r+l-1} \rangle \end{bmatrix} = 0_{r \times l}$$

and that the matrix

$$\hat{\delta} := \begin{bmatrix} \langle \varphi_r, \varphi_r \rangle & \cdots & \langle \varphi_r, \varphi_{r+l-1} \rangle \\ \vdots & & \vdots \\ \langle \varphi_{r+l-1}, \varphi_r \rangle & \cdots & \langle \varphi_{r+l-1}, \varphi_{r+l-1} \rangle \end{bmatrix}$$

is nonsingular and has the other properties mentioned. Let F_l be the $l \times l$ unit upper triangular matrix that contains the coefficients of the polynomials

$$1, \psi_{r+1,r}, \ldots, \psi_{r+l-1,r}$$

(used in the definition of the inner polynomials of the block $\Phi^{(k)}$). Then

$$\hat{K} = K F_l \qquad \text{and} \qquad \hat{\delta} = F_l^T \delta F_l$$

if the matrices K and δ are defined as

$$K := \begin{bmatrix} \langle \varphi_0, \varphi_r \rangle & \langle \varphi_0, z\varphi_r \rangle & \cdots & \langle \varphi_0, z^{l-1}\varphi_r \rangle \\ \vdots & \vdots & & \vdots \\ \langle \varphi_{r-1}, \varphi_r \rangle & \langle \varphi_{r-1}, z\varphi_r \rangle & \cdots & \langle \varphi_{r-1}, z^{l-1}\varphi_r \rangle \end{bmatrix}$$

and

$$\delta := \begin{bmatrix} \langle \varphi_r, \varphi_r \rangle & \langle \varphi_r, z\varphi_r \rangle & \cdots & \langle \varphi_r, z^{l-1}\varphi_r \rangle \\ \langle z\varphi_r, \varphi_r \rangle & & \ddots & \vdots \\ \vdots & \ddots & & \vdots \\ \langle z^{l-1}\varphi_r, \varphi_r \rangle & \cdots & \cdots & \langle z^{l-1}\varphi_r, z^{l-1}\varphi_r \rangle \end{bmatrix}.$$

(In other words, K and δ correspond to the situation where all the inner polynomials in our block are defined by using the standard monomial basis.) Therefore we will

first study the matrices K and δ. Observe that δ is a Hankel matrix. As φ_r is a regular FOP, we may conclude that $\langle \varphi_s, z^t \varphi_r \rangle = 0$ for $t \geq 0$ and $s = 0, 1, \ldots, r-t-1$. Thus

$$K = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \\ \vdots & & \iddots & \times \\ \vdots & \iddots & \iddots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}. \tag{33}$$

Let us consider the first antidiagonal of K whose entries we have not yet proven to be equal to zero. As φ_r is orthogonal to all polynomials of degree $\leq r-1$, all these entries are equal. Indeed,

$$\langle \varphi_{r-1}, z\varphi_r \rangle = \langle \varphi_{r-2}, z^2 \varphi_r \rangle = \cdots = \langle \varphi_{r-l+1}, z^{l-1} \varphi_r \rangle = \langle z^r, \varphi_r \rangle.$$

Note that $\langle z^r, \varphi_r \rangle = \langle \varphi_r, \varphi_r \rangle$, the entry in the upper left corner of δ. As H_{r+1} is singular and $\langle \varphi_r, \varphi_r \rangle = \det H_{r+1} / \det H_r$, it follows that $\langle \varphi_r, \varphi_r \rangle = 0$. This implies that φ_r is in fact orthogonal to all polynomials of degree $\leq r$ and that the upper left entry of δ as well as all the entries on our antidiagonal of K are equal to zero. Now we continue with the next antidiagonal of K. The fact that $\langle \varphi_r, \varphi_r \rangle = 0$ implies that all its entries are equal to $\langle z^{r+1}, \varphi_r \rangle = \langle z\varphi_r, \varphi_r \rangle$. One can easily check that

$$\begin{aligned} R_{r+2}^T H_{r+2} R_{r+2} &= \left[\langle \varphi_p, \varphi_q \rangle \right]_{p,q=0}^{r+1} \\ &= \operatorname{diag}(\delta_0, \delta_1, \ldots, \delta_{k-1}) \oplus F_2^T \begin{bmatrix} 0 & \langle z^{r+1}, \varphi_r \rangle \\ \langle z^{r+1}, \varphi_r \rangle & \langle z\varphi_r, z\varphi_r \rangle \end{bmatrix} F_2 \end{aligned}$$

if R_{r+2} is the unit upper triangular matrix that contains the coefficients of the polynomials $\varphi_0, \varphi_1, \ldots, \varphi_{r+1}$. As H_{r+2} is singular and $\delta_0, \delta_1, \ldots, \delta_{k-1}$ are nonsingular, it follows that $\langle z^{r+1}, \varphi_r \rangle = 0$. Thus φ_r is orthogonal to all polynomials of degree $\leq r+1$ and all the entries on our antidiagonal of K as well as two additional entries of δ are equal to zero. And so on. Eventually we will find that all the entries of K that are marked $\times$ in (33) are determined by the first $l-1$ entries of the first row of δ "in a Hankel way," i.e., by shifting these entries to the north-east. We will also find that $\langle z^{r+2}, \varphi_r \rangle = \cdots = \langle z^{r+l-2}, \varphi_r \rangle = 0$, i.e., φ_r is orthogonal to all polynomials of degree $\leq r+l-2$ (because $H_{r+3}, \ldots, H_{r+l-1}$ are singular) while $\langle z^{r+l-1}, \varphi_r \rangle \neq 0$ (because H_{r+l} is nonsingular). Therefore $\hat{K} = K = 0_{r \times l}$ and δ is a nonsingular lower triangular Hankel matrix. One can easily verify that this implies that $\hat{\delta}$ is indeed nonsingular, symmetric, zero above the main antidiagonal, and equal to $\langle z^{r+l-1}, \varphi_r \rangle$ along the main antidiagonal. This proves the theorem. □

Note that $\langle \Phi^{(p)}, \Phi^{(J)} \rangle = 0_{l_p \times \infty}$ for $p = 0, 1, \ldots, J$ if we set $l_J := \infty$.

Let G be the infinite Gram matrix

$$G := [\, \langle \varphi_p, \varphi_q \rangle \,]_{p,q \geq 0}$$

and let G_k be its $k \times k$ leading principal submatrix for $k = 1, 2, \ldots$. Then

$$G = G_n \oplus 0_{\infty \times \infty}.$$

The matrix G_n is nonsingular and block diagonal, $G_n = \operatorname{diag}(\delta_0, \delta_1, \ldots, \delta_{J-1})$. Let $G^{(1)}$ be the infinite matrix

$$G^{(1)} := [\, \langle \varphi_p, \varphi_1 \varphi_q \rangle \,]_{p,q \geq 0}$$

and let $G_k^{(1)}$ be its $k \times k$ leading principal submatrix for $k = 1, 2, \ldots$. Then

$$G^{(1)} = G_n^{(1)} \oplus 0_{\infty \times \infty}.$$

Note that G and $G^{(1)}$ are both symmetric. As already mentioned, the matrices G_n and $G_n^{(1)}$ will play an important role in Section 3. The following theorem examines the structure of $G_n^{(1)}$. Let us agree to call a matrix $A = [\, a_{p,q} \,]_{p,q=0}^{l-1} \in \mathbb{C}^{l \times l}$ *lower anti-Hessenberg* if $a_{p,q} = 0$ whenever $p + q < l - 2$, i.e., a matrix will be called lower anti-Hessenberg if its entries are equal to zero along all the antidiagonals that lie above the main antidiagonal, except for the antidiagonal that precedes the main antidiagonal.

THEOREM 6. *The following block orthogonality relations hold:*

$$\langle \Phi^{(p)}, \varphi_1 \Phi^{(q)} \rangle = \begin{cases} 0_{l_p \times l_q} & \text{if } |p - q| > 1 \\ \kappa_p & \text{if } p = q + 1 \\ \kappa_q^T & \text{if } p = q - 1 \\ \delta_p^{(1)} & \text{if } p = q \end{cases} \qquad \text{for } p, q = 0, 1, \ldots, J - 1.$$

In other words, the matrix $G_n^{(1)}$ is block tridiagonal,

$$G_n^{(1)} = \begin{bmatrix} \delta_0^{(1)} & \kappa_1^T & & & \\ \kappa_1 & \delta_1^{(1)} & \kappa_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \kappa_{J-2} & \delta_{J-2}^{(1)} & \kappa_{J-1}^T \\ & & & \kappa_{J-1} & \delta_{J-1}^{(1)} \end{bmatrix}.$$

The matrix $\delta_p^{(1)} \in \mathbb{C}^{l_p \times l_p}$ is symmetric and lower anti-Hessenberg for $p = 0, 1, \ldots, J-1$. Its entries are equal to $\langle z^{k_p + l_p - 1}, \varphi_{k_p} \rangle$ along its first nonzero antidiagonal. Also, if all the inner polynomials of the block $\Phi^{(p)}$ where $p \in \{1, \ldots, J-1\}$ are defined by using the standard monomial basis, then $\delta_p^{(1)}$ is a Hankel matrix. Note that $\delta_0^{(1)} = [\, 0 \,]$. All the entries of the matrix κ_p where $p \in \{1, \ldots, J-1\}$ are equal to zero, except for the entry in the south-east corner, which is equal to $\langle z^{k_p + l_p - 1}, \varphi_{k_p} \rangle$.

PROOF. The proof is left to the reader. Use the results obtained in the proof of Theorem 5. □

Thus, for example, the matrices G_n and $G_n^{(1)}$ may look like

$$G_n = \begin{bmatrix}
\otimes & & & & & & & & & & & \\
 & 0 & 0 & \otimes & & & & & & & & \\
 & 0 & \otimes & \times & & & & & & & & \\
 & \otimes & \times & \times & & & & & & & & \\
 & & & & 0 & 0 & \otimes & & & & & \\
 & & & & 0 & \otimes & \times & & & & & \\
 & & & & \otimes & \times & \times & & & & & \\
 & & & & & & & 0 & 0 & 0 & \otimes \\
 & & & & & & & 0 & 0 & \otimes & \times \\
 & & & & & & & 0 & \otimes & \times & \times \\
 & & & & & & & \otimes & \times & \times & \times
\end{bmatrix}$$

and

$$G_n^{(1)} = \begin{bmatrix}
0 & & & \otimes & & & & & & & & \\
 & 0 & \otimes & \times & & & & & & & & \\
 & \otimes & \times & \times & & & & & & & & \\
\otimes & \times & \times & \times & & & \otimes & & & & & \\
 & & & & 0 & \otimes & \times & & & & & \\
 & & & & \otimes & \times & \times & & & & & \\
 & & & \otimes & \times & \times & \times & & & & \otimes & \\
 & & & & & & & 0 & 0 & \otimes & \times \\
 & & & & & & & 0 & \otimes & \times & \times \\
 & & & & & & & \otimes & \times & \times & \times \\
 & & & & & & \otimes & \times & \times & \times & \times
\end{bmatrix}.$$

The entries marked $\otimes$ are different from zero. Also, in each block they are all equal.

3. An accurate algorithm to compute zeros of FOPs

We will now discuss an algorithm to compute zeros of FOPs. More precisely, we will show how FOPs can be computed in their product representation. Therefore, the polynomial φ_n immediately leads to the zeros $z_1, \ldots, z_n$. Our numerical experiments indicate that our algorithm gives very accurate results.

REMARK. Our aim is to present techniques for computing zeros of analytic functions that give very accurate results. As the reader will notice, if we have a choice between several options for a certain part of an algorithm, then we will always choose the option that, in our experience, gives the most accurate results, even if it is the most expensive (though still within the limits of what is reasonable, of course) in terms of number of floating point operations. The emphasis lies on accuracy.

Theorem 4 can be interpreted in the following way: the zeros of the regular FOP of degree n can be calculated by solving a generalized eigenvalue problem. The following theorem shows that this zero/eigenvalue property holds for all regular FOPs. This will enable us to evaluate regular FOPs in their product representation, which is numerically very stable. The theorem also provides a solution to the problem of how to switch from ordinary moments to modified moments.

THEOREM 7. *Let $t \geq 1$ be a regular index and let $z_{t,1}, \ldots, z_{t,t}$ be the zeros of the regular FOP φ_t. Then the eigenvalues of the pencil $G_t^{(1)} - \lambda G_t$ are given by $\varphi_1(z_{t,1}), \ldots, \varphi_1(z_{t,t})$. In other words, they are given by $z_{t,1} - \mu, \ldots, z_{t,t} - \mu$ where $\mu = s_1/s_0$.*

PROOF. The first part of the proof is similar to the proof of Theorem 4. Define the Hankel matrix $H_t^<$ as $H_t^< := [s_{1+k+l}]_{k,l=0}^{t-1}$. We will first show that the zeros of φ_t are given by the eigenvalues of the pencil $H_t^< - \lambda H_t$. The zeros of φ_t are given by the eigenvalues of its companion matrix C_t. Let $\lambda^\star$ be an eigenvalue of C_t and x a corresponding eigenvector. As H_t is nonsingular, we may conclude that $C_t x = \lambda^\star x \Leftrightarrow H_t C_t x = \lambda^\star H_t x$. Using (25) one can easily verify that $H_t C_t = H_t^<$.

Let A_t be the unit upper triangular matrix that contains the coefficients of $\varphi_0, \varphi_1, \ldots, \varphi_{t-1}$. Then G_t can be factorized as $G_t = A_t^T H_t A_t$. As $\varphi_1(z) = z - \mu$ where $\mu = s_1/s_0$, the matrix $G_t^{(1)}$ is given by $[\langle \varphi_r, z\varphi_s \rangle]_{r,s=0}^{t-1} - \mu G_t$. The matrix $[\langle \varphi_r, z\varphi_s \rangle]_{r,s=0}^{t-1}$ can be written as $A_t^T H_t^< A_t$ and thus $G_t^{(1)} = A_t^T (H_t^< - \mu H_t) A_t$. Now let $\lambda^\star$ be an eigenvalue of the pencil $H_t^< - \lambda H_t$ and x a corresponding eigenvector. Then

$$
\begin{aligned}
& H_t^< x = \lambda^\star H_t x \\
\Leftrightarrow \quad & (H_t^< - \mu H_t) x = (\lambda^\star - \mu) H_t x \\
\Leftrightarrow \quad & A_t^T (H_t^< - \mu H_t) A_t y = \varphi_1(\lambda^\star) A_t^T H_t A_t y \quad \text{if } y := A_t^{-1} x \\
\Leftrightarrow \quad & G_t^{(1)} y = \varphi_1(\lambda^\star) G_t y.
\end{aligned}
$$

This proves the theorem. □

COROLLARY 8. *The eigenvalues of $G_n^{(1)} - \lambda G_n$ are given by $z_1 - \mu, \ldots, z_n - \mu$ where $\mu = s_1/s_0$.*

Regular FOPs are characterized by the fact that the determinant of a Hankel matrix is different from zero, while inner polynomials correspond to singular Hankel matrices. To decide whether $\varphi_t(z)$ should be defined as a regular FOP or as an inner polynomial, one could therefore calculate the determinant of H_t and check if it is equal to zero. However, from a numerical point of view such a test "is equal to zero" does not make sense. Because of rounding errors (both in the evaluation of $\langle \cdot, \cdot \rangle$ and the calculation of the determinant) we would encounter only regular FOPs. Strictly speaking one could say that inner polynomials are not needed in numerical calculations. However, the opposite is true! Let us agree to call a regular FOP *well-conditioned* if its corresponding Yule-Walker system (25) is well-conditioned, and *ill-conditioned* otherwise. To obtain a numerically stable algorithm, it is crucial to generate only well-conditioned regular FOPs and to replace ill-conditioned regular FOPs by inner polynomials. Stable look-ahead solvers for linear systems of equations that have Hankel structure are based on this principle [**39, 51, 86**]. In this approach the diagonal blocks in G_n are taken (slightly) larger than strictly necessary to avoid ill-conditioned blocks. A disadvantage is that part of the structure of G_n and $G_n^{(1)}$ gets lost, i.e., there will be some additional fill-in.

Our algorithm for calculating $z_1, \ldots, z_n$ proceeds by computing the polynomials $\varphi_0(z), \varphi_1(z), \ldots, \varphi_n(z)$ in their product representation, starting with $\varphi_0(z) \leftarrow 1$ and

$\varphi_1(z) \leftarrow z - \mu$. At each step we ask ourselves whether it is numerically feasible to generate the next polynomial in the sequence as a regular FOP. As the reader will see, there are several ways to find an answer to this question.

How do we obtain the value of n? Theorem 3 and Equations (23) and (27) imply the following.

THEOREM 9. *Let $t \geq n$. Then $\varphi_t(z_k) = 0$ for $k = 1, \ldots, n$ and $\langle z^p, \varphi_t(z) \rangle = 0$ for all $p \geq 0$.*

The value of n can be determined as follows. Suppose that the algorithm has just generated a (well-conditioned) regular FOP $\varphi_r(z)$. To check whether $n = r$, we scan the sequence

$$\big(|\langle (z-\mu)^\tau \varphi_r(z), \varphi_r(z) \rangle| \big)_{\tau=0}^{N-1-r}. \tag{34}$$

If all the elements are "sufficiently small," then we conclude that indeed $n = r$ and we stop.

The form $\langle \cdot, \cdot \rangle$ is evaluated via numerical integration along γ. In other words, it is approximated by a quadrature sum. We assume that this sum is calculated in the standard way, by adding the terms one by one, in other words, by forming a sequence of partial sums. We ask the quadrature algorithm not only for an approximation of the integral, say `result`, but also for the modulus of the partial sum that has the largest modulus, say `maxpsum`. Then

$$\log_{10} \frac{\texttt{maxpsum}}{|\texttt{result}|}$$

is an estimate for the loss of precision. This information will turn out to be extremely useful, for example in the stopping criterion.

These considerations lead to the following algorithm.

ALGORITHM

input $\langle \cdot, \cdot \rangle$, ϵ_{stop}
output n, zeros
comment zeros $= \{z_1, \ldots, z_n\}$. We assume that $\epsilon_{\text{stop}} > 0$.
$N \leftarrow \langle 1, 1 \rangle$
if $N == 0$ **then**
  $n \leftarrow 0$; zeros $\leftarrow \emptyset$; **stop**
else
  $\varphi_0(z) \leftarrow 1$
  $\mu \leftarrow \langle z, 1 \rangle / N$; $\varphi_1(z) \leftarrow z - \mu$
  $r \leftarrow 1$; $t \leftarrow 0$
  while $r + t < N$ **do**
    regular $\leftarrow$ it is numerically feasible to generate $\varphi_{r+t+1}(z)$ as
      a regular FOP ... **[1]**
    if regular **then**
      generate $\varphi_{r+t+1}(z)$ as a regular FOP ... **[2]**
      $r \leftarrow r + t + 1$; $t \leftarrow 0$
      allsmall $\leftarrow$ **true**; $\tau \leftarrow 0$

```
      while allsmall and (r + τ < N) do
        [ip, maxpsum] ← ⟨(z − μ)^τ φ_r(z), φ_r(z)⟩        ... [3]
        ip ← |ip|
        allsmall ← ( ip/maxpsum < ε_stop )        ... [4]
        τ ← τ + 1
      end while
      if allsmall then
        n ← r; zeros ← roots(φ_r); stop
      end if
    else
      generate φ_{r+t+1}(z) as an inner polynomial     ... [5]
      t ← t + 1
    end if
  end while
  n ← N; zeros ← roots(φ_N); stop
end if
```

Comments:

1. Statement [**1**] is crucial. But how does one decide that it is numerically feasible to generate the next polynomial in the sequence $\varphi_0(z), \varphi_1(z), \ldots, \varphi_n(z)$ as a regular FOP? Suppose that the algorithm has just generated a regular FOP $\varphi_r(z)$. Then (26) implies that if r is a regular index, then $r+1$ is a regular index if and only if $\langle \varphi_r(z), \varphi_r(z) \rangle \neq 0$. This suggests the following criterion: if $|\langle \varphi_r(z), \varphi_r(z) \rangle| / \texttt{maxpsum} < \epsilon_{\text{regular}}$, where $\epsilon_{\text{regular}}$ is some small threshold given by the user, then define $\varphi_{r+1}(z)$ as an inner polynomial, else define it as a regular FOP. However, as we will illustrate in the next section, it is very difficult to choose an appropriate value of $\epsilon_{\text{regular}}$.

   We prefer to use the following criterion: act as if the next polynomial in the sequence, say $\varphi_t(z)$, is defined as a regular FOP, i.e., compute its zeros by computing the eigenvalues of the pencil $G_t^{(1)} - \lambda G_t$ and then check if these zeros lie sufficiently close to the interior of γ. If so, then define $\varphi_t(z)$ as a regular FOP, else define it as an inner polynomial. The idea behind this strategy is the following. If the matrix G_t is singular, in which case also the matrix H_t is singular of course, then the pencil $G_t^{(1)} - \lambda G_t$ has either a number of eigenvalues at infinity or a number of eigenvalues that may assume arbitrary values. Indeed, by using the structure of the matrices $G_t^{(1)}$ and G_t one can easily prove the following result, which complements Theorem 7.

   THEOREM 10. *Let $t \geq 1$ be an integer, let r be the largest regular index less than or equal to t, and let r' be the smallest regular index greater than t. (Define $r' := +\infty$ if $t \geq n$.) Then the eigenvalues of the pencil $G_t^{(1)} - \lambda G_t$ are given by the eigenvalues of the pencil $G_r^{(1)} - \lambda G_r$ and $t - r$ eigenvalues that may assume arbitrary values if $t < r' - 1$ or $t - r$ eigenvalues $\lambda = \infty$ if $t = r' - 1$.*

   Each of these indeterminate eigenvalues corresponds to two corresponding zeros on the diagonals of the generalized Schur decomposition of $G_t^{(1)}$ and

G_t. When actually calculated, these diagonal entries are different from zero because of roundoff errors. The quotient of two such corresponding diagonal entries is a spurious eigenvalue. Our strategy is based on the assumption that, if the matrix H_t, and thus also the matrix G_t, is nearly singular, then the computed eigenvalues of the pencil $G_t^{(1)} - \lambda G_t$ that correspond to the eigenvalues that lie at infinity or that may assume arbitrary values, lie far away from the interior of γ.

The reader may object that our criterion is too strict. Indeed, the zeros of the regular FOPs of degree $< n$ need not lie close to the interior of γ, except if the form $\langle \cdot, \cdot \rangle$ is a true (positive definite) inner product, in which case the zeros of the regular FOPs lie in the convex hull of $\{z_1, \ldots, z_n\}$. (This follows from a general result on orthogonal polynomials. See, e.g., Van Assche [**273**].) Thus it may very well be the case that some of the computed zeros of a well-conditioned regular FOP lie far away from γ, in which case our algorithm decides to define this polynomial as an inner polynomial. In other words, our algorithm may define more inner polynomials than strictly necessary. However, we have done a lot of numerical tests and have found that our strategy leads to very accurate results. Also, compared to the criterion based on inner products of the type $\langle \varphi_r(z), \varphi_r(z) \rangle$, another advantage is that the user doesn't have to supply a threshold such as $\epsilon_{\text{regular}}$.

2. Statement [**2**] means: define $\varphi_{r+t+1}(z)$ as $\varphi_{r+t+1}(z) \leftarrow \prod_{j=1}^{r+t+1}(z - \alpha_j)$. The zeros α_j are computed as $\alpha_j = \mu + \lambda_j$, $j = 1, \ldots, r+t+1$, where $\lambda_1, \ldots, \lambda_{r+t+1}$ are the eigenvalues of the pencil $G_{r+t+1}^{(1)} - \lambda G_{r+t+1}$, cf. Theorem 7.

3. In statement [**3**] we use the inner product $\langle (z-\mu)^\tau \varphi_r(z), \varphi_r(z) \rangle$ and not $\langle z^\tau \varphi_r(z), \varphi_r(z) \rangle$ as it is likely that the former leads to more accurate results than the latter. If $\tau \leq r$, then one may also use $\langle \varphi_\tau(z) \varphi_r(z), \varphi_r(z) \rangle$. In general, if $\tau = \alpha r + \beta$, where $\alpha, \beta \in \mathbb{N}$ with $\beta < r$, then one may use $\langle [\varphi_r(z)]^{\alpha+1} \varphi_\beta(z), \varphi_r(z) \rangle$.

4. Observe that in statement [**4**] we do not compare `ip` with ϵ_{stop} but take into account the loss of precision as estimated by the quadrature algorithm. We have found this heuristic to be very reliable.

5. In statement [**5**] one may define $\varphi_{r+t+1}(z)$ as $\varphi_{r+t+1}(z) \leftarrow (z-\mu)\varphi_{r+t}(z)$ or $\varphi_{r+t+1}(z) \leftarrow \varphi_{t+1}(z)\varphi_r(z)$. In our opinion, both versions are to be preferred to the "classical" $\varphi_{r+t+1}(z) \leftarrow z^{t+1}\varphi_r(z)$.

6. Instead of computing μ, the arithmetic mean of the zeros, as $\mu \leftarrow \langle z, 1 \rangle / N$, one can also use the following formula, which may give a more accurate result: $\mu \leftarrow w + \langle z - w, 1 \rangle / N$, where w is a point inside γ, preferably near the centre of the interior of γ.

7. As we represent our FOPs by using the product representation,

$$\varphi(z) = \prod_{\alpha \in \varphi^{-1}(0)} (z - \alpha),$$

the funtion roots$(\cdot)$ is obviously *not* a function that calculates the zeros of a polynomial from its coefficients in the standard monomial basis.

We have implemented our algorithm in Matlab (for disks) and also in Fortran 90 (for rectangular regions). The latter implementation will be presented in more detail in Section 5. Numerical examples will be given in Section 4 and also in Section 5. They will illustrate the effectiveness of our approach.

4. Numerical examples

In the following examples we have considered the case that γ is a circle. The computations have been done via Matlab 5 (with floating point relative accuracy $\approx 2.2204\,10^{-16}$).

The following integration algorithm is used to approximate the form $\langle \cdot, \cdot \rangle$. Let γ be the circle with centre c and radius ρ. Then

$$\langle \phi, \psi \rangle = \rho \int_0^1 \phi(c + \rho e^{2\pi i\theta}) \psi(c + \rho e^{2\pi i\theta}) \frac{f'(c + \rho e^{2\pi i\theta})}{f(c + \rho e^{2\pi i\theta})} e^{2\pi i\theta}\, d\theta. \tag{35}$$

Since this is the integral of a periodic function over a complete period, the trapezoidal rule is an appropriate quadrature rule. If $F : [0,1] \to \mathbb{C}$ is the integrand in the right-hand side of (35), then the q-point trapezoidal rule approximation to $\langle \phi, \psi \rangle$ is given by

$$\langle \phi, \psi \rangle = \int_0^1 F(\theta)\, d\theta \;\approx\; \frac{1}{q} {\sum_{k=0}^{q}}'' F(k/q) =: T_q.$$

The double prime indicates that the first and the last term of the sum are to be multiplied by $1/2$. As F is periodic with period one, we may rewrite T_q as

$$T_q = \frac{1}{q} \sum_{k=0}^{q-1} F(k/q).$$

This shows that T_q indeed depends on q (and not $q+1$) points. As

$$T_{2q} = \frac{1}{2} T_q + T_{q \to 2q}$$

where

$$T_{q \to 2q} := \frac{1}{2q} \sum_{k=0}^{q-1} F\Big(\frac{2k+1}{2q}\Big),$$

successive doubling of q enables us in each step to reuse the integrand values needed in the previous step. In the following examples we started with $q = 16$ and continued doubling q until $|T_{2q} - T_q|$ was sufficiently small. More precisely, if S_q and $S_{q \to 2q}$ denote the modulus of the partial sum of qT_q respectively $2qT_{q \to 2q}$ that has the largest modulus, then our stopping criterion is given by $|T_{2q} - T_q| \le S_{2q} 10^{-14}$, where $S_{2q} := \max\{S_q, S_{q \to 2q}\}/2q$.

Lyness and Delves [**203**] studied the asymptotic behaviour of the quadrature error. They showed that the modulus of the error made by the q-point trapezoidal rule is asymptotically $\mathcal{O}(A^q)$ where $0 \le A < 1$. More precisely,

$$A := \max\Big\{\frac{|z_I|}{\rho}, \frac{\rho}{|z_E|}, \frac{\rho}{\rho_s}\Big\}$$

where z_I is the zero of f that lies closest to γ and in the interior of γ, z_E is the zero of f that lies closest to γ and in the exterior of γ, and ρ_s is the distance between c and the nearest singularity of f.

EXAMPLE 2. Our first example illustrates the importance of shifting the origin in the complex plane to the arithmetic mean of the zeros. It also compares the two strategies that we have proposed to decide whether it is numerically feasible to generate the next polynomial in the sequence $\varphi_0(z), \varphi_1(z), \ldots, \varphi_n(z)$ as a regular FOP. We will see that it is indeed better to act as if the polynomial is a regular FOP, i.e., to compute its zeros by solving the generalized eigenvalue problem of Theorem 7, and then to check if these zeros lie sufficiently close to the interior of γ. If so, the polynomial is indeed defined as a regular FOP, else it is defined as an inner polynomial.

Suppose that $n = 3$, $z_1 = \epsilon$, $z_2 = \sqrt{3} + i$, $z_3 = \sqrt{3} - i$, and $\nu_1 = \nu_2 = \nu_3 = 1$. That is, suppose that $f(z) = (z - \epsilon)[(z - \sqrt{3})^2 + 1]$. If $\epsilon = 0$, then the Hankel matrix H_2 is exactly singular, i.e., $\varphi_2(z)$ has to be defined as an inner polynomial. We set $\epsilon = 10^{-2}$. Suppose that $\gamma = \{ z \in \mathbb{C} : |z| = 3 \}$. In the quadrature algorithm, we have evaluated the logarithmic derivative $f'(z)/f(z)$ of $f(z)$ via the formula

$$\frac{f'(z)}{f(z)} = \sum_{k=1}^{n} \frac{\nu_k}{z - z_k}. \tag{36}$$

We have taken $\epsilon_{\text{stop}} = 10^{-18}$. Our algorithm proceeds as follows. The total number of zeros N is equal to 3. The polynomial $\varphi_0(z)$ is of course defined as a regular FOP, $\varphi_0(z) \leftarrow 1$. The arithmetic mean μ is approximated by

```
1.158033871706381e+00 + i  1.526556658859590e-16.
```

The polynomial $\varphi_1(z)$ is also defined as a regular FOP, $\varphi_1(z) \leftarrow z - \mu$. The inner product $\langle \varphi_1(z), \varphi_1(z) \rangle$ is equal to 0.02303. To take into account the loss of precision, we divide by S_{2q} to obtain its scaled counterpart, which is equal to 0.01231. Is this that small that we should define $\varphi_2(z)$ as an inner polynomial? It seems not, and we decide to define $\varphi_2(z)$ as a regular FOP. Its zeros are approximated by

```
1.158072473165547e+00 + i  1.513258564730580e-16
2.000049565733722e+02 + i  3.487413819470906e-12
```

Note how large the second zero is! The inner product $\langle \varphi_2(z), \varphi_2(z) \rangle$ is equal to 910.504. Its scaled counterpart is 0.01214. We decide to define $\varphi_3(z)$ as a regular FOP. Its zeros are approximated by

```
1.732050807571817e+00 - i  1.000000000004209e+00
1.000000000661760e-02 + i  3.913802997325630e-12
1.732050807566777e+00 + i  1.000000000002807e+00
```

As $N = 3$, we may stop. The relative errors of the approximations for the zeros of f are $\mathcal{O}(10^{-12})$, except for the zero that approximates $z_1 = \epsilon$, which has a relative error of $\mathcal{O}(10^{-10})$. The absolute errors are $\mathcal{O}(10^{-12})$. The relative errors of the approximations for the multiplicities are $\mathcal{O}(10^{-11})$.

As one of the zeros of $\varphi_2(z)$ lies far away from the interior of γ, we should decide to define $\varphi_2(z)$ as an inner polynomial. Surprisingly, this does not improve the accuracy of the results. However, let us see what happens if we first shift the

origin to μ, or, equivalently, if we consider the circle $\gamma = \{ z \in \mathbb{C} : |z-\mu| = 2 \}$. Note that we change both the centre and the radius of γ. By defining $\varphi_2(z)$ as a regular FOP, the accuracy of the results does not improve. However, if we define $\varphi_2(z)$ as an inner polynomial, the relative errors of the approximations for the zeros of f are $\mathcal{O}(10^{-16})$, except for the zero that approximates $z_1 = \epsilon$, which has a relative error of $\mathcal{O}(10^{-14})$. The absolute errors are $\mathcal{O}(10^{-16})$. The relative errors of the approximations for the multiplicities are $\mathcal{O}(10^{-16})$. In other words, the results are indeed much better. $\diamondsuit$

EXAMPLE 3. Let $f(z) = e^{3z} + 2z\cos z - 1$ and $\gamma = \{z \in \mathbb{C} : |z| = 2\}$. We set $\epsilon_{\text{stop}} = 10^{-18}$. Our algorithm finds that $N = 4$. It defines $\varphi_0(z)$ and $\varphi_1(z)$ as regular FOPs. From the eigenvalues of the pencil $G_2^{(1)} - \lambda G_2$ it concludes that $\varphi_2(z)$ would have a zero of modulus ≈ 43 in case $\varphi_2(z)$ is defined as a regular FOP. Thus the algorithm decides to define $\varphi_2(z)$ as an inner polynomial. The polynomials $\varphi_3(z)$ and $\varphi_4(z)$ are defined as regular FOPs. Our algorithm concludes that $n = 4$. The computed approximations for the zeros of f are given by

```
-1.844233953262213e+00 - i  1.106288924192872e-16
 5.308949302929297e-01 + i  1.331791876751121e+00
 5.308949302929303e-01 - i  1.331791876751121e+00
-5.412337245047638e-15 + i  3.762630283199076e-16
```

The corresponding approximations for the multiplicities are

```
1.000000000000001e+00 + i  9.279422312879846e-17
1.000000000000001e+00 - i  2.415808667423342e-15
1.000000000000001e+00 + i  1.187431378902999e-15
9.999999999999974e-01 + i  1.142195850770565e-15
```

By refining the approximations for the zeros of f via Newton's method, we find that they have a relative error of $\mathcal{O}(10^{-16})$, except for the approximation of $z_4 = 0$, which has an absolute error of $\mathcal{O}(10^{-15})$. If $\varphi_2(z)$ is defined as a regular FOP, the errors are $\mathcal{O}(10^{-13})$. $\diamondsuit$

EXAMPLE 4. Suppose that $f(z) = z^2(z-1)(z-2)(z-3)(z-4) + z\sin z$ and $\gamma = \{z \in \mathbb{C} : |z| = 5\}$. Note that f has a double zero at the origin. Our algorithm finds that $N = 6$. It defines $\varphi_0(z)$, $\varphi_1(z)$, $\varphi_2(z)$, $\varphi_3(z)$, $\varphi_4(z)$ and $\varphi_5(z)$ as regular FOPs. For $k = 2, 3, 4$ and 5, the scaled counterparts of $|\langle \varphi_k(z), \varphi_k(z) \rangle|$ are given by

```
7.040331724952680e-02
1.910625118197985e-03
1.236575744513765e-05
2.425617684377941e-15
```

If we set ϵ_{stop} to a value that is larger than $2.5\,10^{-15}$, then the algorithm stops. It decides (correctly) that $n = 5$. The computed approximations for the zeros of f are given by

```
2.853939307101427e-12 + i  1.218663779565894e-12
1.189065889993786e+00 + i  1.174492347177040e-10
1.728434986506658e+00 + i  1.587636971134256e-10
3.019907328131211e+00 + i  1.757549887398162e-11
4.030381916062330e+00 + i  7.769722658005202e-13
```

The corresponding approximations for the multiplicities are

```
2.000000000021947e+00 + i  9.312570667099433e-12
1.000000000957614e+00 + i  4.352560366652006e-10
9.999999990867526e-01 - i  4.117654248790392e-10
9.999999999418555e-01 - i  2.906361792434579e-11
9.999999999918315e-01 - i  3.739564528915041e-12
```

By refining the approximations for the zeros iteratively via Newton's method, we find that the absolute errors are $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-14})$ and $\mathcal{O}(10^{-16})$, respectively. The refined approximations for the zeros are given by

```
2.853939307034829e-12 + i  1.218663779496334e-12
1.189065890003747e+00 + i  1.218863825423787e-10
1.728434986503592e+00 + i  1.573384095073740e-10
3.019907328131241e+00 + i  1.759027974661844e-11
4.030381916062330e+00 + i  7.768170803990681e-13
```

If we set ϵ_{stop} to a value that is smaller than $2.4\,10^{-15}$, then the algorithm continues. It defines $\varphi_6(z)$ as a regular FOP and stops. The computed approximations for the zeros of f are now given by

```
-3.412647942013791e-11 - i  3.671305864695379e-12
 1.189065887181205e+00 - i  7.212695548387926e-11
 1.728434983085149e+00 - i  2.216255731479441e-11
 3.019907327801388e+00 + i  1.210604626933557e-11
 4.030381916045125e+00 + i  1.488578696458235e-12
-2.858307137641358e+00 + i  1.433406287105723e+00
```

whereas the corresponding approximations for the multiplicities are

```
1.999999999750989e+00 - i  1.982896140226239e-11
9.999999911197371e-01 - i  1.399505295104063e-10
1.000000008572735e+00 + i  1.723707922538161e-10
1.000000000489271e+00 - i  7.095346599149408e-12
1.000000000067249e+00 - i  5.486513691338396e-12
2.023577344691327e-14 - i  9.441050659615426e-15
```

Observe that we find more or less the same approximations for the zeros as before and also a spurious "zero," cf. Theorem 10. Fortunately, the presence of spurious zeros can be easily detected, as their corresponding "multiplicities" are equal to zero. This can be explained as follows. The Vandermonde matrix that corresponds to the calculated approximations for the zeros will almost surely (i.e., with probability one) be nonsingular and therefore the system for the multiplicities will have only one solution, which gives the true zeros of f their correct corresponding multiplicity and the spurious ones multiplicity zero. ♢

EXAMPLE 5. Let $f(z) = z^2(z-2)^2[\,e^{2z}\cos z + z^3 - 1 - \sin z\,]$ and $\gamma = \{z \in \mathbb{C} : |z| = 3\}$. Note that f has a triple zero at the origin and a double zero at $z = 2$. Our algorithm finds that $N = 8$. It defines $\varphi_0(z)$, $\varphi_1(z)$, $\varphi_2(z)$, $\varphi_3(z)$, $\varphi_4(z)$ and $\varphi_5(z)$ as regular FOPs. For $k = 2, 3, 4$ and 5, the scaled counterparts of $|\langle \varphi_k(z), \varphi_k(z) \rangle|$ are given by

```
1.515262455417321e-01
3.291787369922850e-02
2.663301879352920e-03
5.361771800010522e-15
```

If we set ϵ_{stop} to a value that is larger than $5.4\,10^{-15}$, then the algorithm stops. It decides (correctly) that $n = 5$. The computed approximations for the zeros of f are given by

```
-4.607141197276816e-01 + i  6.254277693477422e-01
-4.607141197280875e-01 - i  6.254277693470294e-01
 2.653322006551662e-12 + i  8.232702952841388e-13
 2.000000000000728e+00 + i  2.133638388712091e-13
 1.664682869749229e+00 + i  9.605975127915706e-13
```

The corresponding approximations for the multiplicities are

```
1.000000000003826e+00 + i  5.208513390226470e-12
1.000000000005729e+00 - i  1.653703539980123e-12
2.999999999991895e+00 - i  3.202939915103751e-12
1.999999999987495e+00 - i  3.449339571824102e-12
1.000000000011054e+00 + i  3.097374215634804e-12
```

By refining the approximations for the zeros iteratively via Newton's method, we find that the absolute errors are $\mathcal{O}(10^{-14})$, $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-13})$ and $\mathcal{O}(10^{-12})$, respectively.

If we set ϵ_{stop} to a value that is smaller than 10^{-15}, then the algorithm continues until the very end. For $k = 6$ and 7, the scaled counterparts of $|\langle\varphi_k(z), \varphi_k(z)\rangle|$ are given by

```
3.073186423911271e-15
1.658776045256565e-15
```

The algorithm defines $\varphi_6(z)$ and $\varphi_7(z)$ as regular FOPs. Then it defines $\varphi_8(z)$ as an inner polynomial and stops. The computed approximations for the zeros of f are now given by

```
-4.607141197287676e-01 + i  6.254277693479825e-01
-4.607141197287121e-01 - i  6.254277693478455e-01
 6.096234628216735e-13 + i  4.562446979887560e-14
 2.000000000000117e+00 - i  2.873008259688547e-13
 1.664682869746416e+00 - i  1.002467422184267e-12
-2.764701251474829e+00 + i  1.339901195840935e+00
 1.042158257408570e+00 - i  3.110403055104297e+00
 5.929068287859467e-01 - i  9.822871682718670e-17
```

whereas the corresponding approximations for the multiplicities are

```
1.000000000000205e+00 + i  1.226534018772486e-12
1.000000000000592e+00 - i  1.131203882923728e-12
2.999999999999770e+00 - i  2.952132467539878e-13
1.999999999997560e+00 + i  4.325897066507562e-12
1.000000000001872e+00 - i  4.180646412170987e-12
```

```
5.712723743305506e-16 + i  2.376958372514238e-16
4.180089925492345e-16 - i  2.883533010129535e-16
1.914114074394907e-13 - i  1.417681479033546e-13
```

As discussed in Example 4, the approximations for the multiplicities enable us to locate spurious zeros. ♢

EXAMPLE 6. Let us consider the Wilkinson polynomial of degree 10, $f(z) = \prod_{k=1}^{10}(z-k)$. Suppose that $\gamma = \{z \in \mathbb{C} : |z| = 11\}$. We have evaluated the logarithmic derivative of f via the formula (36). By using Theorem 4 we obtain the following approximations for the zeros of f:

```
1.000069039770840e+00 - i  6.902850482141518e-05
2.004976265975782e+00 - i  3.428732045109014e-03
3.060983206011711e+00 - i  2.595941486831997e-02
4.251792296513047e+00 - i  6.062429488533006e-02
5.546923068519217e+00 - i  6.702269982182261e-02
6.815527607409872e+00 - i  4.182297169281859e-02
7.960662733297854e+00 - i  1.296711240552063e-02
8.997081596675446e+00 - i  1.314415641505938e-03
9.999960580756190e+00 - i  2.207636166991653e-05
3.755000277916281e+00 - i  4.101268365758313e+00
```

The absolute errors of the approximations for the multiplicities are $\mathcal{O}(10^{-1})$.

Now let us see how our algorithm performs. We set $\epsilon_{\text{stop}} = 10^{-18}$. It finds that $N = 10$. It defines $\varphi_0(z), \varphi_1(z), \ldots, \varphi_9(z)$ as regular FOPs. For $k = 2, 3, \ldots, 9$, the scaled counterparts of $|\langle \varphi_k(z), \varphi_k(z) \rangle|$ are given by

```
3.816240840811839e-02
1.299688508906636e-03
3.553824638723713e-05
8.039698366555619e-07
1.469066688160238e-08
2.010063990239241e-10
1.220433804961808e-12
9.741759723349369e-16
```

This clearly indicates that the problem is very ill-conditioned. The algorithm defines $\varphi_{10}(z)$ as an inner polynomial and stops. The relative errors of the approximations for the zeros are given by

```
1.005577196493408e-04
2.863610448143864e-03
2.009371359668208e-02
5.876683150395577e-02
9.999999999999894e-02
8.141857700057402e-02
3.100882100301538e-02
6.609036435112427e-03
5.202783593342222e-04
7.598125455914798e-06
```

respectively. However, if we consider the circle $\gamma = \{z \in \mathbb{C} : |z - 5.5| = 5.5\}$, then the following happens. The algorithm defines only regular FOPs. For $k = 2, 3, \ldots, 9$, the scaled counterparts of $|\langle \varphi_k(z), \varphi_k(z) \rangle|$ are now given by

```
1.120889480330385e+00
3.130941249560411e-01
6.391288244829463e-02
9.005452570225671e-03
1.142227363432899e-03
1.122128881438772e-04
2.425547309622016e-06
2.149981073624070e-08
```

The relative errors of the approximations for the zeros are given by

```
8.362431354624777e-12
4.684075580626331e-10
6.568404577352624e-09
3.252872101214176e-08
6.598954186243715e-08
5.800798368415380e-08
2.196536348456049e-08
3.331673425077182e-09
1.672314153254196e-10
1.651045356918564e-12
```

whereas the absolute errors of the approximations for the multiplicities are

```
4.518653295430768e-11
3.537203913804699e-09
4.763707040887607e-08
1.808818111196228e-07
1.605226012463829e-07
1.316877594500569e-07
1.948007501440231e-07
6.042232266569582e-08
5.351870471708607e-09
9.475094893599826e-11
```

EXAMPLE 7. Recently, Engelborghs, Luzyanina and Roose [**75**] have used our algorithm to compute all the zeros of

$$f(z) = a + bz + z^2 - hz^2 e^{-\tau z},$$

where $a = 1$, $b = 0.5$, $h = -0.82465048736655$ and $\tau = 6.74469732735569$, that lie in the rectangular region

$$\{\, z \in \mathbb{C} : -0.3 \le \operatorname{Re} z \le 0.1, \quad -24.7 \le \operatorname{Im} z \le 24.7 \,\}.$$

The results are shown in Figure 3.

These zeros determine the stability of a steady state solution of a neutral functional differential equation. For the given values of the parameters a, b, h and τ, this steady

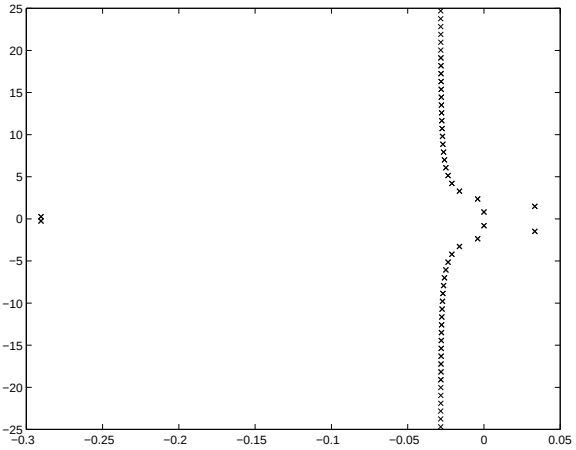


FIGURE 3. Zeros of $a + bz + z^2 - hz^2e^{-\tau z}$ where $a = 1$, $b = 0.5$, $h = -0.82465048736655$ and $\tau = 6.74469732735569$.

state solution is also a Hopf bifurcation point. The authors have transformed the computed zeros into the Floquet multipliers of the corresponding emanating periodic solution. This has enabled them to compare these "exact" Floquet multipliers with the approximations obtained via their approach. ♢

NOTE. As our algorithm not only obtains approximations for the zeros but also the corresponding multiplicities, we can use the modified Newton's method

$$z_k^{(\alpha+1)} = z_k^{(\alpha)} - \nu_k \frac{f(z_k^{(\alpha)})}{f'(z_k^{(\alpha)})}, \qquad \alpha = 0, 1, 2, \ldots,$$

to refine the approximations for the zeros. The modified Newton's method, Van de Vel's related iteration and our multidimensional generalizations of these methods will be discussed in detail in Chapter 7.

More numerical examples will be given in Section 5 and also in Chapter 2.

5. The software package ZEAL

We have implemented the algorithm that we have presented in Section 3 for the case that the curve γ is a rectangle whose edges are parallel to the coordinate axes. Our package is called ZEAL ('ZEros of AnaLytic functions') and is written in Fortran 90. Numerical approximations for the integrals along γ are computed via the quadrature package QUADPACK [**241**].

ZEAL's user interface is inspired by that of our package ZEBEC [**185**], which we will present in Chapter 5. Once the user has specified the rectangle γ, the analytic function f, its first derivative f' and the value of the parameter M (i.e., the number of zeros that are to be calculated simultaneously, cf. our discussion of the method of Delves and Lyness in Section 1), he/she can ask ZEAL to compute only the total number of zeros of f that lie inside γ, to isolate subregions of the interior of γ that contain at most M zeros, to compute all the zeros of f that lie inside γ or to compute

only a specified number of zeros (together with their respective multiplicities). The results can be written to separate files, etc. All these options will be discussed in detail below.

5.1. The approach taken by ZEAL. Given a rectangle whose edges are parallel to the coordinate axes and a positive integer M, we take the following approach:

- We calculate the total number of zeros that lie inside this rectangle.
- Via consecutive subdivisions we obtain a set of subrectangles, each of which contains at most M zeros (counting multiplicities).
- For each of these subrectangles, we calculate approximations for the zeros that lie inside it, together with their respective multiplicities.
- The approximations for the zeros are refined iteratively via the modified Newton's method.

As the function f may have zeros on the boundary of the rectangular box specified by the user, ZEAL starts by perturbing this box. For this purpose a tolerance is used that is taken to be proportional to a power of the machine precision, for example 10 times the square root of the machine precision. The box is then slightly enlarged in an asymmetrical way. The reason for this asymmetric perturbation is to lower the possibility of having a zero close to or on any boundary of the consecutive subdivisions (see also the footnote below). For example, if the starting box is symmetric with respect to the imaginary axis, then the inner boundary at the first subdivision will pass through any imaginary zeros of f.

The total number of zeros of f that lie inside the perturbed box is obtained in the same way as in our package ZEBEC. The real part of the integral in (12) is written as a sum of four integrals, where each integral corresponds to one of the edges of the rectangular region. Details will be given in Chapter 5. Approximations for these integrals are calculated via the adaptive integrator DQAG from QUADPACK. A zero near one of the edges of the rectangle causes the integrand of the corresponding integral to have a "peak." The closer the zero lies to the edge, the sharper this peak is. If the zero lies on the edge, then the integral is divergent. DQAG uses adaptive strategies that enable it to cope with such peaks efficiently. However, if a zero lies too close to an edge (the corresponding peak is too sharp), then DQAG warns us that it has problems in calculating the integral. Our algorithm then slightly moves this edge and restarts. By enlarging the user's box, we may of course include additional zeros. We have decided not to discard any of these zeros ourselves. Rather, we provide the user with the box that eventually has been considered, all the zeros that lie inside this box, and leave it to him/her to filter out unwanted zeros.

If the starting box (as perturbed by ZEAL) contains less than M zeros (counting multiplicities), then approximations for these zeros and the values of their respective multiplicities are computed via our implementation of the algorithm that we have presented in Section 3. Otherwise, the longest edges of the box are halved and the box is subdivided into two equal boxes. The number of zeros in each of these boxes is calculated via numerical integration. If DQAG detects a zero near the inner edge, then this edge is shifted, a process that results in an asymmetric subdivision of the

box.[1] Then the two smaller boxes are examined. A box that does not contain any zero is abandoned. If a box contains less than M zeros, then approximations for these zeros are calculated, together with their respective multiplicities. A box that contains more than M zeros is subdivided again. This process is repeated until a set of boxes has been found, each of which contains at most M zeros, and approximations for all these zeros as well as the values of their respective multiplicities have been computed. The approximations for the zeros are then refined via the modified Newton's method, which takes into account the multiplicity of a zero and converges quadratically.

5.2. The structure of ZEAL. The package ZEAL contains about 6500 lines of code including comments. It is written in Fortran 90 and has been thoroughly tested on various UNIX machines (HP 9000 B160L, IBM RS6000 7012 and SUN SPARC Ultra-2 m1170).

ZEAL consists of 8 files, namely, the main program `Main` and the modules

- `ZEAL_Module`, which contains the main subroutine `ZEAL` and also the subroutine `CHECK_INPUT`,
- `Zeros_Module`, which contains the subroutines `APPROXIMATE` and `INPROD`,
- `Refine_Module`, which contains the subroutines `REFINE` and `NEWTON`,
- `Split`, which contains the subroutines `INBOX` and `SPLITBOX`,
- and finally the modules `Input`, `Error` and `Quad`.

ZEAL also requires the subroutine DQAG from the package QUADPACK [**241**] and a number of subroutines from the BLAS and LAPACK libraries [**12**].

The user can specify the values of the various input parameters by editing the module `Input`. This will be discussed in Subsection 5.3.

The main program `Main` has the following form:

```
PROGRAM Main
  USE Input
  USE ZEAL_Module
  IMPLICIT NONE

  INTEGER :: TOTALNUMBER, DISTINCTNUMBER, REFINEDNUMBER
  INTEGER, DIMENSION(:), POINTER          :: MULTIPLICITIES
  LOGICAL, DIMENSION(:), POINTER          :: REFINEMENT_OK
  COMPLEX(KIND=DP), DIMENSION(:), POINTER :: ZEROS, FZEROS

  CALL ZEAL(TOTALNUMBER,DISTINCTNUMBER,ZEROS,FZEROS,       &
            MULTIPLICITIES,REFINEDNUMBER,REFINEMENT_OK)
END PROGRAM Main
```

[1] QUADPACK uses certain heuristic strategies. They work very well but nevertheless can fail. For example, we have observed that in case the function has several zeros very close (say, at a distance less than 10^{-7}) to the boundary, then DQAG may compute the wrong total number of zeros without giving any warning message. Therefore, we advise the user to make sure (as much as possible) that the zeros are not symmetrical with respect to the boundary. In particular, as many functions have zeros on the coordinate axes, we suggest to consider boxes that are asymmetrical with respect to these axes.

`Main` uses the information specified by the user in `Input` and calls the main subroutine `ZEAL`. This subroutine returns the total number of zeros of the given function that lie inside the given rectangular region, the number of mutually distinct zeros, the refined approximations for the zeros and the values that the function takes at these points, the corresponding multiplicities, the number of approximations for the zeros (as computed by the subroutine `APPROXIMATE`) that ZEAL has been able to refine successfully via the modified Newton's method, and finally for each computed zero a logical variable that indicates whether this refinement procedure has been successful or not.

In the design of ZEAL we have followed the recommendations for precision level maintenance described by Buckley [**44**]. The parameter `DP` that appears in the declaration of the variables `ZEROS` and `FZEROS` is defined in `Input`,

```
INTEGER, PARAMETER :: DP = SELECTED_REAL_KIND(15,70)
```

It determines the precision to which all the floating point calculations are to be done. On the computers that we have used, its current value corresponds to Fortran 77's `DOUBLE PRECISION`.[2]

Let us briefly describe the various parts of ZEAL.

The subroutine `INBOX` calculates the total number of zeros that lie inside the rectangular box given by the user. If some of the zeros lie too close to the boundary of this box and the quadrature routine DQAG fails, then `INBOX` perturbs the box slightly and enlarges it.

The subroutine `SPLITBOX` takes a box and splits it into two boxes. A symmetric splitting, which proceeds by halving the longest edges, is tried first. If the calculation of the integral along the inner edge fails, then it is assumed that some of the zeros lie too close to this edge and the inner edge is shifted.

The subroutine `APPROXIMATE` contains our implementation of the algorithm that we have presented in Section 3. The symmetric bilinear form (23) is evaluated via the subroutine `INPROD`.

The subroutine `NEWTON` contains our implementation of the modified Newton's method. The subroutine `REFINE` calls `NEWTON` to refine the approximations for the zeros that `APPROXIMATE` has computed. If `NEWTON` fails, then `REFINE` tries again from a nearby point. If after eight attempts `NEWTON` still fails, then `REFINE` indicates that it has been unable to refine the given approximation successfully.

The subroutine `ZEAL` forms the main part of the package. `ZEAL` starts by calling `CHECK_INPUT` to check if the input parameters specified by the user are proper. Next it calls `INBOX`. If there are no zeros inside the user's box, then the program stops. If there are less than M zeros inside the box (where the value of M can be specified in `Input`), then `APPROXIMATE` and `REFINE` are called. Else, the box is given

---

[2] This observation is important for the following reason. As documented in its `makefile`, ZEAL uses certain Fortran 77 routines from QUADPACK, BLAS and LAPACK. To enable the user to compile the necessary routines from BLAS and LAPACK in case these libraries are not available on his/her computer system, we have included them with our distribution of ZEAL. However, we have included only the `DOUBLE PRECISION` version of these routines and hence they should be replaced by the corresponding `SINGLE PRECISION` routines in case a change to `DP` requires this. The same holds for the subroutine DQAG from QUADPACK.

to SPLITBOX. The two boxes returned by SPLITBOX are examined. A box that does not contain any zero is abandoned. A box that contains less than M zeros is given to APPROXIMATE and REFINE. A box that contains more than M zeros is put in a list. Then ZEAL takes the next box from this list and calls SPLITBOX. This procedure is repeated until all the zeros have been computed, together with their respective multiplicities.

The program execution terminates normally after the completion of its task. This type of termination is indicated by the value 1 of the variable INFO, which is a global variable declared in the module Error. If the value of this parameter is different from 1, then the termination of the program is abnormal. The cases of abnormal termination are the following:

INFO=0: Improper input parameters.

INFO=2: The procedure for the calculation of the total number of zeros has failed.

INFO=3: The procedure for the isolation of the zeros has failed.

INFO=4: The procedure for the computation of the zeros has failed.

This concludes our discussion of the structure of ZEAL.

5.3. ZEAL's user interface. In the module Input the following parameters have to be set:

LV: a real array of length 2 that contains the x- and y-coordinates of the left lower vertex of the rectangle that is to be examined.

H: a real array of length 2 that specifies the size of this rectangle along the x- and y-direction.

M: an integer that determines the maximum number of zeros (counting multiplicities) that are considered within a subrectangle. M has to be larger than the maximum of the multiplicities of the zeros.

ICON: an integer in $\{1, \ldots, 4\}$ that specifies which calculations are to be done:

1. calculation of the total number of zeros, only,
2. calculation of the total number of zeros and isolation of a set of subrectangles, each of which contains at most M zeros,
3. calculation of the total number of zeros and computation of all the zeros, together with their respective multiplicities,
4. calculation of the total number of zeros and computation of NR zeros, together with their respective multiplicities.

Note that if ICON=4, the user must also supply the desired number of zeros NR. In the other cases (ICON=1,2,3) a value of NR may be supplied but it will not be used by the package.

NUMABS: a real variable that determines the absolute accuracy to which the integrals that calculate the number of zeros are to be evaluated. In case NUMABS = 0.0_DP, only a relative criterion is used.

NUMREL: a real variable that determines the relative accuracy to which the integrals that calculate the number of zeros are to be evaluated. In case NUMREL = 0.0_DP, only an absolute criterion is used.

If NUMABS and NUMREL are both too small, then the numerical integration may be time-consuming. If they are both too large, then the calculated number of zeros may be wrong. The default values of NUMABS and NUMREL are 0.07_DP and 0.0_DP, respectively.

INTABS: a real variable that determines the absolute accuracy to which the integrals that are used to compute approximations for the zeros are to be calculated. If INTABS = 0.0_DP, then only a relative criterion will be used.

INTREL: a real variable that determines the relative accuracy to which the integrals that are used to compute approximations for the zeros are to be calculated. If INTREL = 0.0_DP, then only an absolute criterion will be used.

If INTABS and INTREL are both too small, then the numerical integration may be time-consuming. If they are both too large, then the approximations for the zeros may be very inaccurate and Newton's method, which is used to refine these approximations (see NEWTONZ and NEWTONF), may fail. The default values of INTABS and INTREL are 0.0_DP and 1.0E-12_DP, respectively.

EPS_STOP: a real variable that is used in the stopping criterion that determines the number of mutually distinct zeros. If EPS_STOP is too large, then the computed number of mutually distinct zeros may be too small. If EPS_STOP is too small, then the computed number of mutually distinct zeros may be too large, especially in case the function has many multiple zeros. A recommended value is 1.0E-08_DP.

NEWTONZ **and** NEWTONF**:** these real variables should be specified in case ICON = 3 or 4. They are used as follows. The modified Newton's method, which takes into account the multiplicity of a zero and converges quadratically, is used to refine the calculated approximations for the zeros. The iteration stops if the relative distance between two successive approximations is at most NEWTONZ or the absolute value of the function at the last approximation is at most NEWTONF or if a maximum number of iterations (say, 20) is exceeded.

VERBOSE: a logical variable. ZEAL is allowed to print information (concerning the user's input and the computed results) if and only if VERBOSE is equal to .TRUE.

FILES: a logical variable. If FILES is set equal to .TRUE. then ZEAL generates the files zeros.dat and mult.dat. They contain the computed approximations for the zeros as well as their respective multiplicities. ZEAL also writes the file fzeros.dat, which contains the values that the function takes at the computed approximations for the zeros.

IFAIL: an integer that determines how errors are to be handled. We follow the NAG convention:

- 1. *soft silent error*—control is returned to the calling program.
- -1. *soft noisy error*—an error message is printed and control is returned to the calling program.
- 0. *hard noisy error*—an error message is printed and the program is stopped.

In the module `Input`, the user also has to specify the function f whose zeros ZEAL has to compute as well as its first derivative f'. This is done via the subroutine `FDF`, which takes the following form:

```
SUBROUTINE FDF(Z,F,DF)

COMPLEX(KIND=DP), INTENT(IN)   :: Z
COMPLEX(KIND=DP), INTENT(OUT)  :: F, DF

F = ...
DF = ...

END SUBROUTINE FDF
```

The user also has to specify the logical function `VALREG`. Given a rectangular region specified by its left lower vertex and the sizes of its edges, `VALREG` decides whether the function f is analytic inside this region or not. `VALREG` has the following form:

```
FUNCTION VALREG(LV,H)

LOGICAL VALREG
REAL(KIND=DP), INTENT(IN) :: LV(2), H(2)

VALREG = ...

END FUNCTION VALREG
```

For example, if f is analytic in the entire complex plane, then one may use the statement

```
VALREG = .TRUE.
```

If f has a branch cut along the non-positive real axis, then one may write

```
VALREG = .NOT. ( LV(2)*(LV(2)+H(2)) <= 0.0_DP .AND.   &
                 LV(1) <= 0.0_DP )
```

This concludes our discussion of ZEAL's user interface.

5.4. A few examples of how to use ZEAL. We will now discuss a few numerical examples.

EXAMPLE 8. Suppose that $f(z) = e^{3z} + 2z\cos z - 1$ and that

$$W = \{\, z \in \mathbb{C} \,:\, -2 \le \operatorname{Re} z \le 2, \quad -2 \le \operatorname{Im} z \le 3 \,\}.$$

In other words, W is the rectangular region $[-2,2] \times [-2,3]$. Therefore, we have to define the input parameters `LV` and `H` as

`LV = (/-2.0_DP,-2.0_DP/)` and `H = (/ 4.0_DP, 5.0_DP/)`.

We set `M = 5`. The logical variables `VERBOSE` and `FILES` are set to `.TRUE.`. We start by calculating only the total number of zeros, `ICON = 1`. ZEAL outputs the following.

```
This is ZEAL. Version of October 1998.

Input:

LV     =    -2.00000000000000      -2.00000000000000
H      =     4.00000000000000       5.00000000000000

M      =    5
ICON   =    1

FILES  =    T

Results:

The following box has been considered:

  LV =  -2.00000016391277       -2.00000019371510
  H  =   4.00000035762787        5.00000041723251

Total number of zeros inside this box         =     4
```

The function has four zeros inside the given box. We now ask ZEAL to compute approximations for all these zeros, `ICON=3`.

```
This is ZEAL. Version of October 1998.

Input:

LV     =    -2.00000000000000      -2.00000000000000
H      =     4.00000000000000       5.00000000000000

M      =    5
ICON   =    3

FILES  =    T

Results:

The following box has been considered:

  LV =  -2.00000016391277       -2.00000019371510
  H  =   4.00000035762787        5.00000041723251

Total number of zeros inside this box         =     4

Number of boxes containing at most    5 zeros =     1

These boxes are given by:
```

```
  1)  LV =  -2.00000016391277       -2.00000019371510
      H  =   4.00000035762787        5.00000041723251
      Total number of zeros inside this box =     4

 Final approximations for the zeros and verification:

  1)  Number of mutually distinct zeros     =     4

      z    = ( -1.84423395326221    ,-0.729696337329436E-29 )
      f(z) = ( 0.222044604925031E-15, 0.297690930716218E-28 )
      multiplicity =     1

      z    = ( 0.530894930292930    ,  1.33179187675112     )
      f(z) = ( 0.888178419700125E-15, 0.222044604925031E-14 )
      multiplicity =     1

      z    = ( 0.530894930292930    , -1.33179187675112     )
      f(z) = (-0.266453525910038E-14,-0.444089209850063E-15 )
      multiplicity =     1

      z    = ( 0.277555756299546E-16, 0.732694008769276E-26 )
      f(z) = (  0.00000000000000    , 0.366347004384638E-25 )
      multiplicity =     1
```

If we set `M = 2`, then ZEAL outputs the following.

```
 This is ZEAL. Version of October 1998.

 Input:

 LV     =    -2.00000000000000      -2.00000000000000
 H      =     4.00000000000000       5.00000000000000

 M      =    2
 ICON   =    3

 FILES  =    T

 Results:

 The following box has been considered:

   LV =  -2.00000016391277       -2.00000019371510
   H  =   4.00000035762787        5.00000041723251

 Total number of zeros inside this box         =     4
```

```
   Number of boxes containing at most    2 zeros =     3

   These boxes are given by:

     1)  LV =  -2.00000016391277       0.500000014901161
         H  =   4.00000035762787        2.50000020861626
         Total number of zeros inside this box =     1

     2)  LV =  -2.00000016391277       -2.00000019371510
         H  =   2.00000017881393        2.50000020861626
         Total number of zeros inside this box =     2

     3)  LV =  0.149011611938477E-07   -2.00000019371510
         H  =   2.00000017881393        2.50000020861626
         Total number of zeros inside this box =     1

   Final approximations for the zeros and verification:

     1)  Number of mutually distinct zeros     =     1

         z    = ( 0.530894930292931    ,  1.33179187675112     )
         f(z) = ( 0.888178419700125E-15,-0.177635683940025E-14 )
         multiplicity =     1

     2)  Number of mutually distinct zeros     =     2

         z    = ( -1.84423395326221    ,-0.551251254781237E-21 )
         f(z) = ( 0.222044604925031E-15, 0.224891493487409E-20 )
         multiplicity =     1

         z    = (-0.501336236251204E-20,-0.135361644895767E-20 )
         f(z) = (  0.00000000000000    ,-0.676808224478837E-20 )
         multiplicity =     1

     3)  Number of mutually distinct zeros     =     1

         z    = ( 0.530894930292931    , -1.33179187675112     )
         f(z) = ( 0.888178419700125E-15,-0.444089209850063E-15 )
         multiplicity =     1
```

Finally, suppose that we want ZEAL to compute only two zeros. We set `ICON=4` and `NR=2`.

```
   This is ZEAL. Version of October 1998.

   Input:

   LV     =    -2.00000000000000      -2.00000000000000
```

```
H      =     4.00000000000000       5.00000000000000

M      =    2
NR     =    2
ICON   =    4

FILES  =    T

Results:

The following box has been considered:

  LV =  -2.00000016391277       -2.00000019371510
  H  =   4.00000035762787        5.00000041723251

Total number of zeros inside this box         =     4

Number of boxes containing at most    2 zeros =     3

These boxes are given by:

  1)  LV =  -2.00000016391277       0.500000014901161
      H  =   4.00000035762787        2.50000020861626
      Total number of zeros inside this box =     1

  2)  LV =  -2.00000016391277       -2.00000019371510
      H  =   2.00000017881393        2.50000020861626
      Total number of zeros inside this box =     2

  3)  LV =  0.149011611938477E-07   -2.00000019371510
      H  =   2.00000017881393        2.50000020861626
      Total number of zeros inside this box =     1

Requested number of mutually distinct zeros =     2

Final approximations for the zeros and verification:

  1)  Number of mutually distinct zeros     =     1

      z    = ( 0.530894930292931    , 1.33179187675112      )
      f(z) = ( 0.888178419700125E-15,-0.177635683940025E-14 )
      multiplicity =     1

  2)  Number of mutually distinct zeros     =     2

      z    = ( -1.84423395326221    ,-0.551251254781237E-21 )
```

```
      f(z) = ( 0.222044604925031E-15, 0.224891493487409E-20 )
      multiplicity =     1
```

EXAMPLE 9. Suppose that $f(z) = z^2(z-1)(z-2)(z-3)(z-4) + z\sin z$ and let W be the rectangular region determined by

`LV = (/-0.5_DP,-0.5_DP/)` and `H = (/ 6.0_DP, 2.0_DP/)`.

Note that f has a double zero at the origin. We set `M=5` and `ICON=3`.

```
This is ZEAL. Version of October 1998.

Input:

LV     =   -0.500000000000000     -0.500000000000000
H      =     6.00000000000000       2.00000000000000

M      =    5
ICON   =    3

FILES  =    T

Results:

The following box has been considered:

  LV = -0.500000163912773      -0.500000193715096
  H  =   6.00000035762787        2.00000041723251

Total number of zeros inside this box         =     6

Number of boxes containing at most    5 zeros =     2

These boxes are given by:

  1)  LV = -0.500000163912773      -0.500000193715096
      H  =   3.00000017881393        2.00000041723251
      Total number of zeros inside this box =     4

  2)  LV =   2.50000001490116      -0.500000193715096
      H  =   3.00000017881393        2.00000041723251
      Total number of zeros inside this box =     2

Final approximations for the zeros and verification:

  1)  Number of mutually distinct zeros     =     3

      z    = (-0.444089209850063E-15,-0.284397851396988E-16 )
      f(z) = ( 0.491016012316152E-29, 0.631490085549722E-30 )
```

```
      multiplicity =     2

      z    = (  1.18906588973011    , 0.840925724965599E-27 )
      f(z) = (  0.00000000000000    ,-0.332912397884017E-26 )
      multiplicity =     1

      z    = (  1.72843498616506    ,-0.366031460022549E-27 )
      f(z) = (-0.222044604925031E-15,-0.174867254641548E-26 )
      multiplicity =     1

  2)  Number of mutually distinct zeros     =     2

      z    = (  4.03038191606047    , 0.288920306537196E-28 )
      f(z) = ( 0.155431223447522E-13, 0.308650229769291E-26 )
      multiplicity =     1

      z    = (  3.01990732809571    , 0.185382312726938E-28 )
      f(z) = (-0.105471187339390E-14,-0.402275402644635E-27 )
      multiplicity =     1
```

EXAMPLE 10. Finally, suppose that $f(z) = z^2(z-2)^2[\, e^{2z}\, \cos z + z^3 - 1 - \sin z\,]$ and let W be the region determined by

`LV = (/-1.0_DP,-1.0_DP/)` and `H = (/ 4.0_DP, 2.0_DP/)`.

Note that f has a triple zero at the origin and a double zero at $z = 2$. We set `M=5` and `ICON=3`.

```
This is ZEAL. Version of October 1998.

Input:

LV     =    -1.00000000000000      -1.00000000000000
H      =     4.00000000000000       2.00000000000000

M      =    5
ICON   =    3

FILES  =    T

Results:

The following box has been considered:

  LV =  -1.00000016391277       -1.00000019371510
  H  =   4.00000035762787        2.00000041723251

Total number of zeros inside this box         =     8
```

```
Number of boxes containing at most    5 zeros =     2

These boxes are given by:

  1)  LV =  -1.00000016391277       -1.00000019371510
      H  =   2.00000017881393        2.00000041723251
      Total number of zeros inside this box =     5

  2)  LV =   1.00000001490116       -1.00000019371510
      H  =   2.00000017881393        2.00000041723251
      Total number of zeros inside this box =     3

Final approximations for the zeros and verification:

  1)  Number of mutually distinct zeros     =     3

      z    = (-0.460714119728971    , 0.625427769347768     )
      f(z) = (-0.125408855493498E-14,-0.242634956938915E-14 )
      multiplicity =     1

      z    = (-0.555111512312578E-16, 0.780756160460674E-15 )
      f(z) = ( 0.270707981407843E-45,-0.189411036718915E-44 )
      multiplicity =     3

      z    = (-0.460714119728972    ,-0.625427769347767     )
      f(z) = ( 0.420110659916658E-14, 0.555338820267851E-14 )
      multiplicity =     1

  2)  Number of mutually distinct zeros     =     2

      z    = (  2.00000000000000    , 0.105626554996077E-14 )
      f(z) = (-0.253754972900138E-27,-0.312032161709746E-27 )
      multiplicity =     2

      z    = (  1.66468286974552    ,-0.528536806498078E-28 )
      f(z) = ( 0.276741773088933E-15, 0.405540193714869E-27 )
      multiplicity =     1
```

5.5. Concluding remarks. We have applied our package to various analytic functions and rectangular regions and we have found that it behaves predictably and accurately. ZEAL calculates the total number of zeros that lie inside the given box and then computes approximations for these zeros, together with their respective multiplicities. Our package does not require initial approximations for the zeros.

The user will appreciate the flexibility offered by the input parameter `ICON`. If nothing is known about the zeros that lie inside the given box, one may call ZEAL with `ICON = 1` to obtain the total number of zeros. Then one may proceed with `ICON = 3` to compute approximations for all these zeros, or, if less than the total

number of zeros are required, with `ICON = 4` and `NR` equal to the requested number of zeros. If only a set of boxes is required, each of which contains less than `M` zeros (counting multiplicities), then one may set `ICON = 2`.

6. A derivative-free approach

The results presented in the previous sections are based on the form (23), which involves the logarithmic derivative f'/f. Instead, let us consider the symmetric bilinear form

$$\langle \cdot, \cdot \rangle_\star \, : \, \mathcal{P} \times \mathcal{P} \to \mathbb{C}$$

defined as

$$\langle \phi, \psi \rangle_\star := \frac{1}{2\pi i} \int_\gamma \phi(z)\psi(z) \frac{1}{f(z)} \, dz \tag{37}$$

for any two polynomials $\phi, \psi \in \mathcal{P}$. Again, this form can be evaluated via numerical integration along γ and in what follows we will assume that all the "inner products" $\langle \phi, \psi \rangle_\star$ that are needed have been calculated.

We will show that essentially the same results can be obtained with the form $\langle \cdot, \cdot \rangle_\star$ as previously with $\langle \cdot, \cdot \rangle$. Note that the derivative f' is no longer needed. Of course, in this new approach not the mutually distinct zeros but rather the unknowns $Z_1, \ldots, Z_N$ (introduced by Delves and Lyness, cf. Section 1) are calculated and the multiplicities cannot be computed explicitely. But apart from this, the approach has the same advantages as the algorithm (which we will henceforth call 'our Algorithm') that we have presented in Section 3. In particular, it is self-starting in the sense that it does not require initial approximations for the zeros.

As by assumption the derivative f' is not available to us, we cannot obtain the value of N by evaluating the integral in the right-hand side of (14) numerically. Instead one can use the principle of the argument, cf. our discussion in Subsection 1.1.

The integrand that appears in the right-hand side of (37) has a pole at every zero of f that lies in the interior of γ and the order of the pole is equal to the multiplicity of the zero. Therefore, the residue theorem implies that $\langle \phi, \psi \rangle_\star$ is equal to the sum of the residues of the function $\phi\psi/f$ at these poles. The following result can easily be verified.

PROPOSITION 11. *Suppose that all the N zeros $Z_1, \ldots, Z_N$ of f that lie inside γ are simple. Then*

$$\langle \phi, \psi \rangle_\star = \sum_{k=1}^{N} \frac{\phi(Z_k)\psi(Z_k)}{f'(Z_k)}.$$

In general, if f has multiple zeros, then an elegant expression for $\langle \phi, \psi \rangle_\star$ written as a sum is much more difficult to obtain. Fortunately, it is not necessary to have such an expression available. The proofs of Theorems 3 and 4 depend completely on the details of the way in which $\langle \phi, \psi \rangle$ can be written as a sum, cf. Equation (23). However, as we will see, the corresponding theorems can also be proved in a different way.

Define $s_p^\star := \langle 1, z^p \rangle_\star$ for $p = 0, 1, 2, \ldots$ and let $H_k^\star$ be the $k \times k$ Hankel matrix

$$H_k^\star := \left[s_{p+q}^\star \right]_{p,q=0}^{k-1}$$

for $k = 1, 2, \ldots$. The formal orthogonal polynomials associated with $\langle \cdot, \cdot \rangle_\star$ can be defined as before. The coefficients of regular FOPs can be computed by solving a Yule-Walker system, cf. Equation (25). Also, $t \geq 1$ is a regular index if and only if the matrix $H_t^\star$ is nonsingular.

The residue theorem immediately implies that the polynomial

$$P_N(z) = \prod_{k=1}^{N} (z - Z_k)$$

satisfies

$$\langle z^p, P_N(z) \rangle_\star = 0, \qquad p = 0, 1, 2, \ldots . \tag{38}$$

In this sense, the polynomial $P_N(z)$ behaves with respect to the form $\langle \cdot, \cdot \rangle_\star$ in the same way as the polynomial $\varphi_n(z)$ behaves with respect to $\langle \cdot, \cdot \rangle$, cf. Equations (27) and (28). We will prove that N is the largest regular index for $\langle \cdot, \cdot \rangle_\star$. This will enable us to compute the zeros of the regular FOP $P_N(z)$, i.e., the zeros $Z_1, \ldots, Z_N$, in essentially the same way as our Algorithm applied to the form $\langle \cdot, \cdot \rangle$ computes the zeros of $\varphi_n(z)$, i.e., the mutually distinct zeros $z_1, \ldots, z_n$.

The following lemma will play an important role. Define the set $\mathcal{I}$ as follows:

$$\mathcal{I} := \{ \phi \in \mathcal{P} \ : \ \langle z^p, \phi(z) \rangle_\star = 0 \quad \text{for } p = 0, 1, 2, \ldots \}.$$

LEMMA 12. *The set $\mathcal{I}$ is equal to the ideal generated by the polynomial P_N. In other words,*

$$\mathcal{I} = \{ \phi \in \mathcal{P} \ : \ \exists \alpha \in \mathcal{P} \ : \ \phi = \alpha P_N \}.$$

PROOF. Suppose that $a \in \mathbb{C}$ lies in the interior of γ. Let the function $g : W \to \mathbb{C}$ be meromorphic and suppose that g has neither zeros nor poles on γ. Then the coefficient of $(z-a)^{-p-1}$ in the Laurent expansion of g at the point a is given by the integral

$$\frac{1}{2\pi i} \int_\gamma (z-a)^p \, g(z) \, dz$$

for $p = 0, 1, 2, \ldots$. Let $\phi \in \mathcal{I}$. Then

$$\langle (z - Z_k)^p, \phi(z) \rangle_\star = \frac{1}{2\pi i} \int_\gamma (z - Z_k)^p \frac{\phi(z)}{f(z)} \, dz = 0$$

for $k = 1, \ldots, N$ and $p = 0, 1, 2, \ldots$ and thus the function ϕ / f has a removable singularity at the points $Z_1, \ldots, Z_N$. Thus ϕ has to be a multiple of P_N. This proves the lemma. □

THEOREM 13. *The matrix $H_N^\star$ is nonsingular.*

PROOF. We will prove that P_N is the only monic polynomial of degree N that is orthogonal to all polynomials of lower degree. Suppose that Q_N is another such polynomial. Then $P_N - Q_N$ is of degree at most $N-1$ and hence $\langle P_N - Q_N, Q_N \rangle_\star = 0$. Equation (38) then implies that $\langle Q_N, Q_N \rangle_\star = 0$. Thus Q_N is not only orthogonal

to all polynomials of degree $\leq N-1$ but also to all polynomials of degree N. The polynomial $zP_N - zQ_N$ has degree $\leq N$ and therefore $\langle zP_N - zQ_N, Q_N\rangle_\star = 0$. As $\langle zP_N, Q_N\rangle_\star = \langle P_N, zQ_N\rangle_\star = 0$, it follows that $\langle zQ_N, Q_N\rangle_\star = 0$. Thus Q_N is also orthogonal to all polynomials of degree $N+1$. By continuing this way, one can prove that Q_N is orthogonal to *all* polynomials, $Q_N \in \mathcal{I}$. As Q_N is a monic polynomial of degree N, the previous lemma then implies that $Q_N = P_N$. Thus there is only one monic polynomial of degree N that is orthogonal to all polynomials of lower degree. This implies that the matrix $H_N^\star$ is nonsingular. □

THEOREM 14. *The matrix $H_{N+k}^\star$ is singular for $k = 1, 2, \ldots$.*

PROOF. Instead of the basis of the monomials $\{z^p\}_{p\geq 0}$ we consider the basis $\{\psi_p(z)\}_{p\geq 0}$ where $\psi_p(z) := z^p$ for $p = 0, 1, \ldots, N-1$ and $\psi_{N+p}(z) := z^p P_N(z)$ for $p = 0, 1, 2, \ldots$. Let

$$F_l^\star := \Big[\langle \psi_p, \psi_q\rangle_\star\Big]_{p,q=0}^{l-1}$$

be the corresponding $l \times l$ Gram matrix for $l = 1, 2, \ldots$. Equation (38) then implies that $\det F_{N+k}^\star = 0$ for $k = 1, 2, \ldots$. One can easily verify that $\det F_l^\star = \det H_l^\star$ for $l = 1, 2, \ldots$. This proves the theorem. □

We have now identified $P_N(z)$ as the regular FOP of degree N and we have shown that regular FOPs of degree larger than N do not exist. Note that $s_0^\star$ is equal to the sum of the residues of $1/f$ at the points $Z_1, \ldots, Z_N$ and hence it is not necessarily different from zero. Therefore, the regular FOP of degree 1 with respect to the form $\langle\cdot,\cdot\rangle_\star$ doesn't always exist, in contrast to $\langle\cdot,\cdot\rangle$.

The zero/eigenvalue properties discussed in Section 3 hold not only for $\langle\cdot,\cdot\rangle$ but for every symmetric bilinear form. The zeros $Z_1, \ldots, Z_N$ can therefore also be calculated by solving a generalized eigenvalue problem. The following result can be proved in the same way as Theorem 4. Let $H_k^{\star<}$ be the Hankel matrix

$$H_k^{\star<} := \begin{bmatrix} s_1^\star & s_2^\star & \cdots & s_k^\star \\ s_2^\star & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_k^\star & \cdots & \cdots & s_{2k-1}^\star \end{bmatrix}$$

for $k = 1, 2, \ldots$.

THEOREM 15. *The eigenvalues of the pencil $H_N^{\star<} - \lambda H_N^\star$ are given by $Z_1, \ldots, Z_N$.*

Below we will compare the accuracy obtained via Theorem 4 to the accuracy obtained via Theorem 15.

The zeros $Z_1, \ldots, Z_N$ can also be computed by applying our Algorithm to the form $\langle\cdot,\cdot\rangle_\star$. Let $\{\varphi_t^\star\}_{t\geq 0}$ denote the FOPs associated with $\langle\cdot,\cdot\rangle_\star$. Define the matrices $G_k^\star$ and $G_k^{\star z}$ as

$$G_k^\star := \Big[\langle \varphi_r^\star, \varphi_s^\star\rangle\Big]_{r,s=0}^{k-1} \qquad \text{and} \qquad G_k^{\star z} := \Big[\langle \varphi_r^\star, z\varphi_s^\star\rangle\Big]_{r,s=0}^{k-1}$$

for $k = 1, 2, \ldots$. The following results can be proved in the same way as Theorem 7 and Corollary 8.

THEOREM 16. *Let $t \geq 1$ be a regular index for $\langle \cdot, \cdot \rangle_\star$ and let $z^\star_{t,1}, \ldots, z^\star_{t,t}$ be the zeros of the regular FOP $\varphi^\star_t$. Then the eigenvalues of the pencil $G^{\star z}_t - \lambda G^\star_t$ are given by $z^\star_{t,1}, \ldots, z^\star_{t,t}$.*

COROLLARY 17. *The eigenvalues of $G^{\star z}_N - \lambda G^\star_N$ are given by $Z_1, \ldots, Z_N$.*

If instead of N only an upper bound for N is available, then the value of N can be computed via the stopping criterion of our Algorithm.

We will now discuss a few numerical examples. The computations have been done via Matlab 5 (with floating point relative accuracy $\approx 2.2204\,10^{-16}$). The integration algorithm is the same as the one discussed at the beginning of Section 4.

EXAMPLE 11. Let $f(z) = e^{3z} - 2z\cos z - 1$. Suppose that γ is the circle $\gamma = \{ z \in \mathbb{C} : |z| = 4 \}$. Then $N = 6$. Let us try an approach based on ordinary moments. Table 1 contains approximations for $s_p = \langle 1, z^p \rangle$ and $s^\star_p = \langle 1, z^p \rangle_\star$ for $p = 0, 1, \ldots, 11$. Note that in both cases the order of magnitude changes as p increases. The computed approximations for the zeros $Z_1, \ldots, Z_N$ obtained via

p	s_p	$s^\star_p$
0	6.0	$-7.5\,10^{-2}$
1	2.0	$2.8\,10^{-1}$
2	$-1.4\,10^{1}$	$-9.1\,10^{-1}$
3	$-8.5\,10^{1}$	2.1
4	$-3.0\,10^{1}$	-2.0
5	$7.0\,10^{2}$	2.4
6	$2.5\,10^{3}$	$-2.3\,10^{1}$
7	$-1.0\,10^{3}$	$1.5\,101$
8	$-3.1\,10^{4}$	$9.9\,10^{1}$
9	$-7.6\,10^{4}$	$4.6\,10^{2}$
10	$1.3\,10^{5}$	$-4.8\,10^{2}$
11	$1.2\,10^{6}$	$-5.3\,10^{3}$

TABLE 1. Ordinary moments s_p and $s^\star_p$

Theorem 4 and 15 are shown in Table 2 and 3, respectively. The digits that are not correct are underlined. Observe that the approximations for the zeros are very

$-2.1860794911758\underline{28}\,10^{-13}$	$-$	$i\,1.\underline{72762308312215}3\,10^{-12}$
$5.308949302929\underline{420}\,10^{-1}$	$+$	$i\,1.33179187675\underline{0615}$
$5.308949302928\underline{376}\,10^{-1}$	$-$	$i\,1.33179187675\underline{1221}$
$-1.844233953262\underline{199}$	$-$	$i\,\underline{4.2044941520423}\underline{17}\,10^{-14}$
$1.41460717765\underline{8190}$	$+$	$i\,3.0477220626271\underline{69}$
$1.41460717765\underline{8185}$	$-$	$i\,3.047722062627173$

TABLE 2. Approximations for the zeros obtained via the ordinary moments s_p.

$\underline{5.8791984865934}49 10^{-14}$	$+$	$i\,\underline{1.8968367263982}49 10^{-14}$
$5.30894930292936\underline{6} 10^{-1}$	$+$	$i\,1.33179187675106\underline{6}$
$5.30894930292920\underline{5} 10^{-1}$	$-$	$i\,1.33179187675108\underline{0}$
-1.844233953262213	$-$	$i\,\underline{1.2311963474258}26 10^{-15}$
$1.41460717765818\underline{1}$	$+$	$i\,3.04772206262716\underline{6}$
$1.41460717765818\underline{0}$	$-$	$i\,3.04772206262716\underline{7}$

TABLE 3. Approximations for the zeros obtained via the ordinary moments $s_p^\star$.

accurate. Using ordinary moments has the advantage that only $2N$ integrals have to be calculated and hence, compared to our Algorithm, the arithmetic cost is rather limited. Also, a significant part of the computation required for each integrand is the same for all of the integrands (namely, the computation of f'/f or $1/f$). By programming the quadrature algorithm in such a way that it is able to integrate a *vector* of similar integrals, these common calculations need be done only once for each integrand evaluation point. However, as the following example shows, ordinary moments do not always lead to such accurate results.

EXAMPLE 12. The Wilkinson polynomial and also functions that have clusters of zeros are typical, although somewhat extreme, examples where an approach based on ordinary moments is likely to fail. The following function is another example. Suppose that $f(z) = J_0(z)$, the Bessel function of the first kind and of order zero. It is known that this function has only positive real zeros and that all these zeros are simple (see, e.g., Watson [**295**] and also Chapter 5). In that sense it is related to the Wilkinson polynomial. Suppose that $\gamma = \{\, z \in \mathbb{C} : |z - 15| = 14.5 \,\}$. Then $N = 9$. Table 4 gives for each zero the number of correct significant digits obtained via the ordinary moments $s_p^\star$ (Theorem 15), our Algorithm applied to the form $\langle \cdot, \cdot \rangle_\star$, the ordinary moments s_p (Theorem 4) and our Algorithm applied to the form $\langle \cdot, \cdot \rangle$.

exact zeros	$s_p^\star$	$\langle \cdot, \cdot \rangle_\star$	s_p	$\langle \cdot, \cdot \rangle$
2.404825557695773	5	12	6	12
5.520078110286311	2	11	4	10
8.653727912911013	2	10	4	9
11.79153443901428	3	11	5	9
14.93091770848778	3	11	5	9
18.07106396791092	3	11	4	10
21.21163662987926	4	11	5	11
24.35247153074930	5	11	6	11
27.49347913204025	7	11	7	12

TABLE 4. The number of correct significant digits in case $f(z) = J_0(z)$.

Observe that in both cases the approximations obtained via our Algorithm are more accurate than the approximations obtained via ordinary moments. Of course, there is clearly a trade-off between obtained accuracy and cost. We advise the reader to start

with the cheapest approach, i.e., the approach based on the ordinary moments $s_p^\star$. If the computed approximations for the zeros are not sufficiently accurate to be refined via an iterative method (one that doesn't need the derivative, of course), then one can apply our Algorithm to the form $\langle \cdot, \cdot \rangle_\star$ or switch to one of the approaches that use both f and f'.

EXAMPLE 13. Let us illustrate how the stopping criterion of our Algorithm can be used to determine the value of N in case only an upper bound for N is known. Consider again the function $f(z) = e^{3z} - 2z\cos z - 1$ and suppose that γ is the circle $\gamma = \{ z \in \mathbb{C} : |z| = 5 \}$. Then $N = 7$. Let us assume that only the upper bound 20 is known. Our algorithm defines the FOP $\varphi_1^\star$ as an inner polynomial and $\varphi_2^\star$ as a regular FOP. At this point the algorithm asks itself whether N is equal to two. It computes $|\langle \varphi_2^\star, \varphi_2^\star \rangle_\star|$. To take into account the accuracy lost during the evaluation of the quadrature formula, this quantity is scaled in a certain way, cf. Section 3. The resulting floating point number is given by

$$1.998545018990362,$$

which is certainly not "sufficiently small" (we use 10^{-8} as a threshold) and hence the algorithm continues. It defines $\varphi_3^\star$ as an inner polynomial and $\varphi_4^\star$ as a regular FOP. Then it checks if N is equal to four. It compares

$$1.981687581683116$$

to 10^{-8} and continues. The polynomial $\varphi_5^\star$ is defined as a regular FOP. The algorithm again decides to continue and defines $\varphi_6^\star$ as a regular FOP. The corresponding floating point number is given by

$$0.3794164188056766$$

and the algorithm continues. It defines $\varphi_7^\star$ as a regular FOP. We have now reached the actual value of N. The scaled counterparts of the inner products that correspond to the sequence (34) are given by

$$\begin{aligned}
&9.190814944765118\,10^{-16}\\
&1.799485800789563\,10^{-15}\\
&4.008700446099430\,10^{-15}\\
&5.548436809880727\,10^{-15}\\
&6.603511781861113\,10^{-15}\\
&5.538342691314587\,10^{-15}\\
&3.494634208761963\,10^{-15}\\
&4.116380174988637\,10^{-16}\\
&5.379567405602837\,10^{-15}\\
&8.806423950940129\,10^{-15}\\
&8.912210606016112\,10^{-15}\\
&5.866528582137192\,10^{-15}\\
&1.127215921207880\,10^{-15}
\end{aligned}$$

and hence the algorithm decides that N is equal to seven and it stops. The computed approximations for the zeros are given by

$$
\begin{array}{rcl}
-2.212860324230451\,10^{-11} & + & i\,5.610894531592185\,10^{-12} \\
5.308949303037738\,10^{-1} & + & i\,1.331791876751059 \\
5.308949303027991\,10^{-1} & - & i\,1.331791876755293 \\
-1.844233953258748 & - & i\,1.244550552500599\,10^{-12} \\
1.414607177657119 & + & i\,3.047722062626751 \\
1.414607177657241 & - & i\,3.047722062626826 \\
-4.603562881675490 & + & i\,3.443757237606488\,10^{-14}
\end{array}
$$

The correct significant digits are underlined. Let us now compare this with the approach based on ordinary moments. The following theorem generalizes Theorem 15.

THEOREM 18. *Let t be an integer $\geq N$. The eigenvalues of the pencil $H_t^{\star <} - \lambda H_t^{\star}$ are given by the zeros $Z_1, \ldots, Z_N$ and $t-N$ eigenvalues that may assume arbitrary values.*

PROOF. Instead of the basis of the monomials $\{z^p\}_{p\geq 0}$ we consider again the basis $\{\psi_p(z)\}_{p\geq 0}$ where $\psi_p(z) := z^p$ for $p = 0, 1, \ldots, N-1$ and $\psi_{N+p}(z) := z^p P_N(z)$ for $p = 0, 1, 2, \ldots$, cf. the proof of Theorem 14. Define

$$
F_t^{\star <} := \Big[\langle \psi_p, z\psi_q \rangle_{\star} \Big]_{p,q=0}^{t-1} \qquad \text{and} \qquad F_t^{\star} := \Big[\langle \psi_p, \psi_q \rangle_{\star} \Big]_{p,q=0}^{t-1}.
$$

Then one can easily show that the generalized eigenvalue problem $H_t^{\star <} x = \lambda H_t^{\star} x$ is equivalent to the problem $F_t^{\star <} y = \lambda F_t^{\star} y$. Here $y := U_t^{-1} x$ where U_t denotes the unit upper triangular matrix that contains the coefficients (in the standard monomial basis) of the polynomials $\psi_0(z), \psi_1(z), \ldots, \psi_{t-1}(z)$. Equation (38) then implies that

$$
F_t^{\star <} = \begin{bmatrix} H_N^{\star <} & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad F_t^{\star} = \begin{bmatrix} H_N^{\star} & 0 \\ 0 & 0 \end{bmatrix}.
$$

This proves the theorem. □

Each of the indeterminate generalized eigenvalues mentioned in the previous theorem corresponds to two corresponding zeros on the diagonals of the generalized Schur decomposition of the Hankel matrices $H_t^{\star <}$ and $H_t^{\star}$. When actually calculated, these diagonal entries are different from zero because of roundoff errors and Matlab returns their quotient as an eigenvalue. Thus, by solving the 20×20 generalized eigenvalue problem $H_{20}^{\star <} - \lambda H_{20}^{\star}$ we obtain approximations for the seven zeros $Z_1, \ldots, Z_N$ and 13 spurious eigenvalues. The latter can be detected by evaluating f at the computed eigenvalues and also by taking into account that the computed approximations for the zeros are likely to lie inside γ or at least quite close to it. The approximations for the zeros obtained in this way are 1 to 3 digits less accurate than the approximations obtained via our Algorithm. By solving the 7×7 generalized eigenvalue problem, one obtains approximations that are about as accurate as those computed by our Algorithm.

EXAMPLE 14. Let us consider a function that has multiple zeros. Suppose that

$$
f(z) = z^2 (z-2)^2 [\, e^{2z} \cos z + z^3 - 1 - \sin z \,]
$$

and let $\gamma = \{z \in \mathbb{C} : |z| = 3\}$. Note that f has a triple zero at the origin and a double zero at $z = 2$. The total number of zeros of f that lie inside γ is equal to eight, $N = 8$. By using the ordinary moments $s_p^{\star}$ we obtain the following approximations for the zeros:

$$
\begin{array}{rcl}
1.183531315599526\,10^{-4} & - & i\,8.8406481378441 01\,10^{-7} \\
-5.994094794302300\,10^{-5} & - & i\,1.0205224454414 14\,10^{-4} \\
-5.841218335364184\,10^{-5} & + & i\,1.0293630934607 68\,10^{-4} \\
2.000000113260292 & + & i\,9.2537274020093 06\,10^{-7} \\
1.999999886743732 & - & i\,9.2537367883827 85\,10^{-7} \\
-4.607141197285995\,10^{-1} & + & i\,6.2542776934713 80\,10^{-1} \\
-4.607141197287246\,10^{-1} & - & i\,6.2542776934728 81\,10^{-1} \\
1.664682869740608 & + & i\,1.0933074552212 65\,10^{-12}
\end{array}
$$

We have underlined the correct significant digits. Our Algorithm gives comparable results. Note how the obtained accuracy diminishes as the multiplicity of the zero increases.

Acknowledgements

The results in Section 3 and 6 were obtained in collaboration with Marc Van Barel.

Norman Katz sent me the pages of [**306**] that are related to Theorem 2 whereas Nikolaos Ioakimidis gave me reprints of more than 25 papers of his. I would also like to thank James Lyness and Tien-Yien Li for stimulating conversations and Koen Engelborghs for giving me the permission to include Figure 3.

The first version of the package ZEAL was written during my second research stay (March 22–April 18, 1998) at the Department of Mathematics, University of Patras (Patras, Greece). I would like to thank Michael Vrahatis, Omiros Ragos and Filareti Zafiropoulos for their very kind hospitality and for giving me the opportunity to collaborate with them. I would also like to thank Vlaamse Leergangen Leuven for its generous financial support.

I thank Bernard Mourrain for the stimulating discussions that initiated the results presented in Section 6.

CHAPTER 2

Clusters of zeros of analytic functions

In the previous chapter we have presented an accurate algorithm, based on the theory of formal orthogonal polynomials, for computing zeros of analytic functions. More specifically, given an analytic function f and a Jordan curve γ that does not pass through any zero of f, we have considered the problem of computing *all* the zeros $z_1, \ldots, z_n$ of f that lie inside γ, together with their respective multiplicities $\nu_1, \ldots, \nu_n$. Our principal means of obtaining information about the location of these zeros has been the symmetric bilinear form $\langle \cdot, \cdot \rangle$, cf. Equation (23). This form can be evaluated via numerical integration along γ.

This chapter continues the previous chapter. If f has one or several clusters of zeros, then the mapping from the ordinary moments associated with $\langle \cdot, \cdot \rangle$ to the zeros and their respective multiplicities is very ill-conditioned. We will show that the algorithm that we have presented in Chapter 1 can be used to calculate the centre of a cluster and its size, i.e., the arithmetic mean of the zeros that form a certain cluster and the total number of zeros in this cluster, respectively. This information enables one to zoom into a certain cluster: its zeros can be calculated separately from the other zeros of f. By shifting the origin in the complex plane to the centre of a certain cluster, its zeros become better relatively separated, which is appropriate in floating point arithmetic and reduces the ill-conditioning.

In this chapter we will also attack our problem of computing all the zeros of f that lie inside γ in an entirely different way, based on rational interpolation at roots of unity. We will show how the new approach complements the previous one and how it can be used effectively in case γ is the unit circle.

NOTE. Specifically for clusters of polynomial zeros, let us mention that Hribernig and Stetter [**149**] worked on detection and validation of clusters of zeros whereas Kirrinnis [**178**] studied Newton's iteration towards a cluster.

This chapter corresponds to part of our paper [**186**].

1. How to obtain the centre of a cluster and its weight

Suppose that the zeros of f that lie inside γ can be grouped into m clusters. Let $I_1, \ldots, I_m$ be index sets that define these clusters, and let

$$\mu_j := \sum_{k \in I_j} \nu_k \qquad \text{and} \qquad c_j := \frac{1}{\mu_j} \sum_{k \in I_j} \nu_k z_k$$

for $j = 1, \ldots, m$. In other words, μ_j is equal to the total number of zeros that form cluster j (its "weight") whereas c_j is equal to the arithmetic mean of the zeros

in cluster j (its "centre of gravity"). We assume that the centres $c_1, \ldots, c_m$ are mutually distinct. For $k = 1, \ldots, n$ we also define $\zeta_k := z_k - c_j$ if $k \in I_j$. From the definition of μ_j and c_j it follows that

$$\sum_{k \in I_j} \nu_k \zeta_k = 0, \qquad j = 1, \ldots, m.$$

Define the symmetric bilinear form $\langle \cdot, \cdot \rangle_m$ by

$$\langle \phi, \psi \rangle_m := \sum_{j=1}^{m} \mu_j \phi(c_j) \psi(c_j)$$

for any two polynomials $\phi, \psi \in \mathcal{P}$. This form is related to the form $\langle \cdot, \cdot \rangle$ in an obvious way: instead of the zeros $z_1, \ldots, z_n$ and their multiplicities $\nu_1, \ldots, \nu_n$, we now use the centres of gravity $c_1, \ldots, c_m$ and the weights $\mu_1, \ldots, \mu_m$ of the clusters. Let

$$\delta := \max_{1 \le k \le n} |\zeta_k|.$$

The following theorem tells us that $\langle \cdot, \cdot \rangle_m$ approximates $\langle \cdot, \cdot \rangle$ (and vice versa).

THEOREM 19. *Let $\phi, \psi \in \mathcal{P}$. Then $\langle \phi, \psi \rangle = \langle \phi, \psi \rangle_m + \mathcal{O}(\delta^2)$, $\delta \to 0$.*

PROOF. The following holds:

$$\begin{aligned}
\langle \phi, \psi \rangle &= \sum_{k=1}^{n} \nu_k \phi(z_k) \psi(z_k) \\
&= \sum_{j=1}^{m} \sum_{k \in I_j} \nu_k \phi(c_j + \zeta_k) \psi(c_j + \zeta_k) \\
&= \sum_{j=1}^{m} \sum_{k \in I_j} \nu_k \big(\phi(c_j)\psi(c_j) + \zeta_k \, [\phi(z)\psi(z)]'_{z=c_j} + \mathcal{O}(\zeta_k^2), \ \zeta_k \to 0 \big) \\
&= \sum_{j=1}^{m} \mu_j \phi(c_j) \psi(c_j) + \sum_{j=1}^{m} \Big(\underbrace{\sum_{k \in I_j} \nu_k \zeta_k}_{=0} \Big) [\phi(z)\psi(z)]'_{z=c_j} + \sum_{k=1}^{n} \mathcal{O}(\zeta_k^2), \ \zeta_k \to 0 \\
&= \sum_{j=1}^{m} \mu_j \phi(c_j) \psi(c_j) + \sum_{k=1}^{n} \mathcal{O}(\zeta_k^2), \ \zeta_k \to 0.
\end{aligned}$$

This proves the theorem. □

Define the ordinary moments associated with $\langle \cdot, \cdot \rangle_m$ as

$$s_p^{(m)} := \langle 1, z^p \rangle_m$$

for $p = 0, 1, 2, \ldots$. Observe that $s_0^{(m)} = s_0$ whereas $s_1^{(m)} = s_1$. Define the vectors $\mathbf{s}, \mathbf{s}^{(m)} \in \mathbb{C}^{2N-1}$ as

$$\mathbf{s} := \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{2N-2} \end{bmatrix} \qquad \text{and} \qquad \mathbf{s}^{(m)} := \begin{bmatrix} s_0^{(m)} \\ s_1^{(m)} \\ \vdots \\ s_{2N-2}^{(m)} \end{bmatrix}.$$

The entries of $\mathbf{s}$ determine the Hankel matrix H_N. The previous theorem implies that

$$\frac{\|\mathbf{s} - \mathbf{s}^{(m)}\|_2}{\|\mathbf{s}\|_2} = \mathcal{O}(\delta^2), \ \delta \to 0. \tag{39}$$

Let $H_k^{(m)}$ be the $k \times k$ Hankel matrix

$$H_k^{(m)} := \left[\ s_{p+q}^{(m)} \ \right]_{p,q=0}^{k-1}$$

for $k = 1, 2, \ldots$.

COROLLARY 20. *Let $k \geq 1$. Then* $\det H_k = \det H_k^{(m)} + \mathcal{O}(\delta^2), \ \delta \to 0$.

PROOF. Let k be a positive integer. Then the previous theorem implies that

$$H_k = \left[\ s_{p+q} \ \right]_{p,q=0}^{k-1} = \left[\ s_{p+q}^{(m)} + \mathcal{O}(\delta^2), \ \delta \to 0 \ \right]_{p,q=0}^{k-1}.$$

The result follows by expanding the determinant of the matrix in the right-hand side. □

COROLLARY 21. *The matrix H_m is nonsingular if $\delta \to 0$. Let $t > m$. Then* $\det H_t = \mathcal{O}(\delta^2), \ \delta \to 0$.

PROOF. This follows from the previous corollary and the fact that $H_m^{(m)}$ is nonsingular whereas $H_t^{(m)}$ is singular for all integers $t > m$ (cf. Theorem 3). □

The following theorem should be compared with Theorem 9.

THEOREM 22. *Let t be an integer $\geq m$. Then $\varphi_t(c_j) = \mathcal{O}(\delta^2), \ \delta \to 0$ for $j = 1, \ldots, m$. Also $\langle z^p, \varphi_t(z) \rangle = \mathcal{O}(\delta^2), \ \delta \to 0$ for all $p \geq t$.*

PROOF. Let $t \geq m$. If t is a regular index, then

$$\langle z^p, \varphi_t(z) \rangle = 0, \qquad p = 0, 1, \ldots, t-1,$$

else

$$\langle z^p, \varphi_t(z) \rangle = 0, \qquad p = 0, 1, \ldots, r-1$$

where r is the largest regular index less than t. Corollary 21 implies that $r \geq m$, and thus we may conclude that

$$\langle z^p, \varphi_t(z) \rangle = 0, \qquad p = 0, 1, \ldots, m-1.$$

Theorem 19 then implies that

$$\langle z^p, \varphi_t(z) \rangle_m = \mathcal{O}(\delta^2), \ \delta \to 0, \qquad p = 0, 1, \ldots, m-1.$$

In matrix notation this can be written as

$$\begin{bmatrix} 1 & \cdots & 1 \\ c_1 & \cdots & c_m \\ \vdots & & \vdots \\ c_1^{m-1} & \cdots & c_m^{m-1} \end{bmatrix} \begin{bmatrix} \mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_m \end{bmatrix} \begin{bmatrix} \varphi_t(c_1) \\ \varphi_t(c_2) \\ \vdots \\ \varphi_t(c_m) \end{bmatrix} = \mathcal{O}(\delta^2),\ \delta \to 0.$$

The right-hand side represents a vector in $\mathbb{C}^m$ whose entries are $\mathcal{O}(\delta^2)$, $\delta \to 0$. As the centres $c_1, \ldots, c_m$ are assumed to be mutually distinct and the weights $\mu_1, \ldots, \mu_m$ are different from zero, it follows that

$$\varphi_t(c_j) = \mathcal{O}(\delta^2),\ \delta \to 0, \qquad j = 1, \ldots, m.$$

Theorem 19 then immediately implies that

$$\langle z^p, \varphi_t(z) \rangle = \mathcal{O}(\delta^2),\ \delta \to 0$$

for all $p \geq t$. □

In other words, unless the FOP $\varphi_t(z)$ has a very flat shape near its zeros, we are likely to find good approximations for the centres $c_1, \ldots, c_m$ among the zeros of $\varphi_t(z)$ for all $t \geq m$. Note that

$$\begin{bmatrix} 1 & \cdots & 1 \\ c_1 & \cdots & c_m \\ \vdots & & \vdots \\ c_1^{m-1} & \cdots & c_m^{m-1} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{m-1} \end{bmatrix} + \mathcal{O}(\delta^2),\ \delta \to 0.$$

It follows that approximations for the weights $\mu_1, \ldots, \mu_m$ can be obtained by solving a Vandermonde system.

What happens if we apply the algorithm that we have presented in Chapter 1 in case the zeros of f can be grouped into clusters? The second part of Theorem 22 implies that our algorithm stops at $r = m$ if δ, the maximal size of the clusters, is sufficiently small. It returns the zeros of the FOP $\varphi_m(z)$ that is associated with $\langle \cdot, \cdot \rangle$. Theorem 19 and the fact that the mth degree FOP with respect to $\langle \cdot, \cdot \rangle_m$ is given by $\prod_{j=1}^{m}(z - c_j)$ imply that we can use these zeros as approximations for the centres of the clusters. (This also follows from the first part of Theorem 22, of course.) The computed approximations for the weights of the clusters should be close to integers. We can check this to verify that we have indeed determined the correct value of m. We can also calculate (approximations for) the ordinary moments associated with $\langle \cdot, \cdot \rangle_m$ and verify if (39) is satisfied.

2. A numerical example

In the following example we have again considered the case that γ is a circle, cf. Section 4 of Chapter 1. The computations have been done via Matlab 5.

Stewart's perturbation theory for the generalized eigenvalue problem [**259**] allows us to make a sensitivity analysis. An important result from his first order perturbation theory for simple eigenvalues tells us the following. If λ is a simple

eigenvalue of the pencil $G_t^{(1)} - \lambda G_t$ and λ_ϵ is the corresponding eigenvalue of a perturbed pencil $\tilde{G}_t^{(1)} - \lambda \tilde{G}_t$ with $\|G_t^{(1)} - \tilde{G}_t^{(1)}\|_2 \approx \|G_t - \tilde{G}_t\|_2 \approx \epsilon$, then

$$\frac{|\lambda - \lambda_\epsilon|}{\sqrt{1+|\lambda|^2}\sqrt{1+|\lambda_\epsilon|^2}} \leq \frac{\epsilon}{\sqrt{|y^H G_t^{(1)} x|^2 + |y^H G_t x|^2}} + \mathcal{O}(\epsilon^2) =: \kappa(\lambda, x, y)\epsilon + \mathcal{O}(\epsilon^2)$$

where x and y are the right and left eigenvectors corresponding to λ,

$$G_t^{(1)} x = \lambda G_t \qquad \text{and} \qquad y^H G_t^{(1)} = \lambda y^H G_t,$$

normalized such that $\|x\|_2 = \|y\|_2 = 1$. Let us call $\kappa(\lambda, x, y)$ the *sensitivity factor* of the eigenvalue λ.

EXAMPLE 15. Suppose that $n = 10$,

$$z_1 = -1,$$
$$z_2 = 4, \quad z_3 = 4 + \delta(1+i),$$
$$z_4 = 3i, \quad z_5 = 3i + \delta(10+5i), \quad z_6 = 3i + \delta(-3+4i),$$
$$z_7 = c + \delta(-1+2i), \quad z_8 = c + \delta(1+5i), \quad z_9 = c + \delta(1+i), \quad z_{10} = c + \delta(-2-2i),$$

where $c = -3 + 3i$ and $\delta = 10^{-4}$. Suppose that $\nu_1 = \cdots = \nu_{10} = 1$. Let $f(z)$ be the polynomial that has $z_1, \ldots, z_{10}$ as simple zeros, $f(z) = \prod_{k=1}^{10}(z - z_k)$, and let $\gamma = \{\, z \in \mathbb{C} : |z| = 5 \,\}$. Note that f has four clusters of zeros, $m = 4$, of weight 1, 2, 3 and 4, respectively. We have evaluated the logarithmic derivative of $f(z)$ again via formula (36). As we do not want the algorithm to stop as soon as it has found the approximations for the centres of the clusters, we set ϵ_{stop} to a rather small value, $\epsilon_{\text{stop}} = 10^{-18}$. Our algorithm gives the following results. The total number of zeros is equal to 10. The polynomials $\varphi_0(z)$ and $\varphi_1(z)$ are defined as regular FOPs. The computed eigenvalues of the pencil $G_2^{(1)} - \lambda G_2$ lead to zeros that lie inside γ, and thus the polynomial $\varphi_2(z)$ is defined as a regular FOP. The sensitivity factors of the eigenvalues are equal to

```
3.773765047881042e-02
9.474367914383353e-03
```

The solution of the Vandermonde system that corresponds to the approximations for the zeros of $\varphi_2(z)$ is given by

```
7.050510110011339e+00 - i  2.941868186347331e-01
2.949489889988664e+00 + i  2.941868186347336e-01
```

The algorithm computes $\langle \varphi_2(z), \varphi_2(z) \rangle$. In step [**4**] it compares

```
9.853283227226082e-01
```

with ϵ_{stop} and sets allsmall $\leftarrow$ **false**. The polynomial $\varphi_3(z)$ is defined as a regular FOP. The sensitivity factors of the eigenvalues are equal to

```
5.221220847969128e-03
5.691118489264763e-02
9.440130958262826e-03
```

The solution of the Vandermonde system that corresponds to the approximations for the zeros of $\varphi_3(z)$ is given by

```
1.751672615711886e+00 + i  2.391215777821213e-02
5.761474298212248e+00 + i  1.418723240124189e-01
2.486853086075870e+00 - i  1.657844817906288e-01
```

The algorithm computes $\langle \varphi_3(z), \varphi_3(z) \rangle$. In step [**4**] it compares

```
9.560608076054004e-02
```

with ϵ_{stop} and sets allsmall $\leftarrow$ **false**. The polynomial $\varphi_4(z)$ is defined as a regular FOP. The sensitivity factors of the eigenvalues are equal to

```
5.997609084856927e-03
1.632514220886046e-02
2.198497992029341e-02
4.923118129602375e-03
```

The solution of the Vandermonde system that corresponds to the approximations for the zeros of $\varphi_4(z)$ is given by

```
1.999999998746185e+00 - i  1.260274035855700e-08
4.000000263704519e+00 - i  1.428161173014011e-07
2.999999744825852e+00 + i  2.437473983804684e-07
9.999999927234455e-01 - i  8.832854039794525e-08
```

Observe that these "multiplicities" (actually, they are the weights of the clusters) are at a distance of $\mathcal{O}(10^{-8}) = \mathcal{O}(\delta^2)$ to integers. This is a first indication of the fact that $m = 4$. The algorithm computes $\langle \varphi_4(z), \varphi_4(z) \rangle$. In step [**4**] it compares

```
4.690246227384357e-09
```

with ϵ_{stop}. As we have given ϵ_{stop} a very small value, $\epsilon_{\text{stop}} = 10^{-18}$, the algorithm sets allsmall $\leftarrow$ **false** and continues. It defines the polynomial $\varphi_5(z)$ as a regular FOP. The sensitivity factors of the eigenvalues are equal to

```
1.859124283121160e+02
5.997664519337178e-03
1.632900762729697e-02
4.930300860498997e-03
2.201051194773096e-02
```

Observe that one of the eigenvalues is much more sensitive than the others. The solution of the Vandermonde system that corresponds to the computed approximations for the zeros of $\varphi_5(z)$ is given by

```
-4.028597029147776e-08 - i  1.692141150994100e-07
 2.000000007231077e+00 - i  8.308500282558695e-09
 4.000000510541128e+00 - i  2.651932814361151e-07
 1.000000192451618e+00 - i  8.614030726982243e-08
 2.999999330062146e+00 + i  5.288562050783197e-07
```

Observe that the component that corresponds to the spurious eigenvalue is of size $\mathcal{O}(10^{-8})$ whereas the other components are close to integers. This enables us to deduce the presence of spurious eigenvalues without computing the sensitivity factors. The algorithm computes $\langle \varphi_5(z), \varphi_5(z) \rangle$. In step [**4**] it compares

```
3.544154048709335e-10
```

with ϵ_{stop} and sets allsmall $\leftarrow$ **false**. The polynomial $\varphi_6(z)$ is defined as a regular FOP. The sensitivity factors of the eigenvalues are equal to

```
1.617034259272598e+02
1.220513781306331e+01
5.997660002011287e-03
1.633002045462034e-02
4.929209225445833e-03
2.200115675959345e-02
```

The solution of the Vandermonde system that corresponds to the approximations for the zeros of $\varphi_6(z)$ is given by

```
-1.109848867030467e-07 - i  3.954458493988473e-08
 5.068148299128812e-12 - i  6.003619075497872e-13
 2.000000007044562e+00 - i  9.500131431430428e-09
 4.000000455291260e+00 - i  3.668720544062149e-07
 1.000000170753607e+00 - i  1.185972639722307e-07
 2.999999477890392e+00 + i  5.345146309560269e-07
```

The algorithm computes $\langle \varphi_6(z), \varphi_6(z) \rangle$. In step [**4**] it compares

```
1.189012222825083e-09
```

with ϵ_{stop} and sets allsmall $\leftarrow$ **false**. And so on. The algorithm defines $\varphi_7(z)$ as a regular FOP, $\varphi_8(z)$ as an inner polynomial, $\varphi_9(z)$ as a regular FOP, and finally $\varphi_{10}(z)$ as an inner polynomial. The computed approximations for the zeros of f are given by

```
-9.999999999599027e-01 + i  1.827471507453993e-11
 4.000066008247924e+00 + i  5.012986226260452e-05
 4.000047628710319e+00 - i  2.622572865154105e-04
-5.868580001572310e-05 + i  3.000189455314092e+00
 1.147726665656712e-03 + i  3.000342939244789e+00
-3.000001882960546e+00 + i  3.000324337949845e+00
-3.000258740452308e+00 + i  2.999857462002724e+00
-9.085700394437596e-01 + i  1.866967988946470e+00
-8.509233181288104e-01 - i  6.684482268880147e+00
-4.999300000000002e-01 + i  2.100160000000000e+00
```

The relative errors of the approximations for the zeros that belong to the clusters of weight 1 and 2 are $\mathcal{O}(10^{-11})$ and $\mathcal{O}(10^{-5})$, resp. For the other zeros, the relative errors are at least $\mathcal{O}(10^{-3})$.

If we set $\epsilon_{\text{stop}} = 10^{-6}$, then our algorithm stops at the polynomial of degree 4. We obtain the following approximations for the centres of the clusters:

```
-9.999999564181510e-01 - i  5.152524762408461e-08
 4.000050001653271e+00 + i  5.000694739720757e-05
 2.335838430156945e-04 + i  3.000299920075392e+00
-3.000024926663507e+00 + i  3.000149946356108e+00
```

Let us now focus on the separate clusters. We have considered the circles whose centre is the computed approximation for the centre of a cluster and whose radius is equal to 0.1. The relative errors of the approximations for the zeros that we obtain are $\mathcal{O}(10^{-16})$, $\mathcal{O}(10^{-12})$, $\mathcal{O}(10^{-10})$ and $\mathcal{O}(10^{-6})$ for the clusters of weight 1, 2, 3 and 4, respectively. If we consider a circle whose centre is the computed approximation for the centre of the cluster of weight 4 and whose radius is equal to 10^{-3}, then

the relative errors of the approximations that we obtain for the zeros that lie in this cluster are $\mathcal{O}(10^{-16})$. Apparently, the smaller the radius of the circle is, the more accurate the computed approximations for the zeros are. This can be explained by the fact that the quadrature method gives more accurate approximations for the integrals (in other words, for the data from which approximations for the zeros are computed). $\diamondsuit$

More numerical examples will be given in Section 4.

3. Rational interpolation at roots of unity

We will now approach our problem of computing all the zeros of f that lie inside γ in a different way, based on rational interpolation at roots of unity. We will show how the new approach complements the previous one and how it can be used effectively in case γ is the unit circle. Numerical examples will be given in Section 4.

Let K be a positive integer and let $t_1, \ldots, t_K$ be the Kth roots of unity,

$$t_k := \exp\big(\frac{2\pi i}{K}k\big), \qquad k = 1, \ldots, K.$$

Define $g_{K-1}(z)$ as the polynomial

$$g_{K-1}(z) := s_0 z^{K-1} + s_1 z^{K-2} + \cdots + s_{K-1}.$$

Note that $\deg g_{K-1}(z) = K-1$ as by assumption $s_0 \neq 0$. Without loss of generality we may assume that $g_{K-1}(t_k) \neq 0$ for $k = 1, \ldots, K$. (This condition will be needed in Theorem 23.)

Let $w_K(z) := z^K - 1$ and define the symmetric bilinear form $\langle\langle \cdot, \cdot \rangle\rangle$ as

$$\langle\langle \phi, \psi \rangle\rangle := \sum_{k=1}^{K} \frac{g_{K-1}(t_k)}{w_K'(t_k)} \phi(t_k)\psi(t_k)$$

for $\phi, \psi \in \mathcal{P}$. Note that this form can be evaluated via FFT.

Define the ordinary moments σ_p associated with the form $\langle\langle \cdot, \cdot \rangle\rangle$ as

$$\sigma_p := \langle\langle 1, z^p \rangle\rangle = \sum_{k=1}^{K} \frac{g_{K-1}(t_k)}{w_K'(t_k)} t_k^p$$

for $p = 0, 1, 2, \ldots$ and let $\mathcal{H}_k$ be the $k \times k$ Hankel matrix

$$\mathcal{H}_k := \big[\ \sigma_{p+q}\ \big]_{p,q=0}^{k-1} = \begin{bmatrix} \sigma_0 & \sigma_1 & \cdots & \sigma_{k-1} \\ \sigma_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ \sigma_{k-1} & \cdots & \cdots & \sigma_{2k-2} \end{bmatrix}$$

for $k = 1, 2, \ldots$. Then the regular FOP f_τ of degree $\tau \geq 1$ associated with the form $\langle\langle \cdot, \cdot \rangle\rangle$ exists if and only if the matrix $\mathcal{H}_\tau$ is nonsingular. Also, the following theorem holds (cf. Theorem 3).

THEOREM 23. *$K = \operatorname{rank} \mathcal{H}_{K+p}$ for every nonnegative integer p.*

Thus $\mathcal{H}_K$ is nonsingular whereas $\mathcal{H}_\tau$ is singular for $\tau > K$. The regular FOP f_K of degree K exists while regular FOPs of degree larger than K do not exist. The polynomial f_K is easily seen to be

$$f_K(z) = (z - t_1) \cdot \cdots \cdot (z - t_K) = w_K(z).$$

It is the monic polynomial of degree K that has $t_1, \ldots, t_K$ as simple zeros.

If $\mathcal{H}_K$ is strongly nonsingular, then we have a full set $\{f_0, f_1, \ldots, f_K\}$ of regular FOPs. Else, we can proceed in the same way as with the form $\langle \cdot, \cdot \rangle$. By filling up the gaps in the sequence of existing regular FOPs it is possible to define a sequence $\{f_\tau\}_{\tau=0}^{\infty}$, with f_τ a monic polynomial of degree τ, such that if these polynomials are grouped into blocks according to the sequence of regular indices, then polynomials belonging to different blocks are orthogonal with respect to $\langle\langle \cdot, \cdot \rangle\rangle$. More precisely, define $\{f_\tau\}_{\tau=0}^{\infty}$ as follows. If τ is a regular index, then let f_τ be the regular FOP of degree τ. Else define f_τ as $f_\rho p_{\tau,\rho}$ where ρ is the largest regular index less than τ and $p_{\tau,\rho}$ is an arbitrary monic polynomial of degree $\tau - \rho$. In the latter case f_τ is called an *inner polynomial*. If $p_{\tau,\rho}(z) = z^{\tau-\rho}$ then we say that f_τ is defined *by using the standard monomial basis*. The block orthogonality property is expressed by the fact that the Gram matrix $[\langle\langle f_r, f_s \rangle\rangle]_{r,s=0}^{K-1}$ is block diagonal. The diagonal blocks are nonsingular, symmetric and zero above the main antidiagonal. If all the inner polynomials in a certain block are defined by using the standard monomial basis, then the corresponding diagonal block has Hankel structure, cf. Theorem 5.

The definition of the form $\langle\langle \cdot, \cdot \rangle\rangle$ may seem arbitrary. However, there exists a remarkable connection between the forms $\langle \cdot, \cdot \rangle$ and $\langle\langle \cdot, \cdot \rangle\rangle$.

THEOREM 24. *Let $\phi, \psi \in \mathcal{P}$. If $\deg \phi + \deg \psi \leq K - 1$, then $\langle\langle \phi, \psi \rangle\rangle = \langle \phi, \psi \rangle$.*

PROOF. Let $V(t_1, \ldots, t_K)$ be the Vandermonde matrix with nodes $t_1, \ldots, t_K$,

$$V(t_1, \ldots, t_K) := \begin{bmatrix} 1 & t_1 & \cdots & t_1^{K-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_K & \cdots & t_K^{K-1} \end{bmatrix}.$$

Then

$$\begin{bmatrix} g_{K-1}(t_1) \\ \vdots \\ g_{K-1}(t_K) \end{bmatrix} = V(t_1, \ldots, t_K) \begin{bmatrix} s_{K-1} \\ \vdots \\ s_0 \end{bmatrix}.$$

As $V(t_1, \ldots, t_K)/\sqrt{K}$ is unitary, it follows that

$$\begin{bmatrix} s_{K-1} \\ \vdots \\ s_0 \end{bmatrix} = \frac{1}{K} [V(t_1, \ldots, t_K)]^H \begin{bmatrix} g_{K-1}(t_1) \\ \vdots \\ g_{K-1}(t_K) \end{bmatrix}.$$

As $w_K(z) = z^K - 1$, it follows that $w_K'(z) = K z^{K-1}$ and thus $w_K'(t_k) = K/t_k$ for $k = 1, \ldots, K$. Let $j \in \{1, \ldots, K\}$. Then

$$\sigma_{K-j} = \frac{1}{K} \sum_{k=1}^{K} g_{K-1}(t_k) t_k^{K-j+1} = \frac{1}{K} \sum_{k=1}^{K} g_{K-1}(t_k) \overline{t_k^{j-1}}$$

and thus

$$\begin{bmatrix} \sigma_{K-1} \\ \sigma_{K-2} \\ \vdots \\ \sigma_0 \end{bmatrix} = \frac{1}{K} \overline{\begin{bmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_K \\ \vdots & & \vdots \\ t_1^{K-1} & \cdots & t_K^{K-1} \end{bmatrix}} \begin{bmatrix} g_{K-1}(t_1) \\ g_{K-1}(t_2) \\ \vdots \\ g_{K-1}(t_K) \end{bmatrix}$$

$$= \frac{1}{K} [V(t_1, \ldots, t_K)]^H \begin{bmatrix} g_{K-1}(t_1) \\ \vdots \\ g_{K-1}(t_K) \end{bmatrix}$$

$$= \begin{bmatrix} s_{K-1} \\ s_{K-2} \\ \vdots \\ s_0 \end{bmatrix}.$$

In other words, $s_p = \sigma_p$ for $p = 0, 1, \ldots, K-1$. As $\langle\langle \phi, \psi \rangle\rangle$ depends on σ_p for $p = 0, 1, \ldots, \deg(\phi\psi)$, this proves the theorem. □

COROLLARY 25. *Let τ be a nonnegative integer. If $2\tau - 1 \leq K$, then τ is a regular index for $\langle\langle \cdot, \cdot \rangle\rangle$ if and only if τ is a regular index for $\langle \cdot, \cdot \rangle$. Moreover, if $2\tau \leq K$ and if τ is a regular index, then $f_\tau(z) \equiv \varphi_\tau(z)$. Else, if τ is not a regular index, then $f_\tau(z) = R_{\tau,\rho}(z)\varphi_\tau(z)$ where ρ is the largest regular index less than τ and $R_{\tau,\rho}(z)$ is a rational function of type $[\tau - \rho / \tau - \rho]$. If $f_\tau(z)$ and $\varphi_\tau(z)$ are both defined by using the standard monomial basis, then $R_{\tau,\rho}(z) \equiv 1$.*

COROLLARY 26. *If $K \geq 2n$ and $n \leq \tau \leq \lfloor K/2 \rfloor$, then $f_\tau(z_k) = 0$ for $k = 1, \ldots, n$ and $\langle z^p, f_\tau(z) \rangle = 0$ for all $p \geq 0$. Also, $\langle\langle z^p, f_\tau(z) \rangle\rangle = 0$ for $p = \tau, \ldots, K - 1 - \tau$. (Note that the latter range may be empty.)*

COROLLARY 27. *If $K \geq 2m$ and $m \leq \tau \leq \lfloor K/2 \rfloor$, then $f_\tau(c_j) = \mathcal{O}(\delta^2)$, $\delta \to 0$ for $j = 1, \ldots, m$ and $\langle z^p, f_\tau(z) \rangle = \mathcal{O}(\delta^2)$, $\delta \to 0$ for all $p \geq \tau$. Also, $\langle\langle z^p, f_\tau(z) \rangle\rangle = \mathcal{O}(\delta^2)$, $\delta \to 0$ for $p = \tau, \ldots, K - 1 - \tau$. (Note that the latter range may be empty.)*

Thus, if $K \geq 2N$, then we can apply the algorithm that we have presented in Chapter 1 to the form $\langle\langle \cdot, \cdot \rangle\rangle$ and we will obtain exactly the same results as with the form $\langle \cdot, \cdot \rangle$. This is an interesting fact in its own right. The main reason, though, that motivated us to introduce the form $\langle\langle \cdot, \cdot \rangle\rangle$ is the fact that it is related to rational interpolation. We will show that the "denominator polynomials" in a certain linearized rational interpolation problem that is related to the polynomial $g_{K-1}(z)$ are FOPs with respect to $\langle\langle \cdot, \cdot \rangle\rangle$. This will lead to an alternative way to calculate the FOPs $f_\tau(z)$ and thus, because of Corollary 25, the FOPs $\varphi_\tau(z)$.

Let σ and τ be nonnegative integers such that $\sigma + \tau + 1 = K$. Let $p_\sigma(z)$ and $q_\tau(z)$ be polynomials, where

$$\deg p_\sigma(z) \leq \sigma \quad \text{and} \quad \deg q_\tau(z) \leq \tau, \tag{40}$$

such that the following linearized rational interpolation conditions are satisfied:

$$p_\sigma(t_k) - q_\tau(t_k) g_{K-1}(t_k) = 0, \qquad k = 1, \ldots, K. \tag{41}$$

Each pair of polynomials $(\,p_\sigma(z), q_\tau(z)\,)$ that satisfies the degree conditions (40) and the interpolation conditions (41) is called a *multipoint Padé form* (MPF). The polynomials $p_\sigma(z)$ and $q_\tau(z)$ will be called *numerator polynomial* and *denominator polynomial*, respectively.

The interpolation conditions (41) lead to a system of K linear equations in $K+1$ unknowns, and thus at least one nontrivial (i.e., whose numerator and denominator polynomial are not identically equal to zero) MPF exists. As (41) are homogeneous linear equations, every scalar multiple of a MPF is also a MPF. From now on, we will always assume that MPFs are normalized such that the denominator polynomial is monic. However, the fact that then the number of interpolation conditions is equal to the number of unknown polynomial coefficients, does not guarantee that there exists only one MPF. It merely guarantees that every MPF leads to the same irreducible rational function, called *multipoint Padé approximant* (MPA). Indeed, suppose that there exist two MPAs. Then the numerator polynomial of the difference of these MPAs is a polynomial of degree $\leq \sigma + \tau$ that vanishes at $\sigma + \tau + 1$ points. This numerator polynomial is therefore identically equal to zero, which implies that the MPA is unique.

Let $\mathcal{R}_{\sigma,\tau}$ be the set of rational functions of type $[\sigma/\tau]$, i.e., with numerator degree at most σ and denominator degree at most τ. A rational interpolation problem that is closely related to (41) is the *Cauchy interpolation problem*: find all irreducible rational functions $r_{\sigma,\tau}(z) \in \mathcal{R}_{\sigma,\tau}$ whose denominator polynomial is monic, such that

$$r_{\sigma,\tau}(t_k) = g_{K-1}(t_k), \qquad k = 1, \ldots, K. \tag{42}$$

This interpolation problem is not always solvable. If a solution exists, then it is unique, and it is equal to the MPA. In general, however, the MPA need not solve the Cauchy interpolation problem: the numerator and denominator polynomials of the MPFs may have common zeros at some interpolation points. The MPA may not satisfy the interpolation condition (42) at these points, which are then called *unattainable points*.

Let $r(z) \in \mathcal{R}_{\sigma,\tau}$ and suppose that $r(z) = p(z)/q(z)$ where $p(z)$ and $q(z)$ are relatively prime polynomials. The *defect* of r with respect to $\mathcal{R}_{\sigma,\tau}$ is then defined as

$$\min\{\,\sigma - \deg p(z), \tau - \deg q(z)\,\}.$$

The following theorem provides the general solution of the linearized rational interpolation problem.

THEOREM 28. *The general MPF that corresponds to the degree conditions* (40) *and the interpolation conditions* (41) *is given by*

$$\big(\,p_\sigma(z), q_\tau(z)\,\big) = \big(\,\hat{p}_\sigma(z)s(z)u(z), \hat{q}_\tau(z)s(z)u(z)\,\big),$$

where $\hat{p}_\sigma(z)$, $\hat{q}_\tau(z)$ and $s(z)$ are uniquely determined polynomials, and where $u(z)$ is arbitrary. The polynomials $\hat{p}_\sigma(z)$ and $\hat{q}_\tau(z)$ are relatively prime, and $s(z)$ is a divisor of $w_K(z)$. Let $\hat{\delta}_{\sigma,\tau}$ be the defect of $\hat{p}_\sigma(z)/\hat{q}_\tau(z)$ with respect to $\mathcal{R}_{\sigma,\tau}$. Then $\deg s(z) \leq \hat{\delta}_{\sigma,\tau}$ and $\deg u(z) \leq \hat{\delta}_{\sigma,\tau} - \deg s(z)$. The zeros of $s(z)$ are the unattainable points for the corresponding Cauchy interpolation problem.

PROOF. See, for example, Gutknecht [**124**, p. 549]. □

The literature on rational interpolation is vast. We will not give a full account of all the other issues (in particular, the block structure of the Newton-Padé table) that are involved. The reader may wish to consult the papers by Meinguet [**212**], Antoulas [**13, 14, 15**], Berrut and Mittelmann [**31**] or Gutknecht [**124, 125, 126, 127**], and the references cited therein.

What is of special interest to us, is the fact that the denominator polynomials $q_\tau(z)$ are formal orthogonal polynomials with respect to $\langle\langle\cdot,\cdot\rangle\rangle$.

THEOREM 29. *Let σ and τ be nonnegative integers such that $\sigma+\tau+1=K$. Let $(\,p_\sigma(z),q_\tau(z)\,)$ be a MPF for the degree conditions* (40) *and the interpolation conditions* (41). *Then $\langle\langle z^p,q_\tau(z)\rangle\rangle=0$ for $p=0,1,\ldots,K-2-\deg p_\sigma(z)$ and $\langle\langle z^p,q_\tau(z)\rangle\rangle\neq 0$ if $p=K-1-\deg p_\sigma(z)$.*

PROOF. Apparently this orthogonality relation was already known to Jacobi. As it plays a very important role in our approach, we prefer to give a (short, but explicit) proof. See also Eğecioğlu and Koç [**73**] and Gemignani [**100**] for a slightly weaker version of this theorem.

Let $p\in\{0,1,\ldots,K-2-\deg p_\sigma(z)\}$. Then

$$\sum_{k=1}^{K}\frac{t_k^p p_\sigma(t_k)}{w_K'(t_k)}=\sum_{k=1}^{K}\frac{g_{K-1}(t_k)}{w_K'(t_k)}t_k^p q_\tau(t_k) \tag{43}$$

and $z^p p_\sigma(z)$ is a polynomial of degree $p+\deg p_\sigma(z)\le K-2$. Lagrange's formula for the polynomial $y_{K-1}(z)$ of degree $\le K-1$ that interpolates the polynomial $z^p p_\sigma(z)$ in the points $t_1,\ldots,t_K$ implies that the left hand side of (43) is equal to the coefficient of z^{K-1} of $y_{K-1}(z)$. As $\deg[z^p p_\sigma(z)]<K-1$, it follows that $y_{K-1}(z)\equiv z^p p_\sigma(z)$ and that the coefficient of z^{K-1} of $y_{K-1}(z)$ is equal to zero. It follows that $\langle\langle z^p,q_\tau(z)\rangle\rangle=0$ for $p=0,1,\ldots,K-2-\deg p_\sigma(z)$. A similar reasoning shows that $\langle\langle z^p,q_\tau(z)\rangle\rangle\neq 0$ if $p=K-1-\deg p_\sigma(z)$. This proves the theorem. □

The following theorem implies that the coefficients (in the standard monomial basis) of the numerator polynomial $p_\sigma(z)$ of a MPF $\left(\,p_\sigma(z),q_\tau(z)\,\right)$ that corresponds to the degree conditions (40) and the interpolation conditions (41) can be expressed as inner products with respect to $\langle\langle\cdot,\cdot\rangle\rangle$. This explains how the degree property $\deg p_\sigma(z)\le\sigma$ is related to the formal orthogonality property satisfied by $q_\tau(z)$.

THEOREM 30. *Suppose that $q(z)$ is a polynomial and let $p(z)$ be the polynomial of degree $\le K-1$ that interpolates $g_{K-1}(z)q(z)$ at the points $t_1,\ldots,t_K$. Let $p(z)=:p_0+p_1z+\cdots+p_{K-1}z^{K-1}$. Then $p_k=\langle\langle z^{K-1-k},q(z)\rangle\rangle$ for $k=0,1,\ldots,K-1$.*

PROOF. The Lagrange representation of $p(z)$ is given by

$$p(z)=\sum_{k=1}^{K}\pi_k L_k(z)$$

where

$$\pi_k:=\frac{g_{K-1}(t_k)q(t_k)}{w_K'(t_k)}\quad\text{and}\quad L_k(z):=\frac{w_K(z)}{z-t_k}$$

for $k = 1, \ldots, K$. Let $L_k(z) =: L_{0,k} + L_{1,k} z + \cdots + L_{K-1,k} z^{K-1}$ for $k = 1, \ldots, K$. Note that $L_{K-1,1} = \cdots = L_{K-1,K} = 1$. Let

$$V := \begin{bmatrix} 1 & t_1 & \cdots & t_1^{K-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_K & \cdots & t_K^{K-1} \end{bmatrix}$$

be the Vandermonde matrix with nodes $t_1, \ldots, t_K$ and let

$$L := \begin{bmatrix} L_{0,1} & \cdots & L_{0,K} \\ \vdots & & \vdots \\ L_{K-1,1} & \cdots & L_{K-1,K} \end{bmatrix}$$

be the matrix that contains the coefficients of $L_1(z), \ldots, L_K(z)$. Then

$$\begin{aligned} VL &= \operatorname{diag}(L_1(t_1), \ldots, L_K(t_K)) \\ &= \operatorname{diag}(w'_K(t_1), \ldots, w'_K(t_K)) \\ &= K \operatorname{diag}(\overline{t_1}, \ldots, \overline{t_K}). \end{aligned}$$

As $V/\sqrt{K}$ is unitary, it follows that $V^{-1} = V^H/K$ and thus

$$L = V^H \operatorname{diag}(\overline{t_1}, \ldots, \overline{t_K}) = \begin{bmatrix} t_1^{K-1} & \cdots & t_K^{K-1} \\ \vdots & & \vdots \\ t_1 & \cdots & t_K \\ 1 & \cdots & 1 \end{bmatrix}.$$

In other words, $L_{j,k} = t_k^{K-1-j}$ for $k = 1, \ldots, K$ and $j = 0, 1, \ldots, K-1$. As $p_j = \sum_{k=1}^{K} L_{j,k} \pi_k$ for $j = 0, 1, \ldots, K-1$, it follows that

$$p_j = \sum_{k=1}^{K} \frac{g_{K-1}(t_k)}{w'_K(t_k)} t_k^{K-1-j} q(t_k) = \langle\langle z^{K-1-j}, q(z) \rangle\rangle$$

for $j = 0, 1, \ldots, K-1$. This proves the theorem. □

The following theorem shows how to construct the sequence of FOPs $f_\tau(z)$ from the MPFs $(p_\sigma(z), q_\tau(z))$. Regular FOPs correspond to denominator polynomials whose degree is equal to τ whereas the sizes of the blocks are determined by the actual degrees of the numerator polynomials.

THEOREM 31. *Let σ and τ be nonnegative integers such that $\sigma + \tau + 1 = K$. Let $(p_\sigma(z), q_\tau(z)) = \big(\hat{p}_\sigma(z)s(z)u(z), \hat{q}_\tau(z)s(z)u(z)\big)$ be the general MPF for the degree conditions* (40) *and the interpolation conditions* (41)*, where the polynomials $\hat{p}_\sigma(z)$, $\hat{q}_\tau(z)$, $s(z)$ and $u(z)$ are as in Theorem* 28*. If τ is a regular index for $\langle\langle \cdot, \cdot \rangle\rangle$, then $\deg\big(\hat{q}_\tau(z)s(z)\big) = \tau$, the FOP $f_\tau(z)$ is given by $f_\tau(z) \equiv \hat{q}_\tau(z)s(z)$ and the smallest regular index that is larger than τ is equal to $K - \deg\big(\hat{p}_\sigma(z)s(z)\big)$. Conversely, if $\deg\big(\hat{q}_\tau(z)s(z)\big) = \tau$, then τ is a regular index for $\langle\langle \cdot, \cdot \rangle\rangle$.*

PROOF. Suppose that τ is a regular index for $\langle\langle \cdot, \cdot \rangle\rangle$. Then $\det \mathcal{H}_\tau \neq 0$ and there exists precisely one monic polynomial $f_\tau(z)$ of degree τ such that

$$\langle\langle z^p, f_\tau(z) \rangle\rangle = 0 \qquad \text{for } p = 0, 1, \ldots, \tau - 1.$$

Let $p(z)$ be the polynomial of degree $\leq K-1$ that interpolates $g_{K-1}(z)f_\tau(z)$ at the points $t_1,\ldots,t_K$. Then, according to Theorem 30, $\deg p(z) \leq K-\tau-1 = \sigma$. Thus $\big(p(z), f_\tau(z)\big)$ is a MPF for the degree conditions (40) and the interpolation conditions (41). In other words, there exists a MPF whose denominator polynomial has degree τ. Theorem 28 then implies that there exists a monic polynomial $u_\tau(z)$ of degree $\tau - \deg\big(\hat{q}_\tau(z)s(z)\big)$ such that $f_\tau(z) \equiv \hat{q}_\tau(z)s(z)u_\tau(z)$. If $\deg u_\tau(z) > 0$, then we can choose a different monic polynomial $\tilde{u}_\tau(z)$ of the same degree. Then $f_\tau(z) \not\equiv \hat{q}_\tau(z)s(z)\tilde{u}_\tau(z)$ and, by Theorem 29,

$$\langle\langle z^p, \hat{q}_\tau(z)s(z)\tilde{u}_\tau(z)\rangle\rangle = 0 \qquad \text{for } p = 0,1,\ldots,\tau-1.$$

As $\deg\big(\hat{q}_\tau(z)s(z)\tilde{u}_\tau(z)\big) = \tau$, this contradicts the fact that $f_\tau(z)$ is unique. Thus we may conclude that $\deg\big(\hat{q}_\tau(z)s(z)\big) = \tau$ and $f_\tau(z) \equiv \hat{q}_\tau(z)s(z)$. Now Theorem 29 implies that

$$\langle\langle z^p, f_\tau(z)\rangle\rangle = 0 \qquad \text{for } p = 0,1,\ldots,K-2-\deg\big(\hat{p}_\sigma(z)s(z)\big)$$

and

$$\langle\langle z^p, f_\tau(z)\rangle\rangle \neq 0 \qquad \text{if } p = K-1-\deg\big(\hat{p}_\sigma(z)s(z)\big).$$

The structure of the diagonal blocks of the Gram matrix $[\langle\langle f_r, f_s\rangle\rangle]_{r,s=0}^{K-1}$ then implies that $\det \mathcal{H}_t = 0$ for $t = \tau+1,\ldots,K-1-\deg\big(\hat{p}_\sigma(z)s(z)\big)$ and that $\det \mathcal{H}_t \neq 0$ if $t = K-\deg\big(\hat{p}_\sigma(z)s(z)\big)$.

Suppose that $\deg\big(\hat{q}_\tau(z)s(z)\big) = \tau$. Then there exists only one MPF for the degree conditions (40) and the interpolation conditions (41). The polynomial $\hat{q}_\tau(z)s(z)$ is a monic polynomial of degree τ and, according to Theorem 29,

$$\langle\langle z^p, \hat{q}_\tau(z)s(z)\rangle\rangle = 0 \qquad \text{for } p = 0,1,\ldots,\tau-1.$$

Suppose that there exists another monic polynomial $\tilde{f}_\tau(z)$ of degree τ, $\tilde{f}_\tau(z) \not\equiv \hat{q}_\tau(z)s(z)$, such that

$$\langle\langle z^p, \tilde{f}_\tau(z)\rangle\rangle = 0 \qquad \text{for } p = 0,1,\ldots,\tau-1.$$

Let $p(z)$ be the polynomial of degree $\leq K-1$ that interpolates $g_{K-1}(z)\tilde{f}_\tau(z)$ at the points $t_1,\ldots,t_K$. Then, according to Theorem 30, $\deg p(z) \leq K-\tau-1 = \sigma$. Thus $\big(p(z), \tilde{f}_\tau(z)\big)$ is a MPF for the degree conditions (40) and the interpolation conditions (41). It follows that $\tilde{f}_\tau(z) \equiv \hat{q}_\tau(z)s(z)$. In other words, there exists only one monic polynomial of degree τ that is orthogonal (with respect to $\langle\langle\cdot,\cdot\rangle\rangle$) to all polynomials of lower degree. Thus τ is a regular index for $\langle\langle\cdot,\cdot\rangle\rangle$. This proves the theorem. □

The previous theorem suggests the following look-ahead strategy. Start with $\tau = 0$ and the corresponding MPF $(g_{K-1}(z), 1)$. Then set $\tau \leftarrow K - \deg g_{K-1}(z)$. Note that $\tau = 1$ as $\deg g_{K-1}(z) = K-1$. Compute the corresponding MPF $\big(p_\sigma(z), q_\tau(z)\big)$. Note that, as τ is a regular index for $\langle\langle\cdot,\cdot\rangle\rangle$, this MPF is uniquely defined, i.e., the polynomial $u(z) \equiv 1$ (cf. Theorem 28). Use $K - \deg p_\sigma(z)$ as the next value of τ, and so on. Observe that, if $\deg p_\sigma(z) = \sigma$, then the next value of τ is given by $\tau+1$. The interpolation problems can be solved via the algorithm of Van Barel and Bultheel [**274**]. This algorithm provides the coefficients of the numerator and the denominator polynomial in the standard monomial basis.

Of course, in floating-point arithmetic this strategy will only work if one uses a concept of 'numerical degree' instead of the classical 'degree'. The numerical degree of a polynomial can be defined as follows. Let $\epsilon > 0$. The ϵ-degree of a polynomial $p(z) =: p_0 + p_1 z + \cdots + p_{K-1} z^{K-1} \in \mathcal{P}$ of degree $\leq K-1$ is defined as follows. Let

$$\chi(k) := \frac{\max\{|p_{k+1}|, \ldots, |p_{K-1}|\}}{|p_k|}$$

for all $k \in \{0, 1, \ldots, K-2\}$ such that $p_k \neq 0$ and $\chi(k) := \infty$ otherwise. If

$$\min_{0 \leq k \leq K-2} \chi(k) \leq \epsilon, \tag{44}$$

then the ϵ-degree of $p(z)$ is defined as the index k for which the minimum in (44) is attained. Else, the ϵ-degree of $p(z)$ is set equal to $K-1$.

The following corollaries provide us with a stopping criterion.

COROLLARY 32. *Let σ and τ be nonnegative integers such that $\sigma + \tau + 1 = K$. Let $(p_\sigma(z), q_\tau(z))$ be a MPF for the degree conditions (40) and the interpolation conditions (41). If $K \geq 2n$ and $n \leq \tau \leq \lfloor K/2 \rfloor$, then $\deg p_\sigma(z) \leq \deg q_\tau(z) - 1$.*

PROOF. This follows from Theorem 30, Theorem 24, and Corollary 26. □

COROLLARY 33. *Let σ and τ be nonnegative integers such that $\sigma + \tau + 1 = K$. Let $(p_\sigma(z), q_\tau(z))$ be a MPF for the degree conditions (40) and the interpolation conditions (41). Let $p_\sigma(z) =: p_0 + p_1 z + \cdots + p_{K-1} z^{K-1}$. If $K \geq 2m$ and $m \leq \tau \leq \lfloor K/2 \rfloor$, then $p_k = \mathcal{O}(\delta^2)$, $\delta \to 0$ for $k = \deg q_\tau(z), \ldots, K-1$. In other words, if ϵ is sufficiently small, then the ϵ-degree of $p_\sigma(z)$ is less or equal than $\deg q_\tau(z) - 1$.*

PROOF. This follows from Theorem 30, Theorem 24, and Corollary 27. □

In other words, at the end the (numerical) degree of the numerator polynomial is less or equal than the degree of the denominator polynomial minus one. One can easily verify that this stopping criterion is equivalent to the one used in the algorithm that we have presented in Chapter 1.

Let us consider the problem of how to evaluate the polynomial $g_{K-1}(z)$ at the Kth roots of unity $t_1, \ldots, t_K$. One can easily verify that

$$g_{K-1}(z) = \frac{1}{2\pi i} \int_\gamma \frac{t^K - z^K}{t - z} \frac{f'(t)}{f(t)}\, dt$$

if $z \notin \gamma$. Thus, if $t_k \notin \gamma$ for $k = 1, \ldots, K$, then

$$g_{K-1}(t_k) = \frac{1}{2\pi i} \int_\gamma \frac{t^K - 1}{t - t_k} \frac{f'(t)}{f(t)}\, dt$$

for $k = 1, \ldots, K$.

In case γ is the unit circle, one can obtain accurate approximations for

$$g_{K-1}(t_1), \ldots, g_{K-1}(t_K)$$

in a very efficient way. Let L be a positive integer $\geq K$, preferably a power of 2. Let $\omega_1, \ldots, \omega_L$ be the Lth roots of unity,

$$\omega_l := \exp\Big(\frac{2\pi i}{L} l\Big), \qquad l = 1, \ldots, L.$$

THEOREM 34. *Suppose that γ is the unit circle. Let $v_L \in \mathbb{C}^{L\times 1}$ be the vector*

$$v_L := \frac{1}{L}\begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{L-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_L & \cdots & \omega_L^{L-1} \end{bmatrix}^H \begin{bmatrix} (f'/f)(\omega_1) \\ \vdots \\ (f'/f)(\omega_L) \end{bmatrix}.$$

Then

$$\begin{bmatrix} O_{K\times(L-K)} & I_K \end{bmatrix} v_L \approx \begin{bmatrix} s_{K-1} \\ \vdots \\ s_0 \end{bmatrix}$$

where $O_{K\times(L-K)}$ denotes the $K \times (L-K)$ zero matrix and I_K denotes the $K \times K$ identity matrix. In other words, the K last components of v_L are approximations for $s_{K-1}, \ldots, s_1, s_0$.

PROOF. By approximating

$$s_p = \frac{1}{2\pi\, i}\int_\gamma z^p \frac{f'(z)}{f(z)}\, dz = \frac{1}{2\pi}\int_0^{2\pi} e^{ip\theta} e^{i\theta} \frac{f'(e^{i\theta})}{f(e^{i\theta})}\, d\theta, \qquad p = 0, 1, 2, \ldots,$$

via the trapezoidal rule, we obtain that

$$s_p \approx \frac{1}{L}\sum_{l=1}^{L} \omega_l^{p+1} \frac{f'(\omega_l)}{f(\omega_l)}, \qquad p = 0, 1, 2, \ldots.$$

It follows that

$$s_p \approx \frac{1}{L}\sum_{l=1}^{L} \overline{\omega_l}^{L-1-p} \frac{f'(\omega_l)}{f(\omega_l)}, \qquad p = 0, 1, \ldots, L-1.$$

This proves the theorem. □

Since

$$\begin{bmatrix} g_{K-1}(t_1) \\ \vdots \\ g_{K-1}(t_K) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \cdots & t_1^{K-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_K & \cdots & t_K^{K-1} \end{bmatrix} \begin{bmatrix} s_{K-1} \\ \vdots \\ s_0 \end{bmatrix},$$

it follows that we can obtain approximations for $g_{K-1}(t_1), \ldots, g_{K-1}(t_K)$ via one L-point (inverse) FFT and one K-point FFT.

4. More numerical examples

We have implemented the strategy described in the previous section in Matlab. We have considered the case that γ is the unit circle. Approximations for

$$g_{K-1}(t_1), \ldots, g_{K-1}(t_K)$$

have been computed by using Theorem 34. The interpolation problems have been solved via the algorithm of Van Barel and Bultheel [**274**].

EXAMPLE 16. Let us reconsider the problem that we have studied in Example 15. As the corresponding γ is given by $\gamma = \{ z \in \mathbb{C} : |z| = 5 \}$, we divide all the zeros by 5 to transform the problem to the unit disk. Recall that $N = n = 10$ whereas $m = 4$. We set $L = 512$ and $K = 22$.

In Figure 1 we plot the logarithm with base 10 of the modulus of the coefficients of $p_\sigma(z)$ for $\tau = 0, 1, \ldots, 5$. (The logarithm of the modulus of the lowest degree coefficient is shown on the left. In general, the coefficient of z^k corresponds to the abscis $k + 1$.) Note that $\sigma = 21 - \tau$.

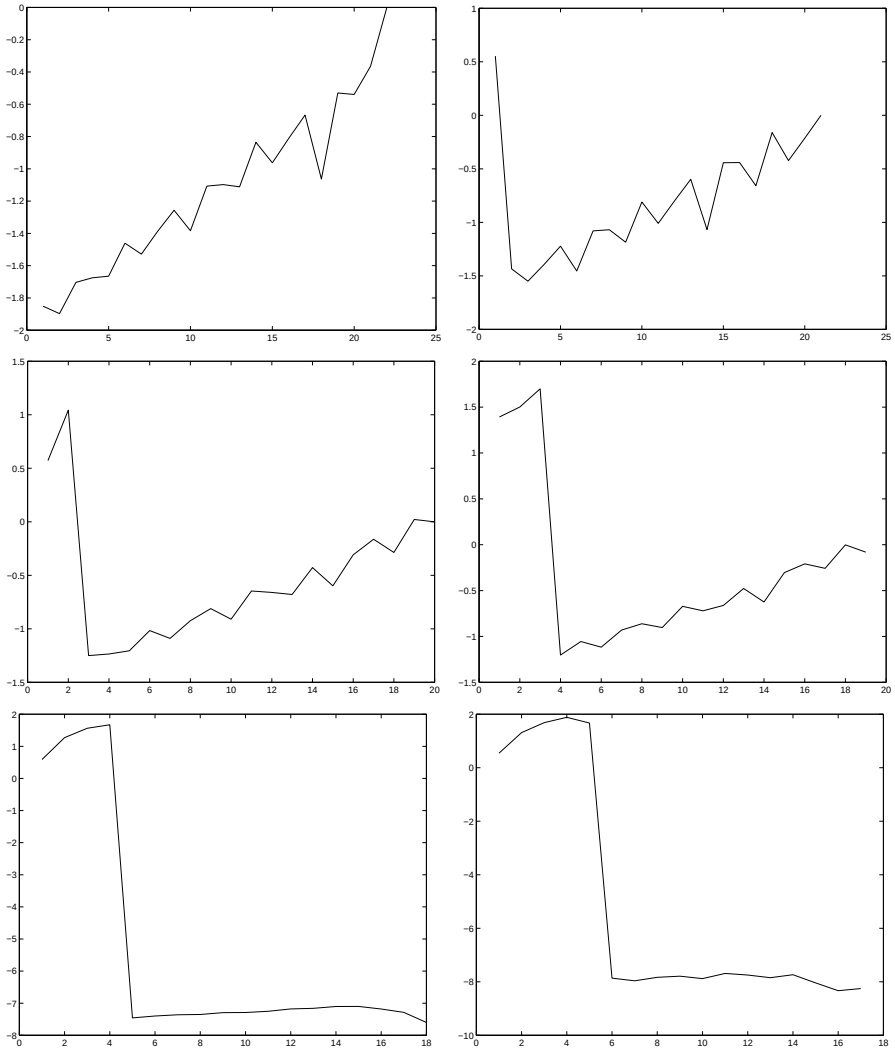


FIGURE 1. The coefficients of $p_\sigma(z)$ for $\tau = 0, 1, \ldots, 5$

Clearly $m = 4$. By multiplying the zeros of $q_4(z)$, as computed via the Matlab command `roots`, by 5 to transform them back to the setting of Example 15, we obtain the following:

```
-9.999999564178451e-01 - i  5.152536484956870e-08
 4.000050001653282e+00 + i  5.000694740214643e-05
 2.335838430343232e-04 + i  3.000299920075351e+00
-3.000024926663527e+00 + i  3.000149946356137e+00
```

These values are to be compared with the approximations for the centres of the clusters that we have obtained in Example 15, namely

```
-9.999999564181510e-01 - i  5.152524762408461e-08
 4.000050001653271e+00 + i  5.000694739720757e-05
 2.335838430156945e-04 + i  3.000299920075392e+00
-3.000024926663507e+00 + i  3.000149946356108e+00
```

The figure that corresponds to $\tau = 5$ is included to illustrate that the results given in Corollary 33 hold not only for $\tau = m$ but for $\tau \geq m$. The zeros of $q_5(z)$ lead to the same approximations for the centres as the zeros of $q_4(z)$ and one spurious "centre."

EXAMPLE 17. Let

$$f(z) = (\sinh(2z^2) + \sinh(10z) - 1) \times (\sinh(2z^2) + \sinh(10z) - 1.01)(\sinh(2z^2) + \sinh(10z) - 1.02).$$

This function has 21 simple zeros inside the unit circle. They form 7 clusters, where each cluster consists of 3 zeros. Thus $N = n = 21$ and $m = 7$. This example was also studied by Sakurai et al. [**248**]. We set $L = 512$ and $K = 42$.

In Figure 2 we plot the logarithm with base 10 of the modulus of the coefficients of $p_\sigma(z)$ for $\tau = 0, 1, \ldots, 7$.

The zeros of $q_7(z)$ are given by

```
-1.848537713183581e-01 - i  8.949141853554533e-01
-1.848537713183412e-01 + i  8.949141853554334e-01
-1.003354151041395e-01 - i  3.061151582728444e-01
-1.003354151030711e-01 + i  3.061151582802838e-01
 1.335489810139705e-01 - i  6.084120926164355e-01
 1.335489810131479e-01 + i  6.084120926165633e-01
 8.777826151937687e-02 + i  8.843042856595357e-12
```

These match the approximations for the centres of the clusters that were given in [**248**]. ♢

Acknowledgements

The results in this chapter were obtained in collaboration with Marc Van Barel.

Tetsuya Sakurai contributed to Theorem 19, 22 and 24. I met Tetsuya in person while he was staying at the Department of Mathematics and Computer Science of the University of Antwerp. I would like to thank Annie Cuyt for inviting us both to give a talk at her colloquium "Computer Arithmetic and Numerical Techniques" (October 23, 1997).

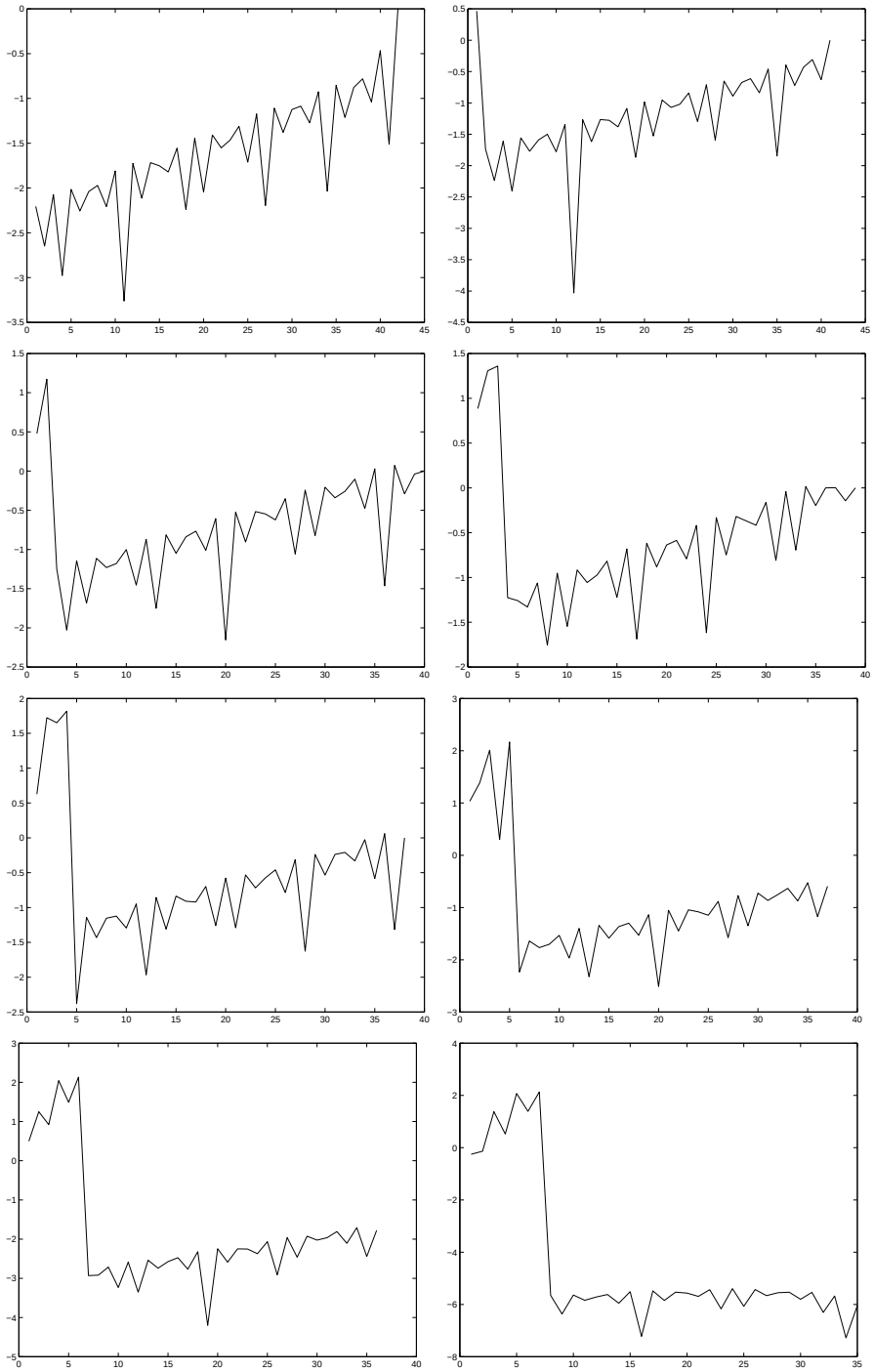


FIGURE 2. The coefficients of $p_\sigma(z)$ for $\tau = 0, 1, \ldots, 7$

CHAPTER 3

Zeros and poles of meromorphic functions

In the previous chapters we have considered the problem of computing all the zeros of an analytic function that lie in the interior of a Jordan curve. We have seen how the algorithm that we have proposed can also be used to locate clusters of zeros. Our algorithm gives accurate results. It is based on the theory of formal orthogonal polynomials. Information concerning the location of the zeros is obtained via numerical integration.

We will show how these results can be used to tackle the problem of computing all the zeros and poles of a meromorphic function that lie in the interior of a Jordan curve. More precisely, given a meromorphic function f and a positively oriented Jordan curve γ that does not pass through any zero or pole of f, we will present an algorithm for computing all the zeros and poles of f that lie inside γ, together with their respective multiplicities and orders. An upper bound for the total number of poles of f that lie inside γ is assumed to be known. Initial approximations for the zeros and poles are not needed.

This chapter is inspired by our paper [**191**].

1. Introduction

Let W be a simply connected region in $\mathbb{C}$, $f : W \to \mathbb{C}$ analytic in W, and γ a positively oriented Jordan curve in W that does not pass through any zero of f. In Chapter 1 we have considered the problem of computing *all* the zeros $z_1, \ldots, z_n$ of f that lie inside γ, together with their respective multiplicities $\nu_1, \ldots, \nu_n$. The number of mutually distinct zeros of f that lie inside γ is denoted by n while N stands for the total number of zeros of f that lie inside γ, $N = \nu_1 + \cdots + \nu_n$. The algorithm that we have presented is based on the theory of formal orthogonal polynomials associated with the symmetric bilinear form

$$\langle \cdot, \cdot \rangle \; : \; \mathcal{P} \times \mathcal{P} \to \mathbb{C} \; : \; (\phi, \psi) \mapsto \langle \phi, \psi \rangle := \frac{1}{2\pi i} \int_\gamma \phi(z)\psi(z)\frac{f'(z)}{f(z)}\, dz.$$

As the logarithmic derivative f'/f has a simple pole at z_k with residue ν_k for $k = 1, \ldots, n$, Cauchy's Theorem implies that

$$\langle \phi, \psi \rangle = \sum_{k=1}^{n} \nu_k \phi(z_k)\psi(z_k). \tag{45}$$

Our algorithm proceeds as follows. It computes $z_1, \ldots, z_n$ as the zeros of the nth degree formal orthogonal polynomial (FOP) that is associated with $\langle \cdot, \cdot \rangle$. The value of n is determined indirectly. Once n and $z_1, \ldots, z_n$ have been found, the problem

becomes linear and $\nu_1, \ldots, \nu_n$ are calculated by solving a Vandermonde system. The total number of zeros N can be computed via numerical integration. It is an upper bound for n. The fact that such an upper bound is available plays a crucial role in the determination of the value of n.

Now suppose that f is not analytic but meromorphic in W and suppose that f has neither zeros nor poles on γ. We will show how the algorithm that we have presented in Chapter 1 can be extended to compute *all* the zeros and poles of f that lie in the interior of γ, together with their respective multiplicities and orders. Let P denote the total number of poles of f that lie inside γ (i.e., the number of poles where each pole is counted according to its order). Suppose from now on that $N + P > 0$. Let p denote the number of mutually distinct poles of f that lie inside γ. Let $y_1, \ldots, y_p$ be these poles and $\mu_1, \ldots, \mu_p$ their respective orders. An easy calculation shows that f'/f has a simple pole at y_l with residue $-\mu_l$ for $l = 1, \ldots, p$. Therefore

$$\langle \phi, \psi \rangle = \sum_{k=1}^{n} \nu_k \phi(z_k)\psi(z_k) - \sum_{l=1}^{p} \mu_l \phi(y_l)\psi(y_l). \tag{46}$$

This expression is of the same type as (45). The theorems that we have proven in Chapter 1 do not rely on the fact that the ν_k's are known to be positive integers. They merely use the fact that $\nu_k \neq 0$ for $k = 1, \ldots, n$. It is therefore likely that these results can be extended to cover (46). We will see that $z_1, \ldots, z_n$ and $y_1, \ldots, y_p$ can indeed be calculated as the zeros of the FOP of degree $n + p$. Provided, of course, that an upper bound M for $m := n + p$ is known. Assume that an upper bound $\hat{P}$ for P is known and define the ordinary moments

$$s_r := \langle 1, z^r \rangle$$

for $r = 0, 1, 2, \ldots$. Then

$$s_r = \nu_1 z_1^r + \cdots + \nu_n z_n^r - \mu_1 y_1^r - \cdots - \mu_p y_p^r$$

for $r = 0, 1, 2, \ldots$. In particular, $s_0 = N - P$ and we may assume that the value of s_0 has been computed. As $n + p \leq N + P = s_0 + 2P \leq s_0 + 2\hat{P}$, it follows that we may take $M = s_0 + 2\hat{P}$. In case γ is the unit circle, an upper bound for P can be obtained by using the heuristic approach of Gleyse and Kaliaguine [**103**].

For previous attempts to tackle the problem of computing all the zeros and poles of a meromorphic function that lie in the interior of a Jordan curve, we refer to Abd-Elall, Delves and Reid [**1**] and also Ioakimidis [**156, 157**].

2. Theoretical considerations and numerical algorithm

Instead of (46) we will consider the following even more general setting. Let $\langle \cdot, \cdot \rangle : \mathcal{P} \times \mathcal{P} \to \mathbb{C}$ be any symmetric and bilinear form such that

$$\langle \phi, \psi \rangle = \sum_{k=1}^{m} \lambda_k \phi(x_k)\psi(x_k) \tag{47}$$

for any $\phi, \psi \in \mathcal{P}$, where $m \in \mathbb{N}_0$, $\lambda_1, \ldots, \lambda_m \in \mathbb{C}_0$ and $x_1, \ldots, x_m \in \mathbb{C}$ are unknown. Suppose that the points $x_1, \ldots, x_m$ are known to be mutually distinct. Assume that

we have an 'oracle' at our disposal that provides us with the value of $\langle \phi, \psi \rangle$ for any $\phi, \psi \in \mathcal{P}$ upon simple request. Let an upper bound M for m be given.

Our unknowns m, $x_1, \ldots, x_m$ and $\lambda_1, \ldots, \lambda_m$ can be calculated in the same way as n, $z_1, \ldots, z_n$ and $\nu_1, \ldots, \nu_n$ have been calculated in Chapter 1. The key point is that the FOP of degree m that is associated with the form $\langle \cdot, \cdot \rangle$ is given by

$$(z - x_1) \cdot \cdots \cdot (z - x_m),$$

as one can easily verify.

Let us start by formulating what could be called the "theoretical" solution, cf. Theorems 3 and 4 and Corollary 8. This will ultimately lead to a numerical algorithm that generalizes the algorithm that we have presented in Chapter 1.

Theorems 3 and 4 have been formulated in terms of ordinary moments and Hankel matrices whereas Corollary 8 relies upon formal orthogonal polynomials. These results can be formulated in terms of polynomials from an arbitrary basis for $\mathcal{P}$. Indeed, let ψ_k be a monic polynomial of degree k for $k = 0, 1, 2, \ldots$. Define the $k \times k$ matrices F_k and $F_k^{(1)}$ as

$$F_k := \left[\ \langle \psi_r, \psi_s \rangle \ \right]_{r,s=0}^{k-1} \qquad \text{and} \qquad F_k^{(1)} := \left[\ \langle \psi_r, \psi_1 \psi_s \rangle \ \right]_{r,s=0}^{k-1}$$

for $k = 1, 2, \ldots$.

The following theorem characterizes m.

THEOREM 35. *$m = \operatorname{rank} F_{m+r}$ for every nonnegative integer r. In particular, $m = \operatorname{rank} F_M$.*

PROOF. Let r be a nonnegative integer. The matrix F_{m+r} can be written as

$$\begin{aligned} F_{m+r} &= \sum_{k=1}^{m} \lambda_k \begin{bmatrix} \psi_0(x_k)\psi_0(x_k) & \cdots & \psi_0(x_k)\psi_{m+r-1}(x_k) \\ \vdots & & \vdots \\ \psi_{m+r-1}(x_k)\psi_0(x_k) & \cdots & \psi_{m+r-1}(x_k)\psi_{m+r-1}(x_k) \end{bmatrix} \\ &= \sum_{k=1}^{m} \lambda_k \begin{bmatrix} \psi_0(x_k) \\ \vdots \\ \psi_{m+r-1}(x_k) \end{bmatrix} \begin{bmatrix} \psi_0(x_k) & \cdots & \psi_{m+r-1}(x_k) \end{bmatrix}. \end{aligned}$$

This implies that $\operatorname{rank} F_{m+r} \leq m$. However, F_m is nonsingular. Indeed, one can easily verify that F_m can be factorized as $F_m = V_m D_m V_m^T$ where V_m is the Vandermonde-like matrix

$$V_m := [\, \psi_r(x_s) \,]_{r=0,s=1}^{m-1,m}$$

and D_m is the diagonal matrix

$$D_m := \operatorname{diag}(\lambda_1, \ldots, \lambda_m).$$

Therefore $\operatorname{rank} F_{m+r} \geq m$. It follows that $\operatorname{rank} F_{m+r} = m$. □

Thus the regular FOP of degree m exists whereas regular FOPs of degree $> m$ do not exist. Note that, in contrast to the setting of Chapters 1 and 2, the regular FOP of degree one need not exist. This will be the only important point in which the algorithm that we will present will be different from the algorithm that we have presented in Chapter 1. If $\langle 1, 1 \rangle = \sum_{k=1}^{m} \lambda_k \neq 0$, then the regular FOP of degree

one exists and it is given by $\varphi_1(z) = z - \mu$ where $\mu := \langle 1, z\rangle / \langle 1, 1\rangle$. In the case of meromorphic functions, the condition $\langle 1, 1\rangle \neq 0$ means that $N \neq P$. For analytic functions, we have that $P = 0$ and thus the regular FOP of degree one always exists (if we assume, of course, that $N > 0$).

The following theorem shows how $x_1, \ldots, x_m$ can be computed by solving a generalized eigenvalue problem.

THEOREM 36. *The eigenvalues of the pencil* $F_m^{(1)} - \lambda F_m$ *are* $\psi_1(x_1), \ldots, \psi_1(x_m)$.

PROOF. Define V_m as the Vandermonde-like matrix

$$V_m := [\, \psi_r(x_s) \,]_{r=0,s=1}^{m-1,m}$$

and let D_m and $D_m^{(1)}$ be the diagonal matrices

$$D_m := \mathrm{diag}(\lambda_1, \ldots, \lambda_m) \quad \text{and} \quad D_m^{(1)} := \mathrm{diag}(\lambda_1 \psi_1(x_1), \ldots, \lambda_m \psi_1(x_m)).$$

Then F_m and $F_m^{(1)}$ can be factorized as $F_m = V_m D_m V_m^T$ and $F_m^{(1)} = V_m D_m^{(1)} V_m^T$. Let $\lambda^\star$ be an eigenvalue of the pencil $F_m^{(1)} - \lambda F_m$ and x a corresponding eigenvector. Then

$$\begin{aligned}
& F_m^{(1)} x = \lambda^\star F_m x \\
\Leftrightarrow \quad & (V_m D_m^{(1)} V_m^T) x = \lambda^\star (V_m D_m V_m^T) x \\
\Leftrightarrow \quad & D_m^{(1)} y = \lambda^\star D_m y \quad \text{if } y := V_m^T x \\
\Leftrightarrow \quad & \mathrm{diag}(\psi_1(x_1), \ldots, \psi_1(x_m)) y = \lambda^\star y.
\end{aligned}$$

This proves the theorem. □

As $\psi_1(z)$ is a polynomial of degree one, the "nodes" $x_1, \ldots, x_m$ can be calculated by applying the previous theorem. Once m and $x_1, \ldots, x_m$ have been found, the problem becomes linear and the "weights" $\lambda_1, \ldots, \lambda_m$ can be computed by solving a Vandermonde-like system of linear equations.

The question remains, of course, which polynomials $\psi_k(z)$ to choose. Again, as in Chapter 1, we will obtain very accurate numerical results by using the formal orthogonal polynomials associated with $\langle \cdot, \cdot \rangle$.

If F_m is strongly nonsingular, then we have a full set $\{ \varphi_0, \varphi_1, \ldots, \varphi_m \}$ of regular FOPs. If not, then the gaps in the sequence of existing regular FOPs can be filled up in the same way as before. We may therefore assume that a suitable sequence $\{\varphi_t\}_{t=0}^{\infty}$ has been defined.

It is a trivial task to generalize Theorems 7 (concerning the computation of zeros of regular FOPs) and 9 (concerning the stopping criterium/the determination of the value of m).

These considerations lead to the following algorithm.

ALGORITHM

input $\langle \cdot, \cdot \rangle$, M, ϵ_{stop}
output m, nodes
comment nodes $= \{x_1, \ldots, x_m\}$. We assume that $M \geq m$ and $\epsilon_{\mathrm{stop}} > 0$.
$\varphi_0(z) \leftarrow 1$

$r \leftarrow 0$
$s_0 \leftarrow \langle 1, 1 \rangle$
if $s_0 == 0$ **then**
  $\mu \leftarrow 0$
  $\varphi_1(z) \leftarrow z$
  $t \leftarrow 1$
else
  $\mu \leftarrow \langle 1, z \rangle / s_0$
  $\varphi_1(z) \leftarrow z - \mu$
  $r \leftarrow 1$; $t \leftarrow 0$
end if
while $r + t < M$ **do**
  regular $\leftarrow$ it is numerically feasible to generate $\varphi_{r+t+1}(z)$ as
    a regular FOP
  if regular **then**
    generate $\varphi_{r+t+1}(z)$ as a regular FOP
    $r \leftarrow r + t + 1$; $t \leftarrow 0$
    allsmall $\leftarrow$ **true**; $\tau \leftarrow 0$
    while allsmall **and** $(r + \tau < M)$ **do**
      $[\,\mathrm{ip}, \mathrm{maxpsum}\,] \leftarrow \langle (z - \mu)^\top \varphi_r(z), \varphi_r(z) \rangle$
      ip $\leftarrow |\mathrm{ip}|$
      allsmall $\leftarrow$ (ip/maxpsum $< \epsilon_{\mathrm{stop}}$)
      $\tau \leftarrow \tau + 1$
    end while
    if allsmall **then**
      $m \leftarrow r$; nodes $\leftarrow$ roots(φ_r); **stop**
    end if
  else
    generate $\varphi_{r+t+1}(z)$ as an inner polynomial
    $t \leftarrow t + 1$
  end if
end while
$m \leftarrow M$; nodes $\leftarrow$ roots(φ_N); **stop**

Similar comments as those given after the formulation of the algorithm in Chapter 1 apply.

3. A numerical example

We have modified the Matlab implementation that we have already used in Section 4 of Chapter 1.

EXAMPLE 18. Suppose that

$$f(z) = \frac{1}{z^2(z-1)(z^2+9)} + z \sin z + e^{-3z} + 4$$

and let γ be the circle $\{z \in \mathbb{C} : |z| = 2\}$. We set the upper bound $\hat{P} = 5$. It turns out that $\langle 1, 1 \rangle = 0$ (in other words, $N = P$) and thus $M = s_0 + 2\hat{P} = 10$. We set

$\epsilon_{\text{stop}} = 10^{-8}$. Our algorithm reacts as follows. It defines $\varphi_0(z)$ as a regular FOP, $\varphi_1(z)$ as an inner polynomial and $\varphi_2(z)$ as a regular FOP. The scaled counterpart of $|\langle\varphi_2(z), \varphi_2(z)\rangle|$ is equal to

```
4.821901380151357e-01
```

The algorithm defines $\varphi_3(z)$ as an inner polynomial and $\varphi_4(z)$ as a regular FOP. The scaled counterpart of $|\langle\varphi_4(z), \varphi_4(z)\rangle|$ is equal to

```
1.776524543847086e-01
```

The algorithm defines $\varphi_5(z)$ as a regular FOP. For $k = 0, 1, \ldots, 4$, the scaled counterparts of $|\langle z^k\varphi_5(z), \varphi_5(z)\rangle|$ are given by

```
2.387781621321191e-15
2.887901534660305e-15
1.268594438890019e-15
3.904276363179922e-15
2.772600727960076e-15
```

The algorithm decides that $n+p = 5$ and stops. The absolute errors of the computed approximations for the zeros and poles are $\mathcal{O}(10^{-12})$. After one step of iterative refinement, we obtain the following results:

x_k	λ_k
0.97843635600921	1
0.16974891913248	1
−0.13327146070751	1
1.00000000000000	−1
0.00000000000000	−2

Note how zeros and poles can be distinguished by checking the signs of the λ_k's. $\diamondsuit$

NOTE. As our algorithm provides not only approximations for the zeros and poles but also the corresponding multiplicities and orders, we can use the modified Newton's method

$$z_k^{(\alpha+1)} = z_k^{(\alpha)} - \nu_k \frac{f(z_k^{(\alpha)})}{f'(z_k^{(\alpha)})}, \qquad y_l^{(\alpha+1)} = y_l^{(\alpha)} + \mu_l \frac{f(y_l^{(\alpha)})}{f'(y_l^{(\alpha)})}, \qquad \alpha = 0, 1, 2, \ldots,$$

to refine the approximations for the zeros and poles.

CHAPTER 4

Systems of analytic equations

In Chapter 1 we have considered the problem of computing all the zeros of an analytic function f that lie in the interior of a Jordan curve γ. The algorithm that we have presented computes not only accurate approximations for the zeros but also their respective multiplicities. It does not require initial approximations for the zeros. In Chapter 2 we have seen how our algorithm can be used to locate clusters of zeros of analytic functions whereas in Chapter 3 we have adapted it to handle the problem of calculating zeros and poles of meromorphic functions. As our approach relies upon integrals along γ that involve the logarithmic derivative f'/f, it could be called a *logarithmic residue* based approach.

In this chapter we will present a logarithmic residue based approach for the problem of computing zeros of analytic mappings (in other words, for solving systems of analytic equations). A multidimensional logarithmic residue formula is available in the literature. This formula involves the integral of a differential form, which we will transform into a sum of Riemann integrals. More precisely, if d denotes the dimension of the mapping (i.e., the number of equations, which is assumed to be equal to the number of variables), then this sum consists of d Riemann integrals of dimension $2d-1$. We will show how the zeros and their respective multiplicities can be computed from these integrals by solving a generalized eigenvalue problem that has Hankel structure and d Vandermonde systems, cf. Theorems 3 and 4 and Equation (30).

It turns out that these integrals are difficult to evaluate numerically. Therefore we prefer to use ordinary moments instead of modified moments, even if this means that the computed approximations for the zeros will be less accurate. By using a cubature package that can handle *vectors* of similar integrals over a common integration region, we will be able to calculate all the ordinary moments that are needed simultaneously. This will enable us to reduce the cost of our approach. Indeed, a significant part of the computation required for each integrand will be the same for all of the integrands and thus these common calculations need to be done only once for each integrand evaluation point.

This chapter corresponds to our paper [**183**].

1. Introduction

Let $d \geq 1$ be a positive integer. Consider a polydisk D in $\mathbb{C}^d$ (i.e., D is the Cartesian product of d disks in $\mathbb{C}$) and let $f = (f_1, \ldots, f_d) : \overline{D} \to \mathbb{C}^d$ be a mapping that is analytic in $\overline{D}$ and has no zeros on the boundary of D. The latter implies that f has only a finite number of zeros in D and that these zeros are all isolated [**3**,

Theorem 2.4]. We consider the problem of computing these zeros, together with their respective multiplicities. (For a precise definition of the multiplicity of a zero of an analytic mapping, we refer to Chapter 7.)

Let $Z_f(D)$ denote the set of zeros of f that lie in D and let $\mu_a(f)$ denote the multiplicity of a zero $a \in Z_f(D)$.

In case $d = 1$, the classical logarithmic residue formula that we have used in the previous chapters tells us that

$$\frac{1}{2\pi i} \int_{\partial D} \varphi(z) \frac{f'(z)}{f(z)} \, dz = \sum_{a \in Z_f(D)} \mu_a(f) \, \varphi(a) \tag{48}$$

if $\varphi : \overline{D} \to \mathbb{C}$ is analytic in D and continuous in $\overline{D}$. By evaluating the integral in the left-hand side numerically, we have been able to obtain information about the location of the zeros of f.

A multidimensional generalization of (48) is available in the theory of functions of several complex variables [**3**, Theorem 3.1]. It involves the integral of a differential form. To prepare for the numerical evaluation of this integral, we transform it into a Riemann integral, or rather, a sum of d Riemann integrals. This result is formulated in Theorem 37 and looks as follows:

$$I(\varphi) := \sum_{k=1}^{d} I_k(\varphi) = \sum_{a \in Z_f(D)} \mu_a(f) \, \varphi(a)$$

if $\varphi : \overline{D} \to \mathbb{C}$ is analytic in D and continuous in $\overline{D}$ and where $I_1(\varphi), \ldots, I_d(\varphi)$ are certain Riemann integrals over the unit cube in $\mathbb{R}^{2d-1}$. Observe that the total number of zeros of f that lie in D is equal to $I(1)$.

The proof of Theorem 37 is formulated in the language of differential forms. This cannot be avoided. Readers who feel less at ease with differential forms may consult [**62**], [**83**] or [**287**] for an introduction and [**222**] for a thorough exposition of analysis on complex manifolds.

Unfortunately, the integrals that appear in Theorem 37 tend to be difficult to evaluate numerically. The efficient numerical evaluation of these integrals is still an open problem. It represents a challenge for the numerical integration community. (More details about the various numerical integration algorithms that we have tried will be given in Section 4.)

NOTE. These integrals are similar to the Kronecker-Picard integrals that appear in topological degree based methods for computing solutions to twice continuously-differentiable systems of real equations. See, for example, Erdelsky [**76**], O'Neil and Thomas [**228**] and also Ragos, Vrahatis and Zafiropoulos [**243**], and Kavvadias and Vrahatis [**172**].

We will therefore assume that we are able to evaluate the functional $I(\varphi)$ for every function φ that satisfies the hypotheses of Theorem 37. In particular, we will suppose that the total number of zeros of f that lie in D can be computed. Our unknowns are the number of mutually distinct zeros of f that lie in D, these zeros themselves, and their respective multiplicities. In Section 3 we will show how specific

choices of φ enable us to calculate our unknowns by solving a generalized eigenvalue problem that has Hankel structure and d Vandermonde systems.

NOTE. Algebraic mappings are of course a special case of analytic mappings. Systems of polynomial equations have received considerable interest in recent years. Several classes of methods have been developed for their solution: Gröbner bases, homotopy continuation, sparse resultants and interval methods (see, for example, [**53, 74, 78, 200, 218, 282, 286**]). We will not compare our approach with these methods because they have been developed specifically for systems of polynomial equations whereas we consider systems of arbitrary analytic equations, a problem that has received much less attention in the literature.

2. A multidimensional logarithmic residue formula

Let J_f denote the Jacobian matrix of f and let $J_{[k]}$ be the Jacobian matrix of f with the kth column replaced with $\begin{bmatrix} f_1 & \cdots & f_d \end{bmatrix}^T$:

$$J_{[k]} := \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_{k-1}} & f_1 & \frac{\partial f_1}{\partial z_{k+1}} & \cdots & \frac{\partial f_1}{\partial z_d} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_d}{\partial z_1} & \cdots & \frac{\partial f_d}{\partial z_{k-1}} & f_d & \frac{\partial f_d}{\partial z_{k+1}} & \cdots & \frac{\partial f_d}{\partial z_d} \end{bmatrix}, \qquad k = 1, \ldots, d.$$

Suppose also that the polydisk D is given by

$$D = D_1 \times \cdots \times D_d$$

where

$$D_k = \{\, z \in \mathbb{C} : |z - C_k| < R_k \,\}, \qquad k = 1, \ldots, d,$$

with $C_1, \ldots, C_d \in \mathbb{C}$ and $R_1, \ldots, R_d > 0$.

THEOREM 37. *Let $\varphi : \overline{D} \to \mathbb{C}$ be analytic in D and continuous in $\overline{D}$. Define $I_k(\varphi)$ for $k = 1, \ldots, d$ as the integral*

$$I_k(\varphi) := \rho_k \int_{[0,1]^{2d-1}} \frac{\varphi(z_1, \ldots, z_d)\, \det J_f(z_1, \ldots, z_d)\, \overline{\det J_{[k]}(z_1, \ldots, z_d)}}{\left(|f_1(z_1, \ldots, z_d)|^2 + \cdots + |f_d(z_1, \ldots, z_d)|^2\right)^d} \times e^{2\pi i\, \theta_k}\, r_1 \cdots r_{k-1}\, r_{k+1} \cdots r_d\, dr_1 \cdots dr_{k-1}\, dr_{k+1} \cdots dr_d\, d\theta_1 \cdots d\theta_d$$

with

$$\rho_k = \rho(d, R_1, \ldots, R_d; k) := 2^{d-1}(d-1)!\, R_1^2 \cdots R_{k-1}^2\, R_k\, R_{k+1}^2 \cdots R_d^2$$

and where

$$z_k = z_k(\theta_k) = C_k + R_k\, e^{2\pi i\, \theta_k} \qquad 0 \le \theta_k \le 1$$

and

$$z_l = z_l(r_l, \theta_l) = C_l + r_l\, R_l\, e^{2\pi i\, \theta_l} \qquad 0 \le r_l, \theta_l \le 1$$

for $l \in \{1, \ldots, d\} \setminus \{k\}$. Then

$$I(\varphi) := \sum_{k=1}^{d} I_k(\varphi) = \sum_{a \in Z_f(D)} \mu_a(f)\, \varphi(a). \tag{49}$$

Theorem 37 is a corollary of the following theorem.

THEOREM 38. *Let D be a polydisk in $\mathbb{C}^d$ and let $f = (f_1, \ldots, f_d) : \overline{D} \to \mathbb{C}^d$ be a mapping that is analytic in $\overline{D}$ and has no zeros on the boundary of D. Let $\varphi : \overline{D} \to \mathbb{C}$ be analytic in D and continuous in $\overline{D}$. Define the differential form $\omega(f, \overline{f})$ as*

$$\omega(f, \overline{f}) := \frac{(d-1)!}{(2\pi i)^d} \frac{1}{|f|^{2d}} \sum_{k=1}^{d} (-1)^{k-1} \overline{f}_k \, d\overline{f}_{[k]} \wedge df.$$

Then

$$\int_{\partial D} \varphi \, \omega(f, \overline{f}) = \sum_{a \in Z_f(D)} \mu_a(f) \, \varphi(a).$$

PROOF. Yuzhakov and Roos [**3**, Theorem 3.1] proved this result for arbitrary bounded domains in $\mathbb{C}^d$ with a piecewise smooth boundary. □

The notations used in the formulation of the previous theorem,

$$\overline{f} = (\overline{f_1}, \ldots, \overline{f_d})$$
$$|f| = \sqrt{|f_1|^2 + \cdots + |f_d|^2}$$
$$df = df_1 \wedge \cdots \wedge df_d$$
$$df_{[k]} = df_1 \wedge \cdots \wedge \widetilde{df_k} \wedge \cdots \wedge df_d, \qquad k = 1, \ldots, d,$$

are classical. (The tilde over the form df_k means that this form is omitted and does not appear in the product.)

To get rid of the differential forms in Theorem 38, we proceed as follows. Define the form $\eta(f)$ as

$$\eta(f) := \sum_{k=1}^{d} (-1)^{k-1} f_k \, df_1 \wedge \cdots \wedge \widetilde{df_k} \wedge \cdots \wedge df_d.$$

This form is sometimes called the *Leray form* [**182**]. Then

$$\omega(f, \overline{f}) = \frac{(d-1)!}{(2\pi i)^d} \frac{1}{|f|^{2d}} \, \eta(\overline{f}) \wedge df.$$

If $f = f(z_1, \ldots, z_d)$, then $df = \det J_f(z_1, \ldots, z_d) \, dz_1 \wedge \cdots \wedge dz_d$. The following lemma shows what happens with the Leray form in this case.

LEMMA 39. *If $f = f(z_1, \ldots, z_d)$, then*

$$\eta(f) = \sum_{k=1}^{d} (-1)^{k-1} \det J_{[k]}(z_1, \ldots, z_d) \, dz_1 \wedge \cdots \wedge \widetilde{dz_k} \wedge \cdots \wedge dz_d.$$

PROOF. Let j be an integer between 1 and d, $j \in \{1, \ldots, d\}$. Then

$$\sum_{k=1}^{d} (-1)^{k-1} \det J_{[k]} \, (dz_1 \wedge \cdots \wedge \widetilde{dz_k} \wedge \cdots \wedge dz_d)\left(\frac{\partial}{\partial z_1}, \ldots, \widetilde{\frac{\partial}{\partial z_j}}, \ldots, \frac{\partial}{\partial z_d}\right)$$
$$= (-1)^{j-1} \det J_{[j]}.$$

By expanding $\det J_{[j]}$ along the jth column, we obtain that

$$(-1)^{j-1}\det J_{[j]}$$
$$= (-1)^{j-1}\sum_{k=1}^{d}(-1)^{k+j} f_k \det \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_1}{\partial z_{j-1}} & \frac{\partial f_1}{\partial z_{j+1}} & \cdots & \frac{\partial f_1}{\partial z_d} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial f_{k-1}}{\partial z_1} & \cdots & \frac{\partial f_{k-1}}{\partial z_{j-1}} & \frac{\partial f_{k-1}}{\partial z_{j+1}} & \cdots & \frac{\partial f_{k-1}}{\partial z_d} \\ \frac{\partial f_{k+1}}{\partial z_1} & \cdots & \frac{\partial f_{k+1}}{\partial z_{j-1}} & \frac{\partial f_{k+1}}{\partial z_{j+1}} & \cdots & \frac{\partial f_{k+1}}{\partial z_d} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial f_d}{\partial z_1} & \cdots & \frac{\partial f_d}{\partial z_{j-1}} & \frac{\partial f_d}{\partial z_{j+1}} & \cdots & \frac{\partial f_d}{\partial z_d} \end{bmatrix}.$$

The determinant in the right-hand side is by definition equal to

$$(df_1 \wedge \cdots \wedge \widetilde{df_k} \wedge \cdots \wedge df_d)(\frac{\partial}{\partial z_1}, \ldots, \widetilde{\frac{\partial}{\partial z_j}}, \ldots, \frac{\partial}{\partial z_d}).$$

This proves the lemma. □

It follows that

$$\omega(f,\overline{f}) = \sum_{k=1}^{d}\omega_k(f,\overline{f})$$

where

$$\omega_k(f,\overline{f}) := (-1)^{k-1}\,\frac{(d-1)!}{(2\pi i)^d}\,\frac{\det J_f\,\overline{\det J_{[k]}}}{|f|^{2d}}\,d\overline{z}_{[k]}\wedge dz, \qquad k=1,\ldots,d.$$

The boundary of D is given by

$$\partial D = \partial D_{[1]} \cup \cdots \cup \partial D_{[d]}$$

where

$$\partial D_{[k]} := D_1 \times \cdots \times D_{k-1} \times \partial D_k \times D_{k+1} \times \cdots \times D_d, \qquad k=1,\ldots,d,$$

and thus

$$\int_{\partial D}\varphi\,\omega(f,\overline{f}) = \sum_{k=1}^{d}\int_{\partial D_{[k]}}\varphi\,\omega_k(f,\overline{f}).$$

Now let us introduce polar coordinates. Let $C\in\mathbb{C}$ and $R>0$. If $z=z(\theta)=C+R\,e^{2\pi i\,\theta}$, then $dz = 2\pi i\,R\,e^{2\pi i\,\theta}\,d\theta$. And if $z=z(r,\theta)=C+rR\,e^{2\pi i\,\theta}$, then $d\overline{z}\wedge dz = 2(2\pi i)rR^2\,dr\wedge d\theta$. Thus

$$\int_{\partial D_{[k]}}\varphi\,\omega_k(f,\overline{f})$$
$$= (-1)^{k-1}\,\rho_k\int_{\partial D_{[k]}}\frac{\varphi\,\det J_f\,\overline{\det J_{[k]}}}{(|f_1|^2+\cdots+|f_d|^2)^d}\,[\prod_{l=1,\,l\neq k}^{d} r_l]\,e^{2\pi i\,\theta_k}\,dr_{[k]}\wedge d\theta$$

for $k=1,\ldots,d$, where

$$\rho_k = \rho(d,R_1,\ldots,R_d;k) := 2^{d-1}(d-1)!\,R_1^2\cdots R_{k-1}^2\,R_k\,R_{k+1}^2\cdots R_d^2$$

and in the integral on the right-hand side

$$z_k = z_k(\theta_k) = C_k + R_k\, e^{2\pi i\,\theta_k}$$

and

$$z_l = z_l(r_l, \theta_l) = C_l + r_l\, R_l\, e^{2\pi i\,\theta_l}$$

for $l \in \{1, \ldots, d\} \setminus \{k\}$. In [**3**] it is assumed that $\mathbb{C}^d$ has the orientation determined by the form $dr_1 \wedge \cdots \wedge dr_d \wedge d\theta_1 \wedge \cdots \wedge d\theta_d$ and that the boundary of D is assigned the orientation induced by that of D. This implies that $\partial D_{[k]}$ has the orientation determined by the form $(-)^{k-1}\, dr_{[k]} \wedge d\theta$. Therefore

$$\int_{\partial D_{[k]}} \varphi\, \omega_k(f, \overline{f}) = \rho_k \int_{[0,1]^{2d-1}} \frac{\varphi(z_1, \ldots, z_d)\, \det J_f(z_1, \ldots, z_d)\, \overline{\det J_{[k]}(z_1, \ldots, z_d)}}{\left(|f_1(z_1, \ldots, z_d)|^2 + \cdots + |f_d(z_1, \ldots, z_d)|^2\right)^d}$$
$$\times\, e^{2\pi i\,\theta_k}\, r_1 \cdots r_{k-1}\, r_{k+1} \cdots r_d\, dr_1 \cdots dr_{k-1}\, dr_{k+1} \cdots dr_d\, d\theta_1 \cdots d\theta_d$$

for $k = 1, \ldots, d$, where

$$z_k = z_k(\theta_k) = C_k + R_k\, e^{2\pi i\,\theta_k} \qquad 0 \le \theta_k \le 1$$

and

$$z_l = z_l(r_l, \theta_l) = C_l + r_l\, R_l\, e^{2\pi i\,\theta_l} \qquad 0 \le r_l, \theta_l \le 1$$

for $l \in \{1, \ldots, d\} \setminus \{k\}$. This proves Theorem 37.

3. The algorithm

In this section we will show how the zeros and their respective multiplicities can be computed by solving a generalized eigenvalue problem that has Hankel structure and d Vandermonde systems. Our approach is inspired by Theorems 3 and 4 and Equation (30). We have already explained why we prefer to use ordinary moments instead of modified moments.

First we introduce some notation. The total number of zeros of f that lie in D will be denoted by N. As explained in Section 1 we will assume that the value of N can be computed numerically. From now on, we will also suppose that $N > 0$. Let n be the number of mutually distinct zeros of f that lie in D. Let

$$(z_1^{(1)}, \ldots, z_d^{(1)}), \ldots, (z_1^{(n)}, \ldots, z_d^{(n)})$$

denote these zeros and let $\nu_1, \ldots, \nu_n$ be their respective multiplicities. Without loss of generality we may assume that $z_d^{(p)} \neq z_d^{(q)}$, $p \neq q$. Indeed, if one first applies a random unitary linear transformation to the unknowns $z = (z_1, \ldots, z_d)$, then this condition is satisfied almost surely, i.e., with probability one. Analogous results can be formulated in case $z_k^{(p)} \neq z_k^{(q)}$, $p \neq q$, for some $k \in \{1, \ldots, d-1\}$. We leave this to the reader. What happens if our algorithm is applied in case the dth components of the zeros of f that lie in D are not mutually distinct, will be discussed at the end of this section.

Define s_p for $p = 0, 1, 2, \ldots$ as

$$s_p := I(z_d^p)$$

where $I(\cdot)$ is defined in (49). We will assume that the sequence $(s_p)_{p\geq 0}$ can be computed numerically. By Theorem 37

$$s_p = \nu_1 [z_d^{(1)}]^p + \cdots + \nu_n [z_d^{(n)}]^p, \qquad p = 0, 1, 2, \ldots .$$

In particular, $s_0 = N$. Define

$$H_k := \begin{bmatrix} s_0 & s_1 & \cdots & s_{k-1} \\ s_1 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ s_{k-1} & \cdots & \cdots & s_{2k-2} \end{bmatrix}$$

for $k = 1, 2, \ldots$. Also, let V_n be the Vandermonde matrix with nodes $z_d^{(1)}, \ldots, z_d^{(n)}$,

$$V_n := \left[[z_d^{(l)}]^{k-1} \right]_{k,l=1}^{n} .$$

The following theorem characterizes n, the number of mutually distinct zeros, cf. Theorem 3.

THEOREM 40. *$n = \operatorname{rank} H_{n+p}$ for every nonnegative integer p. In particular, $n = \operatorname{rank} H_N$.*

PROOF. The proof is similar to that of Theorem 3. It is based on a factorization of the matrix H_n. The Vandermonde matrix V_n is regular since $z_d^{(1)}, \ldots, z_d^{(n)}$ are assumed to be mutually distinct. □

The dth components $z_d^{(1)}, \ldots, z_d^{(n)}$ of the zeros can be calculated by solving a generalized eigenvalue problem that has Hankel structure, cf. Theorem 4. Define

$$H_n^< := \left[s_{1+k+l} \right]_{k,l=0}^{n-1} .$$

THEOREM 41. *The eigenvalues of the pencil $H_n^< - \lambda H_n$ are given by $z_d^{(1)}, \ldots, z_d^{(n)}$.*

PROOF. The proof is similar to that of Theorem 4. □

Once $z_d^{(1)}, \ldots, z_d^{(n)}$ have been found, the multiplicities $\nu_1, \ldots, \nu_n$ can be computed by solving the Vandermonde system

$$V_n \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{bmatrix} . \tag{50}$$

REMARK. Theoretically the $N - n$ smallest singular values of H_N are equal to zero. In practice, as by evaluating the corresponding integrals numerically we can only obtain approximations for the s_p's and because of roundoff errors in the SVD computation, this will not be the case. If the numerical rank n of H_N is difficult to determine, it is safe to consider H_N as a matrix of full rank and to solve an $N \times N$ generalized eigenvalue problem and associated Vandermonde system.

THEOREM 42. *For every integer $\alpha \geq n$ the eigenvalues of the pencil*

$$\left[s_{1+k+l}\right]_{k,l=0}^{\alpha-1} - \lambda \left[s_{k+l}\right]_{k,l=0}^{\alpha-1} \tag{51}$$

are given by $z_d^{(1)}, \ldots, z_d^{(n)}$ and $\alpha - n$ eigenvalues that may assume arbitrary values.

PROOF. This follows from Theorem 41 and by taking into account that $(s_p)_{p\geq 0}$ is a linear recurring sequence:

$$s_{n+p} + \tau_1 s_{n-1+p} + \cdots + \tau_n s_p = 0, \qquad p = 0, 1, 2, \ldots,$$

if $\tau_1, \ldots, \tau_n$ are defined as the coefficients of the monic polynomial of degree n that has $z_d^{(1)}, \ldots, z_d^{(n)}$ as simple zeros,

$$\prod_{k=1}^{n} (z - z_d^{(k)}) =: z^n + \tau_1 z^{n-1} + \cdots + \tau_n.$$

Note that the latter implies that the matrices

$$\left[s_{1+k+l}\right]_{k,l=0}^{\alpha-1} \qquad \text{and} \qquad \left[s_{k+l}\right]_{k,l=0}^{\alpha-1}$$

have the same null spaces. □

The previous theorem tells us that the generalized eigenvalue problem (51) has $\alpha - n$ eigenvalues that may assume arbitrary values. Each of these indeterminate eigenvalues corresponds to two corresponding zeros on the diagonals of the generalized Schur decomposition of the Hankel matrices $\left[s_{1+k+l}\right]_{k,l=0}^{\alpha-1}$ and $\left[s_{k+l}\right]_{k,l=0}^{\alpha-1}$. When actually calculated, these diagonal entries are different from zero because of roundoff errors. The quotient of two such corresponding diagonal entries is an eigenvalue that is not the dth component of a zero of f. Fortunately, cf. the concluding paragraph of Example 4, the corresponding Vandermonde system enables us to detect such spurious "dth components." Indeed, the Vandermonde matrix whose nodes are given by the eigenvalues of (51) is still regular and therefore the corresponding Vandermonde system has only one solution, which gives the true dth components their correct corresponding multiplicity and the spurious ones "multiplicity" zero.

Once n, $z_d^{(1)}, \ldots, z_d^{(n)}$ and $\nu_1, \ldots, \nu_n$ have been found, the unknowns

$$z_1^{(1)}, \ldots, z_1^{(n)}, \ldots, z_{d-1}^{(1)}, \ldots, z_{d-1}^{(n)}$$

can be obtained as follows. Define $t_{k,p}$ for $k = 1, \ldots, d-1$ and $p = 0, 1, 2, \ldots$ as

$$t_{k,p} := I(z_k \, z_d^p).$$

We will assume that the sequences $(t_{k,p})_{p\geq 0}$ can be computed for $k = 1, \ldots, d-1$. By Theorem 37

$$t_{k,p} = \nu_1 \, z_k^{(1)} \, [z_d^{(1)}]^p + \cdots + \nu_n \, z_k^{(n)} \, [z_d^{(n)}]^p$$

for $k = 1, \ldots, d-1$ and $p = 0, 1, 2, \ldots$. It follows that $z_k^{(1)}, \ldots, z_k^{(n)}$ can be calculated from the solution of the Vandermonde system

$$V_n \begin{bmatrix} \nu_1 z_k^{(1)} \\ \nu_2 z_k^{(2)} \\ \vdots \\ \nu_n z_k^{(n)} \end{bmatrix} = \begin{bmatrix} t_{k,0} \\ t_{k,1} \\ \vdots \\ t_{k,n-1} \end{bmatrix}, \qquad k = 1, \ldots, d-1.$$

As already discussed in Chapter 1, problems of numerical linear algebra that involve Vandermonde or Hankel matrices truly deserve their reputation of being ill-conditioned [**98, 271**]. By moving the origin in the z_d-plane to the arithmetic mean of the dth components of the zeros,

$$z_d' := \frac{\nu_1 z_d^{(1)} + \cdots + \nu_n z_d^{(n)}}{\nu_1 + \cdots + \nu_n} = \frac{I(z_d)}{I(1)},$$

ill-conditioning is reduced significantly. Therefore we will use shifted versions of the integrals s_p and $t_{k,p}$, denoted by $\hat{s}_p$ and $\hat{t}_{k,p}$. The results of this section then lead to the following algorithm.

ALGORITHM

1. $N \leftarrow I(1)$
2. $z_d' \leftarrow I(z_d)/N$
3. $\hat{s}_0 \leftarrow N; \quad \hat{s}_1 \leftarrow 0; \quad \hat{s}_p \leftarrow I(\,(z_d - z_d')^p\,)$ for $p = 2, \ldots, 2N-1$
4. $n \leftarrow \operatorname{rank} \left[\hat{s}_{k+l}\right]_{k,l=0}^{N-1}$
5. Calculate the eigenvalues $\lambda_1, \ldots, \lambda_n$ of the pencil $\left[\hat{s}_{1+k+l}\right]_{k,l=0}^{n-1} - \lambda \left[\hat{s}_{k+l}\right]_{k,l=0}^{n-1}$.
6. $z_d^{(k)} \leftarrow \lambda_k + z_d'$ for $k = 1, \ldots, n$
7. Solve the Vandermonde system

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \ldots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} = \begin{bmatrix} \hat{s}_0 \\ \hat{s}_1 \\ \vdots \\ \hat{s}_{n-1} \end{bmatrix}.$$

8. $\hat{t}_{k,p} \leftarrow I(\,z_k(z_d - z_d')^p\,)$ for $k = 1, \ldots, d-1$ and $p = 0, \ldots, n-1$
9. Solve the Vandermonde system

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_n \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \ldots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} x_{1,1} & \ldots & x_{d-1,1} \\ x_{1,2} & \ldots & x_{d-1,2} \\ \vdots & & \vdots \\ x_{1,n} & \ldots & x_{d-1,n} \end{bmatrix} = \begin{bmatrix} \hat{t}_{1,0} & \ldots & \hat{t}_{d-1,0} \\ \hat{t}_{1,1} & \ldots & \hat{t}_{d-1,1} \\ \vdots & & \vdots \\ \hat{t}_{1,n-1} & \ldots & \hat{t}_{d-1,n-1} \end{bmatrix}.$$

$z_k^{(l)} \leftarrow x_{k,l}/\nu_l$ for $k = 1, \ldots, d-1$ and $l = 1, \ldots, n$

In practice, as explained in the previous remark, we may dispense with step 4 and solve an $N \times N$ generalized eigenvalue problem and associated Vandermonde system. The components of the solution of the latter will be rounded to the nearest integers and eigenvalues that have "multiplicity" zero are thrown away.

REMARK. What happens if our algorithm is applied in case the dth components of the zeros of f that lie in D are not mutually distinct? In this case only a subset of the zeros will be determined correctly, namely the zeros whose dth component occurs only once. To illustrate what happens to the other zeros, suppose for example that $z_d^{(1)} = z_d^{(2)}$ whereas $z_d^{(3)}, \ldots, z_d^{(n)}$ are mutually distinct and

$$\{ z_d^{(1)} \} \cap \{ z_d^{(3)}, \ldots, z_d^{(n)} \} = \emptyset.$$

Then

$$s_p = (\nu_1 + \nu_2)\, [z_d^{(1)}]^p + \nu_3\, [z_d^{(3)}]^p + \cdots + \nu_n\, [z_d^{(n)}]^p$$

for $p = 0, 1, 2, \ldots$ and

$$t_{k,p} = (\nu_1\, z_k^{(1)} + \nu_2\, z_k^{(2)})\, [z_d^{(1)}]^p + \nu_3\, z_k^{(3)}\, [z_d^{(3)}]^p + \cdots + \nu_n\, z_k^{(n)}\, [z_d^{(n)}]^p$$

for $k = 1, \ldots, d-1$ and $p = 0, 1, 2, \ldots$. It follows that our algorithm will replace each group of zeros that have the same dth component by a point in $\mathbb{C}^d$ whose dth component is equal to the dth component that is shared by these zeros and whose other components are given by the arithmetic mean of the corresponding components of these zeros.

4. Numerical examples

Let us discuss a few numerical examples.

EXAMPLE 19. We have chosen $d = 4$ (the number of variables) and $n = 5$ (the number of mutually distinct zeros). The zeros and their respective multiplicities are listed in the following table.

$(z_1^{(k)}, \ldots, z_d^{(k)})$	ν_k
$(\ 1,\ \ i, -2,\ 1)$	1
$(-3,\ \ 0,\ \ 2,\ 2)$	2
$(\ 2, -i,\ \ 5,\ 3)$	1
$(\ 1,\ \ 0, -1,\ 4)$	2
$(\ 0,\ \ 1,\ \ 3,\ 5)$	1

Then $N = 7$ (the total number of zeros) and $z'_d = (1 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + 1 \cdot 5)/7 = 3$ (the arithmetic mean of the last components). The shifted moments for the last components are given by

$$\hat{s}_p = 1 \cdot (1-3)^p + 2 \cdot (2-3)^p + 1 \cdot (3-3)^p + 2 \cdot (4-3)^p + 1 \cdot (5-3)^p$$

and thus $\hat{s}_p = (2 + 2^p)(1 + (-1)^p)$ for $p \geq 1$. We computed the eigenvalues of the pencil

$$\left[\hat{s}_{1+k+l}\right]_{k,l=0}^{N-1} - \lambda\, \left[\hat{s}_{k+l}\right]_{k,l=0}^{N-1}$$

using LAPACK's routine CGEGV [**12**]. The calculations were performed in single precision arithmetic on an IBM SP2. The results are shown in the following table, in which each generalized eigenvalue is represented as a pair (α_k, β_k) of two corresponding diagonal entries in the generalized Schur decomposition of $\left[\hat{s}_{1+k+l}\right]_{k,l=0}^{N-1}$ and $\left[\hat{s}_{k+l}\right]_{k,l=0}^{N-1}$.

α_k	β_k	$z_d^{(k)}$	ν_k
-30316.83	15158.42	1.000000	1.000001
-2322.758	2322.756	1.999999	1.999996
0.003323	542.2963	3.000006	1.000017
950.4792	950.4736	4.000006	1.999985
33612.10	16806.05	5.000000	0.999998
0.000000	0.000233		
0.000000	0.000102		

The number of mutually distinct zeros is clearly equal to five. The generalized eigenvalues α_k/β_k corresponding to the last two pairs (α_k, β_k) represent spurious dth components and are thrown away. Also shown in this table are the computed values $z_d' + \alpha_k/\beta_k$ of the dth components and the corresponding solutions of the Vandermonde system (50) for the multiplicities. The computed approximations for the other components of the zeros are shown in the following table.

$z_1^{(k)}$	$z_2^{(k)}$	$z_3^{(k)}$
$1.000002 - i\,0.000000$	$0.000000 + i\,1.000001$	$-2.000006 - i\,0.000000$
$-2.999997 + i\,0.000000$	$-0.000000 - i\,0.000002$	$2.000013 + i\,0.000001$
$2.000010 - i\,0.000000$	$0.000000 - i\,1.000002$	$4.999987 - i\,0.000002$
$0.999992 + i\,0.000000$	$-0.000001 + i\,0.000002$	$-1.000007 + i\,0.000001$
$-0.000002 + i\,0.000000$	$1.000000 - i\,0.000000$	$3.000006 - i\,0.000000$

The previous example illustrates how our algorithm is able to compute the unknown zeros and multiplicities from the integrals $\hat{s}_p$ and $\hat{t}_{k,p}$. As already mentioned in Section 1, the efficient numerical evaluation of these integrals is a problem that remains to be tackled. It represents a challenge for the numerical integration community. Nevertheless we would like to present a small example in which these integrals have been calculated numerically.

EXAMPLE 20. Consider the problem of computing all the zeros of

$$f = f(z_1, z_2) = (\sin z_1 + z_1^2 + e^{z_2} - \cos(2z_2), \cos z_1 + z_2^3 + e^{2z_2} - 2)$$

that lie in the polydisk

$$D = \{z_1 \in \mathbb{C} \,:\, |z_1| < 1\} \times \{z_2 \in \mathbb{C} \,:\, |z_2| < 1\}.$$

In this case we have to integrate over the unit cube in $\mathbb{R}^3$. We tried several numerical integration strategies:

- lattice rules (see for example Joe and Sloan [**164, 165, 255**], Beckers and Haegemans [**22**] and also Sidi [**254**] and Laurie [**197**]);
- Monte Carlo methods (see, e.g., Kalos and Whitlock [**171**] and Fishman [**81**]);
- the software package DCUHRE [**26**], written by Berntsen, Espelid and Genz.

DCUHRE gave the best results. This package implements an adaptive algorithm for numerical integration over hyperrectangular regions. First we calculated $\mathrm{Re}\, I(1)$. We requested an absolute accuracy of 0.1 and obtained that $\mathrm{Re}\, I(1) \approx 1.998$. Thus $N = 2$. Next we calculated the arithmetic mean of the second components. A crude approximation for this mean is sufficient to reduce ill-conditioning and therefore we requested a relative accuracy of only 0.1. Finally we calculated all the other integrals needed by our algorithm. One of the very interesting features of DCUHRE

is that it is able to integrate a *vector* of similar integrals over a common integration region. Since a significant part of the computation required for each integrand is the same for all of the integrands, these common calculations need to be done only once for each integrand evaluation point. We requested a relative accuracy of 10^{-5}. DCUHRE needed about 10^5 functional evaluations to obtain this accuracy. With these approximations for the integrals $\hat{s}_p$ and $\hat{t}_{k,p}$ as input, our algorithm obtained that f has two zeros in D ($n = 2$), each of multiplicity one. We refined the approximations for these zeros iteratively via Newton's method. The zeros of f that lie in D are given by $(0, 0)$ and $(-0.72011062161456, 0.11033979708375)$. $\diamondsuit$

Acknowledgements

I had interesting conversations with Franki Dillen about differential forms and stimulating discussions with Ronald Cools and Alan Genz on the logarithmic residue integrals that appear in this chapter. I would like to thank Johan Quaegebeur for reading the proof of Theorem 37.

Part 2

CHAPTER 5

Computing simple zeros of Bessel functions

In this chapter, we consider the problem of computing simple zeros of analytic functions. In particular, we focus on the Bessel functions $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ and $H_\nu^{(2)}(z)$, or their first derivative, in case the argument $z \in \mathbb{C} \setminus (-\infty, 0]$ and the order $\nu \in \mathbb{R}$. We present a software package, called ZEBEC[1], for computing all the zeros of one of these functions that lie inside a rectangle whose edges are parallel to the coordinate axes. In Chapter 6 we will consider the functions $J_n(z) \pm iJ_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ in case $n \in \mathbb{N}$. The zeros of these functions play an important role in certain physical applications. We will prove that all the zeros that lie in $\mathbb{C}_0$ are simple. As ZEBEC can be easily extended to calculate zeros of any analytic function, provided that the zeros are known to be simple, it is the package of choice to calculate the zeros of $J_n(z) \pm iJ_{n+1}(z)$ or $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$. Numerical examples will be given.

This chapter corresponds to our paper [**185**].

1. Introduction

The Bessel equation

$$z^2u''(z) + zu'(z) + (z^2 - \nu^2)u(z) = 0, \qquad z, \nu \in \mathbb{C}, \tag{52}$$

appears in many problems of mathematical physics. Two linearly independent solutions are given by the *Bessel function of the first kind* of order ν

$$J_\nu(z) := \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\,\Gamma(k+\nu+1)} \Big(\frac{z}{2}\Big)^{\nu+2k}, \qquad z \in \mathbb{C} \setminus (-\infty, 0],$$

and the *Bessel function of the second kind* of order ν (also called *Neumann's function*)

$$Y_\nu(z) := \frac{J_\nu(z)\cos(\nu\pi) - J_{-\nu}(z)}{\sin(\nu\pi)}, \qquad z \in \mathbb{C} \setminus (-\infty, 0], \quad \nu \in \mathbb{C} \setminus \mathbb{Z}.$$

For integer ν the right-hand side becomes indeterminate, and in this case

$$Y_n(z) := \lim_{\nu \to n} Y_\nu(z), \qquad n \in \mathbb{Z}.$$

Also of interest are the *Bessel functions of the third kind*, or *Hankel functions*,

$$H_\nu^{(1)}(z) := J_\nu(z) + i\,Y_\nu(z) \quad \text{and} \quad H_\nu^{(2)}(z) := J_\nu(z) - i\,Y_\nu(z). \tag{53}$$

[1]Our package is part of the CPC Program Library. Information on how to obtain ZEBEC can be found at url `http://www.cpc.cs.qub.ac.uk/cpc/cgi-bin/list_summary.pl/?CatNumber=ADIO`.

The above considered functions are analytic with respect to z in the complex plane cut along the non-positive real axis and entire functions of the order ν for fixed z. They also have several other interesting features [**295**].

The zeros and turning points of Bessel functions are important in many branches of physical sciences and technology. They appear in the problem of cyclic membrane vibrations, the temperature distribution in a solid cylinder or in a solid sphere, the diffraction of a plane electromagnetic wave by a conducting cylinder, quantum billiards, etc. For an electrostatic interpretation of the zeros of Bessel functions, we refer to [**220**]. See also [**219**].

Some theoretical results about these zeros and turning points are available: inequalities, bounds, regions of existence or non-existence, power series expansions, Chebyshev series expansions, and qualitative results concerning their location [**2, 24, 56, 151, 152, 153, 179, 180, 198, 236, 237, 238, 239, 295**].

Hurwitz's Theorem, for example, gives information about the zeros of $J_\nu(z)$ in case the order ν is real.

THEOREM 43. *Let ν be an arbitrary real number. Then the function $J_\nu(z)$ has an infinite number of positive real zeros, and a finite number $2N(\nu)$ of conjugate complex zeros, where:*

- $N(\nu) = 0$ *if* $\nu > -1$ *or* $\nu = -1, -2, \ldots,$
- $N(\nu) = m$ *if* $-(m+1) < \nu < -m, m = 1, 2, \ldots.$

In the second case, if $[-\nu]$ is odd, then there is a pair of purely imaginary zeros among the conjugate complex zeros.

In [**240, 242**] a software package was presented for computing zeros of $J_\nu(x)$, in case $x > 0$ and $\nu > -1$, and turning points of $J_\nu(x)$, i.e., zeros of $J'_\nu(x)$, in case $x > 0$ and $\nu > 0$. The package RFSFNS [**291**] can be used to calculate zeros and turning points of $J_\nu(x)$ and $Y_\nu(x)$, in case $x > 0$ and $\nu \geq 0$. Recently, Segura and Gil [**253**] have presented two programs to calculate real zeros of $J_\nu(x)$ for real orders ν (positive or negative). Their algorithm proceeds by applying Newton's method to the monotonic function $x^{2\nu-1}J_\nu(x)/J_{\nu-1}(x)$. To evaluate this function, they use the continued fraction representation of the ratio $J_\nu(x)/J_{\nu-1}(x)$ and apply the Lentz-Thompson algorithm. Ikebe et al. [**154, 155**] showed how the complex zeros of $J_\nu(z)$ can be calculated from an eigenvalue problem for infinite compact complex symmetric matrices. In this chapter we present a reliable and portable package for computing zeros or turning points of $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ and $H_\nu^{(2)}(z)$, in case the argument z belongs to the complex plane cut along the non-positive real axis and the order ν is real. The package is called ZEBEC: 'ZEros of BEssel functions Complex.' ZEBEC is capable of calculating all the zeros or turning points that lie inside a rectangle whose edges are parallel to the coordinate axes. It is known that any solution of (52) has only simple zeros, except possibly at $z = 0$ (see, e.g., [**118**, p. 79] or [**295**, p. 479]). In particular, this holds for $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ and $H_\nu^{(2)}(z)$. Besides, the derivative of any solution of (52) also has only simple zeros, except possibly at $z = 0$ or $z = \pm\nu$ [**175, 176, 267**]. Therefore, the zeros that we set out to compute are simple, except possibly at $z = 0$ and $z = \pm\nu$. As we restrict the argument z to $\mathbb{C} \setminus (-\infty, 0]$ and the order ν to $\mathbb{R}$, we do not have to

consider the point $z = 0$ and one of the points $z = \pm\nu$. By evaluating the function and its first derivative at $z = \nu$ (in case $\nu > 0$) or $z = -\nu$ (in case $\nu < 0$), one can easily verify whether the function has a multiple zero at this point. If this is indeed the case, then ZEBEC notifies the user and asks for a different rectangular region. Else, the computation starts.

2. Computing simple zeros of analytic functions

Given a rectangle whose edges are parallel to the coordinate axes, we take the following approach.

(1) We calculate the total number of zeros that lie inside this rectangle.
(2) We isolate all these zeros via consecutive subdivisions. In other words, we obtain a set of subrectangles, each of which contains precisely one zero.
(3) For each of these subrectangles, we calculate the zero that lies inside it.

For the first two phases we use logarithmic residue integrals, whereas for the third phase we use a generalized method of bisection. All these ingredients will be briefly explained below.

Let W be a simply connected region in $\mathbb{C}\backslash(-\infty, 0]$, f one of the Bessel functions mentioned in Section 1, and γ a positively oriented Jordan curve in W that does not pass through any zero of f. As f is analytic in W, the total number of zeros of f that lie inside γ is given by

$$N := \frac{1}{2\pi i} \int_\gamma \frac{f'(z)}{f(z)}\, dz.$$

Suppose that a parametric representation of the curve γ is given by $z = \gamma(t)$, $a \le t \le b$, and that the mapping $\gamma : [a, b] \to \mathbb{C}$ is piecewise continuously differentiable in $[a, b]$. Thus, a partition $a = t_0 < t_1 < \cdots < t_{r-1} < t_r = b$ of the interval $[a, b]$ can be found, such that the restriction of γ to each subinterval $[t_{j-1}, t_j]$ is continuously differentiable.

Let $W^\star := \{\, (x, y) \in \mathbb{R}^2 \;:\; x + i\,y \in W \,\}$ and $u, v : W^\star \to \mathbb{R}$, with

$$u(x, y) := \operatorname{Re} f(x + i\,y) \qquad \text{and} \qquad v(x, y) := \operatorname{Im} f(x + i\,y).$$

By restricting u and v to the curve γ, we obtain

$$u(t) := u\big(x(t), y(t)\big) \qquad \text{and} \qquad v(t) := v\big(x(t), y(t)\big),$$

where $x(t) := \operatorname{Re}\gamma(t)$, $y(t) := \operatorname{Im}\gamma(t)$ and $t \in [a, b]$. Then, by using the Cauchy-Riemann equations for u and v, one can easily verify that

$$N = \frac{1}{2\pi} \sum_{j=1}^{r} \int_{t_{j-1}}^{t_j} \frac{u\,\frac{dv}{dt} - v\,\frac{du}{dt}}{u^2 + v^2}\, dt,$$

where we have considered the above partition of $[a, b]$. This formula for N is the Kronecker integral for the topological degree of the real mapping $F = (u, v)$ at the origin relative to the interior of γ. For an introduction to degree theory we refer the interested reader to Lloyd [**201**]. Also, for a detailed description of the Kronecker integral, see Hoenders and Slump [**144, 145**].

Let us consider the case that γ is a rectangle whose edges are parallel to the coordinate axes. Suppose that it has left lower vertex (x_0, y_0) and that its edges have length h_1 and h_2. In other words, suppose that γ is the boundary of

$$[x_0, x_0 + h_1] \times [y_0, y_0 + h_2].$$

Define the functions ψ and φ in $W^\star$ as

$$\psi(x, y) := \frac{u(x, y)\, v_x(x, y) - v(x, y)\, u_x(x, y)}{[u(x, y)]^2 + [v(x, y)]^2}$$

and

$$\varphi(x, y) := \frac{u(x, y)\, v_y(x, y) - v(x, y)\, u_y(x, y)}{[u(x, y)]^2 + [v(x, y)]^2}.$$

Then one can easily verify that

$$N = h_1 \left(I_1 - I_3\right) + h_2 \left(I_2 - I_4\right)$$

with

$$\begin{aligned}
I_1 &:= \frac{1}{2\pi} \int_0^1 \psi(x_0 + t h_1, y_0)\, dt \\
I_2 &:= \frac{1}{2\pi} \int_0^1 \varphi(x_0 + h_1, y_0 + t h_2)\, dt \\
I_3 &:= \frac{1}{2\pi} \int_0^1 \psi(x_0 + t h_1, y_0 + h_2)\, dt \\
I_4 &:= \frac{1}{2\pi} \int_0^1 \varphi(x_0, y_0 + t h_2)\, dt.
\end{aligned}$$

These formulae are the basis of the phases (1) and (2) of our method.

As f may have zeros on the boundary of the rectangular box specified by the user, the algorithm starts by perturbing this box. For this purpose a tolerance is used that is taken to be proportional to a power of the machine precision, for example 10 times the square root of the machine precision. The box is then slightly enlarged in an asymmetrical way. The reason for this asymmetric perturbation is to lower the possibility of having a zero close to or on any boundary of the consecutive subdivisions. For example, if the starting box is symmetric with respect to the imaginary axis, then the inner boundary at the first subdivision will pass through any imaginary zeros of f.

The total number of zeros of f that lie inside the perturbed box is obtained by calculating the integrals $I_1, \ldots, I_4$ via the adaptive integrator DQAG from the package QUADPACK [**241**]. A zero near one of the edges of the rectangle causes the integrand of the corresponding integral to have a "peak." The closer the zero lies to the edge, the sharper this peak is. If the zero lies on the edge, then the integral is divergent. DQAG uses adaptive strategies that enable it to cope with such peaks efficiently. However, if a zero lies too close to an edge (the corresponding peak is too sharp), then DQAG warns us that it has problems in calculating the integral. Our algorithm then slightly moves this edge and restarts. By enlarging the user's

box, we may of course include additional zeros. We have decided not to discard any of these zeros ourselves. Rather, we provide the user with the box that eventually has been considered, all the zeros that lie inside this box, and leave it to him/her to filter out unwanted zeros.

If the starting box (as perturbed by the method) contains exactly one zero, then this zero is calculated via the package CHABIS [**288, 289**]. This package implements a generalized method of bisection, known as *characteristic bisection*, which we will explain below. Otherwise, the longest edges of the box are halved and the box is subdivided into two equal boxes. The number of zeros in each of these boxes is calculated via numerical integration. If DQAG detects a zero near the inner edge, then this edge is shifted, a process that results in an asymmetric subdivision of the box. Then the two smaller boxes are examined. A box that does not contain any zero is abandoned. A box that contains precisely one zero is handed to CHABIS. A box that contains more than one zero is subdivided again. This process is repeated until all the zeros have been isolated — a set of boxes has been found, each of which contains precisely one zero — and computed.

The package CHABIS can be used to solve systems of nonlinear equations. It is based on the topological degree. Let us briefly describe how it works. First, what is known as a *characteristic polyhedron* (CP) is constructed. Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be the mapping that defines the nonlinear system. Every polyhedron that has 2^d vertices and is such that the signs of the components of F evaluated at these vertices produce all the possible combinations of -1 and 1 is by definition a CP for F. Under certain assumptions on the boundary of a CP, this property implies that the topological degree of F at the origin relative to the interior of the CP is different from zero. Kronecker's Theorem [**201**] then implies that F has at least one zero in the interior of the CP. Note that the topological degree of F relative to the interior of the CP does not have to computed explicitely. Given a CP for F, a smaller CP can be constructed by replacing a certain vertex of the CP by the midpoint of its longest edge. The vertex that will be replaced is such that the signs of the components of F at this vertex coincide with those at the midpoint of the longest edge. The new polyhedron is again a CP, and thus it also contains at least one zero of F. This refinement process, which is called *characteristic bisection*, is repeated until the diameter of the polyhedron is smaller than a given tolerance.

The approach taken by CHABIS has several advantages. First, as soon as an initial characteristic polyhedron has been found, the convergence is guaranteed. The zeros of F have to be regular though. (A zero $x^\star \in \mathbb{R}^d$ of F is called regular if the Jacobian matrix of F at $x^\star$ is nonsingular.) Otherwise, the theory of the topological degree is not applicable. A second advantage is that the number of iterations needed to approximate a zero to a certain accuracy is known a priori. Furthermore, as the method depends only on the signs of the components of F, it is suitable for solving problems with imprecise function values or involving infinite series expansions. For example, in the case of Bessel and Airy functions Vrahatis et al. [**290, 293**] showed that the sign stabilizes after summing a relatively small number of terms of the power series, which leads to a considerable speed-up of the calculations. A theoretical approach to the problem of computing complex zeros of Bessel functions by using only the signs of the respective power series was presented in [**292**].

As already mentioned in Section 1, the functions $J'_\nu(z)$, $Y'_\nu(z)$, $H_\nu^{(1)\prime}(z)$ and $H_\nu^{(2)\prime}(z)$ may have a multiple zero at $z = \pm\nu$. As our algorithm can manage only simple zeros, we have to examine if such a multiple zero exists inside the rectangular region specified by the user. This can be done very easily: if the rectangular box contains $z = \nu$ (in case $\nu > 0$) or $z = -\nu$ (in case $\nu < 0$), then the algorithm evaluates the function and its first derivative at this point. If these values are both very small, then ZEBEC warns the user and asks for a different rectangular region. Else, the computation starts.

3. The packages BESSCC, QUADPACK and CHABIS

In this section we give a brief outline of the packages that ZEBEC uses: BESSCC, QUADPACK and CHABIS.

BESSCC. This package was written by Thompson and Barnett [**268**]. It computes the modified Bessel functions $I_\nu(z)$ and $K_\nu(z)$ as well as their first derivative for complex argument z and real order ν. To obtain $J_\nu(z)$ from these functions, we have used the following analytic continuation and reflection formulae [**2**]: if $\nu \geq 0$, then

$$J_\nu(z) = \begin{cases} e^{i\frac{\pi}{2}\nu} I_\nu(-iz) & \text{if } \operatorname{Im} z \geq 0, \\ e^{-i\frac{\pi}{2}\nu} I_\nu(iz) & \text{if } \operatorname{Im} z < 0, \end{cases}$$

else

$$J_\nu(z) = J_{-\nu}(z)\cos(\pi\nu) + Y_{-\nu}(z)\sin(\pi\nu).$$

For $Y_\nu(z)$ the following holds: if $\nu \geq 0$, then

$$Y_\nu(z) = \begin{cases} ie^{i\frac{\pi}{2}\nu} I_\nu(-iz) - \frac{2}{\pi} e^{-i\frac{\pi}{2}\nu} K_\nu(-iz) & \text{if } \operatorname{Im} z \geq 0, \\ \overline{Y_\nu(\overline{z})} & \text{if } \operatorname{Im} z < 0, \end{cases}$$

else

$$Y_\nu(z) = Y_{-\nu}(z)\cos(\pi\nu) - J_{-\nu}(z)\sin(\pi\nu).$$

We have evaluated the Bessel functions of the third kind by using their definition (53). To obtain formulae for the derivatives, we have differentiated the previous identities.

For compatibility purposes, and with the permission of the authors, we have also made a few small changes to BESSCC.

QUADPACK. This package was written by Piessens et al. [**241**]. It is a widely used package for automatic integration. It consists of 12 quadrature routines. Here the routine DQAG is used, which implements a globally adaptive integrator for calculating definite integrals over finite intervals. DQAG is based on Gauss-Kronrod quadrature rules. QUADPACK is written in ANSI standard Fortran 77.

CHABIS. This package was written by Vrahatis [**288**]. It can be used to solve nonlinear systems of equations in case the only computable information consists of the algebraic signs of the components of the mapping that defines the nonlinear system. CHABIS starts by locating at least one solution that lies inside a polyhedron. Then it applies a generalized method of bisection to this polyhedron. The package

consists of nine subprograms, one of which is called by the user. CHABIS is written in ANSI standard Fortran 77.

4. The package ZEBEC

The package ZEBEC ('ZEros of BEssel functions Complex') contains about 7000 lines of code including comments. It is written in Fortran 77 and has been thoroughly tested on various UNIX machines (DEC DS5000/7012, HP 9000 B160L, IBM RS6000 7012, SUN SPARC Ultra-2 m1170) as well as on a PC IBM compatible.

ZEBEC consists of seven parts, namely, the main program `ZEBEC`, the subroutines `MANAGE`, `INBOX`, `SPLIT`, `RCOMP`, and `FDF` and the function `FNC`, and a set of functions related to the integrals $I_1, \ldots, I_4$. ZEBEC also requires the subroutine DQAG from the package QUADPACK [**241**], the package CHABIS [**288**] and the package BESSCC [**268**] from the CPC Program Library.

In the main program `ZEBEC` the following parameters have to be set:

`MAXRT`: a positive integer that determines the maximum number of zeros that may be requested.

`ICASE`: an integer in $\{1, \ldots, 8\}$ that specifies which function is to be considered:

1. J, the Bessel function of the first kind,
2. the derivative of J,
3. Y, the Bessel function of the second kind,
4. the derivative of Y,
5. $H^{(1)} = J + iY$, the Bessel function of the third kind,
6. the derivative of $H^{(1)}$,
7. $H^{(2)} = J - iY$, the Bessel function of the third kind,
8. the derivative of $H^{(2)}$.

`XNU`: a real variable that specifies the order of the Bessel function.

`XO`: a real array of length 2 that contains the x- and y-coordinates of the left lower vertex of the rectangle that is to be examined.

`H`: a real array of length 2 that specifies the size of this rectangle along the x- and y-direction.

`ICON`: an integer in $\{1, \ldots, 4\}$ that specifies which calculations are to be done:

1. calculation of the total number of zeros, only,
2. calculation of the total number of zeros and isolation of each one of them,
3. calculation of the total number of zeros, isolation and computation of each one of them,
4. calculation of the total number of zeros; isolation and computation of `NR` zeros.

Note that if `ICON=4`, then the user also has to supply the value of the requested number of zeros `NR`.

`EPSILO:` a real variable that is used in the stopping criterion for the computation of the zeros. Termination occurs if the algorithm estimates that the infinity norm of the mapping $F := (u, v)$, where $u := \mathrm{Re}\, f$ and $v := \mathrm{Im}\, f$ and f is one of the Bessel functions or their first derivative, at an approximate zero is at most `EPSILO` or if the size of the box that contains an approximate zero is at most `4.D0*EPSILO`.

If `EPSILO` is set to a value that is less than the machine precision `EPSMCH`, then `EPSILO` is set equal to `5.D0*EPSMCH`. `EPSMCH` is computed within ZEBEC.

`EPSABS:` a real variable that determines the absolute accuracy to which the integrals are to be evaluated. If `EPSABS=0.D0`, then only a relative precision criterion will be used.

`EPSREL:` a real variable that determines the relative accuracy to which the integrals are to be evaluated. If `EPSREL=0.D0`, then only an absolute precision criterion will be used.

If `EPSABS` and `EPSREL` are both too small, then the numerical integration may be time consuming. If they are both too large, then the calculated number of zeros may be wrong. The default values of `EPSABS` and `EPSREL` are `0.07D0` and `0.0D0`, respectively.

`ACC:` a variable whose value determines a target relative accuracy for BESSCC. If `ACC` is larger than `0.0001` or less than the machine precision, then it is set equal to its default value `ACCDEF=10E-6`.

Let us briefly describe the various parts of ZEBEC.

The subroutine `INBOX` calculates the total number of zeros that lie inside the rectangular box given by the user. If some of the zeros lie too close to the boundary of this box and the quadrature routine DQAG fails, then `INBOX` perturbs the box slightly and enlarges it.

The subroutine `SPLIT` takes a box and splits it into two boxes. A symmetric splitting, which proceeds by halving the longest edges, is tried first. If the calculation of the integral along the inner edge fails, then it is assumed that some of the zeros lie too close to this edge, and the inner edge is shifted.

The subroutine `RCOMP` takes a box that contains exactly one zero and returns an approximation for this zero. `RCOMP` calls CHABIS.

The subroutine `MANAGE` forms the main part of the package. `MANAGE` starts by calling `INBOX`. If there are no zeros inside the user's box, then the program stops. If there is exactly one zero inside this box, then `RCOMP` is called (if `ICON=3` or `4`). Else, the box is given to `SPLIT`. The two boxes returned by `SPLIT` are examined. A box that does not contain any zero is abandoned. A box that contains exactly one zero is given to `RCOMP` (if `ICON=3` or `4`). A box that contains more than one zero is put in a list. Then `MANAGE` takes the next box from this list and calls `SPLIT`. This procedure is repeated until all the requested zeros are isolated (if `ICON=2`, `3` or `4`) and computed (if `ICON=3` or `4`).

DQAG requires function and derivative values. These are provided by the subroutine `FDF`. CHABIS needs only function values. These are provided by the function `FNC`. Both `FDF` and `FNC` call BESSCC.

The program execution terminates normally after the completion of its task. This type of termination is indicated by the value 1 of the output variable INFO. If the value of this parameter is different from 1, then the termination of the program is abnormal. The cases of abnormal termination are the following:

INFO=0: Improper input parameters:

- the input values of ICASE or ICON are out of range, or
- NR exceeds MAXRT, or
- the box specified by X0 and H crosses the non-positive real axis, or
- the box specified by the user contains a multiple zero at $z = \nu$ or $z = -\nu$ (ICASE=2, 4, 6 or 8), or
- H(1) or H(2) is negative, or
- EPSABS or EPSREL is negative.

INFO=2: The procedure for the calculation of the total number of zeros has failed.

INFO=3: The procedure for the isolation of the zeros has failed.

INFO=4: The procedure for the computation of the zeros has failed.

Upon normal termination, the main output values of the program are given by the following parameters of MANAGE:

NRPERT: an integer that is equal to the number of zeros that exist in the examined box in case ICON is not equal to 3.

LFLRT: an integer that is equal to the number of zeros that exist in the examined box in case ICON is equal to 3.

XOFIN: a real array of size 2×MAXRT that contains the x- and y-coordinates of the left lower vertices of the rectangles that have been found to contain exactly one zero.

HFIN: a real array of size 2×MAXRT that specifies the size of these rectangles along the x- and y-direction.

ROOTS: a real array of size 2×MAXRT that contains the real and imaginary parts of the zeros that have been found in the examined box.

FROOTS: a real array of size 2×MAXRT that contains the values of the function (real and imaginary parts) at the approximations for the zeros that have been found.

5. An example of how to use ZEBEC

Let us demonstrate how ZEBEC can be used to calculate the total number of zeros that lie inside a rectangular box, to isolate these zeros, and to compute them. Suppose that we want to calculate five zeros of $Y_{-15.3}(z)$ that lie inside the box

$$\{\, z \in \mathbb{C} \,:\, -22 \le \mathrm{Re}\, z \le 23, \quad 0.5 \le \mathrm{Im}\, z \le 100.5 \,\}.$$

The corresponding input values are: ICASE=3, XNU=-15.3D0, ICON=4, NR=5, and X0(1)=-22.D0, X0(2)=0.5D0, H(1)=45.D0, H(2)=100.D0. We request an accuracy of EPSILO=1.0D-13.

For this example the main program of ZEBEC is the following:

```
      PROGRAM ZEBEC
      IMPLICIT NONE

      INTEGER MAXRT
      PARAMETER (MAXRT = 100)

      INTEGER IFLRT(MAXRT), ICASE, ICON, NR, INFO, NRKEEP, IBESS
      INTEGER J, LFLRT, NRPERT, IMAXRT

      DOUBLE PRECISION XO(2), H(2), XOPERT(2), HPERT(2)
      DOUBLE PRECISION ROOTS(2,MAXRT), FROOTS(2,MAXRT)
      DOUBLE PRECISION POINTS(2,MAXRT), STEPS(2,MAXRT), NRS(MAXRT)
      DOUBLE PRECISION XOFIN(2,MAXRT), HFIN(2,MAXRT)
      DOUBLE PRECISION RINTS(6,MAXRT), ERRS(6,MAXRT)
      DOUBLE PRECISION EPSILO, XNU, EPSABS, EPSREL, ACC

      EXTERNAL MANAGE

      COMMON /BLK1/ ICON, NR
      COMMON /BLK2/ INFO
      COMMON /BLK3/ XNU
      COMMON /BLK4/ ICASE
      COMMON /BLK5/ IBESS
      COMMON /BLK6/ EPSABS, EPSREL
      COMMON /BLK7/ ACC
      COMMON /BLK8/ IMAXRT

      DATA   ICASE,    XNU,      ICON,    NR
     +     / 3,        -15.3D0,  4,       5 /
      DATA   XO(1),    XO(2),    H(1),    H(2)
     +     / -22.D0,   0.5D0,    45.D0,   100.D0 /

      EPSILO = 1.0D-13

      EPSABS = 0.07D0
      EPSREL = 0.0D0

      ACC = 1.D-13

      PRINT 9999, ICASE, XNU, XO, H, ICON, EPSILO

      NRKEEP = NR

      IBESS = 0
      IMAXRT = 0
```

```
 CALL MANAGE(MAXRT,XO,H,XOPERT,HPERT,EPSILO,NRPERT,
+            XOFIN,HFIN,ROOTS,FROOTS,POINTS,STEPS,
+            RINTS,ERRS,NRS,IFLRT,LFLRT)

 IF ( INFO .EQ. 0 ) THEN
   PRINT 9998
   GO TO 10
 END IF

 IF ( IBESS .EQ. 2 ) PRINT 9997

 IF ( INFO .EQ. 2 ) THEN
   PRINT 9996
   GO TO 10
 END IF

 IF ( IMAXRT .EQ. 1 ) THEN
   PRINT 9995, MAXRT
   STOP
 END IF

 PRINT 9994, XOPERT(1), XOPERT(2), HPERT(1), HPERT(2)

 IF ( ICON .NE. 3 ) THEN
   PRINT 9993, NRPERT
 ELSE
   PRINT 9993, LFLRT
 END IF

 IF ( NRPERT .EQ. 0 ) GO TO 10
 IF ( ICON .EQ. 1 ) GO TO 10

 IF ( INFO .EQ. 3 ) THEN
   PRINT 9992
   GO TO 10
 ENDIF

 IF ( ICON .EQ. 4 ) THEN
   PRINT 9991, NRKEEP, NR
 ELSE
   PRINT 9990, LFLRT
 END IF

 PRINT 9989
 DO J = 1,LFLRT
   PRINT 9988, J, XOFIN(1,J), XOFIN(2,J), HFIN(1,J), HFIN(2,J)
```

```
      END

      IF ( ICON .EQ. 2 ) GO TO 10

      IF ( INFO .EQ. 4 ) THEN
        PRINT 9987
        GO TO 10
      ENDIF

      PRINT 9986
      DO J = 1,LFLRT
        IF ( IFLRT(J) .EQ. 1 ) THEN
          PRINT 9985, J, ROOTS(1,J), ROOTS(2,J),
     +                  FROOTS(1,J), FROOTS(2,J)
        ELSE
          PRINT 9984, J
        END IF
      END

   10 PRINT 9983, INFO
      STOP

 9999 FORMAT (/2X, ' STARTING VALUES :' /3X, 17('-'),
     +       //2X, ' ICASE  :    ', I1,
     +        /2X, ' ORDER  :   ', F22.15,
     +       //2X, ' XO     :   ', F22.15, F23.15,
     +        /2X, ' H      :   ', F22.15, F23.15,
     +       //2X, ' ICON   :    ', I1,
     +        /2X, ' EPSILO :   ', F22.15,
     +       //3X, 67('-'),
     +       //2X, ' RESULTS :' /3X, 9('-'))

 9998 FORMAT (/2X, '  * * * IMPROPER INPUT PARAMETERS * * *'//)

 9997 FORMAT (/2X, ' * THE PROCEDURE FOR THE CALCULATION OF THE',
     +             ' BESSEL',
     +        /2X, '   FUNCTION FAILED. THE RESULTS MAY BE',
     +             ' INACCURATE *')

 9996 FORMAT (/2X, '  * * * THE PROCEDURE FOR THE CALCULATION OF',
     +        /2X, '        THE TOTAL NUMBER OF ZEROS FAILED * * *'//)

 9995 FORMAT (/2X, ' * * * THE NUMBER OF ZEROS EXCEEDS MAXRT = ',I5,
     +        /2X, '       INCREASE THE VALUE OF MAXRT * * *'//)

 9994 FORMAT (/2X, ' THE FOLLOWING BOX WAS CONSIDERED:',
```

```
     +       //5X, 'XO = ', F22.15, F24.15,
     +        /5X, 'H  = ', F22.15, F24.15 )

 9993 FORMAT (/2X, ' THE TOTAL NUMBER OF ZEROS WITHIN',
     +             ' THIS BOX IS : ',I5)

 9992 FORMAT (/2X, ' * * * THE PROCEDURE FOR THE ISOLATION OF',
     +        /2X, '       THE ZEROS FAILED * * *'//)

 9991 FORMAT (/2X, ' NUMBER OF ZEROS REQUESTED  : ',I5,/
     +        /2X, ' NUMBER OF ZEROS ISOLATED   : ',I5)

 9990 FORMAT (/2X, ' NUMBER OF ZEROS ISOLATED   : ',I5)

 9989 FORMAT (/2X, ' BOXES CONTAINING A SINGLE ZERO :',
     +        /3X, 32('-'))

 9988 FORMAT ( /2X, I4,')  XO = ', F22.15, F24.15,
     +              /9X, 'H  = ', F22.15, F24.15 )

 9987 FORMAT (/2X, ' * * * THE PROCEDURE FOR THE COMPUTATION OF',
     +        /2X, '       THE ZEROS FAILED * * *'//)

 9986 FORMAT (/2X, ' FINAL APPROXIMATE ZEROS AND',
     +             ' VERIFICATION :',
     +        /3X, 42('-') )

 9985 FORMAT ( /2X, I4,')  Z    = (', F22.15, ',', F22.15, ' )',
     +              /9X, 'F(Z) = (', F22.15, ',', F22.15, ' )' )

 9984 FORMAT ( /2X, I4,')'/)

 9983 FORMAT (/2X, ' EXIT PARAMETER :   INFO = ',I2)

      END
```

ZEBEC outputs the following:

```
  STARTING VALUES :
  -----------------

  ICASE  :    3
  ORDER  :      -15.300000000000001

  XO     :      -22.000000000000002       .500000000000000
  H      :       45.000000000000000    100.000000000000000
```

```
ICON   :    4
EPSILO :         .000000000000100

RESULTS :
---------

THE FOLLOWING BOX WAS CONSIDERED:

  XO =    -22.000000163912774        .499999806284904
  H  =     45.000000357627865     100.000000417232520

THE TOTAL NUMBER OF ZEROS WITHIN THIS BOX IS :    16

NUMBER OF ZEROS REQUESTED  :     5

NUMBER OF ZEROS ISOLATED   :     5

BOXES CONTAINING A SINGLE ZERO :
--------------------------------

  1)  XO =     11.750000104308129        .499999806284904
      H  =     11.250000089406966      12.500000052154063

  2)  XO =       .500000014901161        .499999806284904
      H  =     11.250000089406966       6.250000026077032

  3)  XO =      6.125000059604645       6.749999832361937
      H  =      2.812500022351742       1.562500006519258

  4)  XO =      6.125000059604645       8.312499838881195
      H  =      2.812500022351742       1.562500006519258

  5)  XO =       .500000014901161       9.874999845400453
      H  =      5.625000044703484       3.125000013038516

FINAL APPROXIMATE ZEROS AND VERIFICATION :
------------------------------------------

  1)  Z    = (    12.507257919321071,     4.095557539693683 )
      F(Z) = (     -.000000000000003,     -.000000000000039 )

  2)  Z    = (    10.378711252301940,     6.178243183678395 )
      F(Z) = (     -.000000000000009,      .000000000000031 )

  3)  Z    = (     8.447945724224951,     7.613850557712050 )
```

```
       F(Z) = (      .000000000000075,      .000000000000033 )

   4)  Z    = (     6.607246778783210,     8.648294108469159 )
       F(Z) = (      .000000000000034,     -.000000000000027 )

   5)  Z    = (     1.305877373221013,    10.127220235489998 )
       F(Z) = (      .000000000000057,     -.000000000000045 )

 EXIT PARAMETER :   INFO =  1
```

More numerical examples can be found in [**185**].

6. Concluding remarks

The package ZEBEC has been applied to Bessel functions of various orders and random boxes. We have found that it behaves predictably and accurately. It calculates with certainty the total number of zeros that lie inside a given box, isolates each one of them, and then computes all these zeros or a requested number of them.

The user will appreciate the flexibility offered by the input parameter `ICON`. If nothing is known about the zeros that lie inside the given box, one may call ZEBEC with `ICON = 1` to obtain the total number of zeros. Then one may proceed with `ICON = 3` to isolate and compute all these zeros, or, if less than the total number of zeros are required, with `ICON = 4` and `NR` equal to the requested number of zeros. If only a set of boxes is required, each of which contains exactly one zero, then one may set `ICON = 2`.

Our package can be applied to any analytic function that has only simple zeros, provided that a Fortran 77 routine exists to evaluate this function and its first derivative. In fact, in Chapter 6 we will consider the functions $J_n(z) \pm iJ_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ in case $n \in \mathbb{N}$. The zeros of these functions play an important role in certain physical applications. We will prove that all the zeros that lie in $\mathbb{C}_0$ are simple. Thus ZEBEC is the package of choice to calculate approximations for these zeros.

Acknowledgements

The larger part of ZEBEC was written during two research stays (June 2–27, 1997 and March 22–April 18, 1998) at the Department of Mathematics, University of Patras (Patras, Greece). I would like to thank Michael Vrahatis, Omiros Ragos and Filareti Zafiropoulos for their very kind hospitality and for giving me the opportunity to collaborate with them. I would also like to thank Vlaamse Leergangen Leuven for its generous financial support.

CHAPTER 6

On the zeros of $J_n(z) \pm iJ_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$

The zeros of $J_n(z) \pm iJ_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$, where $n \in \mathbb{N}$, play an important role in certain physical applications. At the origin these functions have a zero of multiplicity n (if $n \geq 1$) and $2n+2$, respectively. We prove that all the zeros that lie in $\mathbb{C}_0$ are simple. As ZEBEC, the software package for computing zeros or turning points of Bessel functions that we presented in the previous chapter, can be easily extended to calculate zeros of any analytic function, provided that its zeros are known to be simple, it is the package of choice to calculate the zeros of $J_n(z) \pm iJ_{n+1}(z)$ or $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$. We tabulate the first 30 zeros of $J_5(z) - iJ_6(z)$ that lie in the fourth quadrant as computed by ZEBEC.

This chapter corresponds to our paper [**193**].

1. Introduction

In this chapter we focus on the complex zeros of the entire functions

$$z \mapsto J_n(z) \pm iJ_{n+1}(z) \tag{54}$$

and

$$z \mapsto [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z) \tag{55}$$

where $n \in \mathbb{N}$ and $J_n(z)$ denotes the Bessel function of the first kind of order n,

$$J_n(z) = \left(\frac{z}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\,(n+k)!} \left(\frac{z}{2}\right)^{2k}, \qquad |z| < \infty. \tag{56}$$

Let $F_n(z) := J_n(z) - iJ_{n+1}(z)$ and $G_n(z) := [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$.

The zeros of these functions play an important role in physical applications. The zeros of $F_0(z)$ are of interest to the specialist in the problem of water wave runup on a sloping beach, cf. Synolakis [**263, 264**]. The equation $F_n(z) = 0$ arises in problems of wave reflection from composite beaches, i.e., beaches with multiple slopes [**266**]. MacDonald [**205**] used the zeros of $G_0(z)$ to plot representative streamlines for the steady motion of a viscous fluid in a long tube, of constant radius, which rotates about its axis (the $\hat{z}$ axis) with an angular velocity that changes discontinuously at $\hat{z} = 0$ from one constant value to another of the same sign.

MacDonald derived asymptotic formulae for the zeros of $F_0(z)$, $F_n(z)$ and $G_n(z)$ in [**204**], [**205**] and [**206**], resp. See also Rawlins [**244**]. These formulae permit to locate zeros of large modulus. MacDonald observed that the accuracy of the asymptotic formulae for the zeros of $F_n(z)$ deteriorates as n increases. Also, asymptotic

formulae may be inadequate for smaller zeros. These have to be computed by some numerical procedure. To obtain the smaller zeros of $F_n(z)$, MacDonald [**206**] truncated the ascending series of $F_n(z)$ and calculated the zeros of the truncated series via Newton's method.

We suggest to use the package ZEBEC, which we presented in the previous chapter, to compute zeros of (54) or (55). Given a rectangular region in the complex plane, ZEBEC is able to compute all the zeros of the Bessel functions $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$, $H_\nu^{(2)}(z)$ or their first derivatives, where $z \in \mathbb{C} \setminus (-\infty, 0]$ and $\nu \in \mathbb{R}$, that lie inside this region. The package can be easily extended to calculate zeros of any analytic function, provided that the zeros are known to be simple. We will show that all the zeros of (54) and (55) that lie in $\mathbb{C}_0$ are simple. Hence ZEBEC is the package of choice to calculate these zeros.

This chapter is organized as follows. In Section 2 and 3 we summarize relevant properties of the zeros of (54) and (55), respectively. At $z = 0$ these functions have a zero of multiplicity n (if $n \geq 1$) and $2n + 2$, respectively. We prove that all the zeros that lie in $\mathbb{C}_0$ are simple. In Section 4 we examine how ZEBEC can be used to calculate zeros of (54) or (55). We tabulate the first 30 zeros of $J_5(z) - iJ_6(z)$ that lie in the fourth quadrant as computed by ZEBEC.

2. The zeros of $J_n(z) \pm iJ_{n+1}(z)$

If $n \geq 1$, then $J_n(z)$ has a zero of multiplicity n at the origin, whereas $J_{n+1}(z)$ has a zero of multiplicity $n+1$ at the origin. Therefore, if $n \geq 1$, then the functions $J_n(z) \pm iJ_{n+1}(z)$ have a zero of multiplicity n at the origin. However, they don't have other real zeros.

THEOREM 44. *Except for $z = 0$, the functions $J_n(z) \pm iJ_{n+1}(z)$ have no zeros on the real axis.*

PROOF. Suppose that $z^\star \in \mathbb{R}_0$ is such that

$$J_n(z^\star) - iJ_{n+1}(z^\star) = 0. \tag{57}$$

The Taylor series (56) implies that

$$\overline{J_n(z)} = J_n(\overline{z}) \tag{58}$$

and thus $J_n(z)$ and $J_{n+1}(z)$ take real values on the real axis. Therefore (57) cannot hold unless $J_n(z^\star) = J_{n+1}(z^\star) = 0$. However, the positive zeros of $J_n(z)$ are interlaced with those of $J_{n+1}(z)$, cf. Watson [**295**, §15.22 on page 479]. Thus, no positive zero of $J_n(z)$ coincides with a positive zero of $J_{n+1}(z)$. Also, (56) implies that

$$J_n(-z) = (-1)^n J_n(z). \tag{59}$$

Therefore, the zeros of $J_n(z)$ and $J_{n+1}(z)$ are symmetric about the origin. Thus, no negative zero of $J_n(z)$ coincides with a negative zero of $J_{n+1}(z)$ and we may conclude that $J_n(z^\star) = J_{n+1}(z^\star) = 0$ cannot hold if $z^\star \in \mathbb{R}_0$. This shows that $J_n(z) - iJ_{n+1}(z)$ has no zeros in $\mathbb{R}_0$. The proof for $J_n(z) + iJ_{n+1}(z)$ is analogous. □

Except for $z = 0$, the functions $J_n(z) \pm iJ_{n+1}(z)$ have no zeros on the imaginary axis. This is a corollary of the following theorem.

THEOREM 45. *The following integral representations hold:*

$$J_n(z) + iJ_{n+1}(z) = \frac{1}{\sqrt{\pi}\,\Gamma(n+\frac{1}{2})}\left(\frac{z}{2}\right)^n \int_{-1}^{1} e^{izt}(1+t)^{n+\frac{1}{2}}(1-t)^{n-\frac{1}{2}}\,dt$$

and

$$J_n(z) - iJ_{n+1}(z) = \frac{1}{\sqrt{\pi}\,\Gamma(n+\frac{1}{2})}\left(\frac{z}{2}\right)^n \int_{-1}^{1} e^{izt}(1+t)^{n-\frac{1}{2}}(1-t)^{n+\frac{1}{2}}\,dt.$$

PROOF. The Poisson integral representation for Bessel functions of the first kind (see for example [**77**, formula (7) on page 81]) tells us that

$$\Gamma(n+\frac{1}{2})J_n(z) = \frac{1}{\sqrt{\pi}}\left(\frac{z}{2}\right)^n \int_{-1}^{1} e^{izt}(1-t^2)^{n-\frac{1}{2}}\,dt. \tag{60}$$

By replacing n by $n+1$, we obtain that

$$\Gamma(n+\frac{3}{2})J_{n+1}(z) = \frac{1}{\sqrt{\pi}}\left(\frac{z}{2}\right)^{n+1} \int_{-1}^{1} e^{izt}(1-t^2)^{n+\frac{1}{2}}\,dt.$$

We divide this equation by $n+\frac{1}{2}$ and integrate the integral in the right-hand side by parts. This gives

$$\Gamma(n+\frac{1}{2})J_{n+1}(z) = \frac{1}{\sqrt{\pi}}\left(\frac{z}{2}\right)^{n+1} \int_{-1}^{1} \frac{e^{izt}}{iz}(1-t^2)^{n-\frac{1}{2}}2t\,dt. \tag{61}$$

By combining (60) and (61) one obtains the given integral representations for $J_n(z)+iJ_{n+1}(z)$ and $J_n(z)-iJ_{n+1}(z)$. □

NOTE. The functions $J_n(z) \pm iJ_{n+1}(z)$ also play a role in asymptotics of orthogonal polynomials on the unit circle [**17, 285**]. The integral representations given in the previous theorem can also be found in [**285**].

COROLLARY 46. *Except for $z = 0$, the functions $J_n(z) \pm iJ_{n+1}(z)$ have no zeros on the imaginary axis.*

PROOF. This follows immediately from the fact that the integrands (and thus also the integrals) in the representations given in the previous theorem are positive if z lies on the imaginary axis. □

Equation (58) implies that the zeros of $J_n(z) + iJ_{n+1}(z)$ are the reflections of the zeros of $J_n(z) - iJ_{n+1}(z)$ about the real axis. Also, (58) and (59) imply that the zeros of $J_n(z) \pm iJ_{n+1}(z)$ are symmetric with respect to the imaginary axis. Therefore, as far as the zeros are concerned, we may restrict our attention to only one of the functions $J_n(z) \pm iJ_{n+1}(z)$, for example $J_n(z) - iJ_{n+1}(z)$, and to the right half plane. The following theorem implies that it is even sufficient to consider $J_n(z) - iJ_{n+1}(z)$ only in the fourth quadrant.

THEOREM 47. *All the zeros of $J_n(z) + iJ_{n+1}(z)$ lie in the upper half plane whereas all the zeros of $J_n(z) - iJ_{n+1}(z)$ lie in the lower half plane.*

PROOF. See Tadepalli and Synolakis [**266**]. □

We already know that $J_n(z) \pm iJ_{n+1}(z)$ has a zero of multiplicity n at the origin if $n \geq 1$. The following theorem tells us that all the other zeros are simple.

THEOREM 48. *The zeros of $J_n(z) \pm iJ_{n+1}(z)$ that lie in $\mathbb{C}_0$ are simple.*

PROOF. Suppose that $z^\star \in \mathbb{C}_0$ is a multiple zero of $F_n(z) = J_n(z) - iJ_{n+1}(z)$. Then

$$F_n(z^\star) = J_n(z^\star) - iJ_{n+1}(z^\star) = 0 \tag{62}$$

and

$$F_n'(z^\star) = J_n'(z^\star) - iJ_{n+1}'(z^\star) = 0.$$

By using the relations [**77**, §7.2.8]

$$\begin{aligned} J_n'(z) &= \frac{n}{z}J_n(z) - J_{n+1}(z) \\ J_{n+1}'(z) &= J_n(z) - \frac{n+1}{z}J_{n+1}(z) \end{aligned}$$

we obtain that

$$F_n'(z^\star) = \big(\frac{n}{z^\star} - i\big)J_n(z^\star) - \big(1 - i\frac{n+1}{z^\star}\big)J_{n+1}(z^\star) = 0. \tag{63}$$

Equations (62) and (63) represent a homogeneous linear system of equations in $J_n(z^\star)$ and $J_{n+1}(z^\star)$. Its determinant is equal to $i\frac{2n+1}{z^\star} \neq 0$. It follows that $J_n(z^\star) = J_{n+1}(z^\star) = 0$. This implies that $z^\star$ must be real. Indeed, Bessel functions of the first kind and order > -1 have only real zeros, cf. Watson [**295**, §15.27]. However, this is impossible as we have shown in Theorem 44 that $F_n(z)$ has no zeros in $\mathbb{R}_0$. This proves that all the zeros of $F_n(z)$ that lie in $\mathbb{C}_0$ are simple. The proof for $J_n(z) + iJ_{n+1}(z)$ is analogous. □

3. The zeros of $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$

If $n \geq 1$, then $J_n(z)$ has a zero of multiplicity n at the origin. The Bessel functions $J_{n+1}(z)$ and $J_{n+2}(z)$ have a zero of multiplicity $n+1$ respectively $n+2$ at the origin. Therefore, if $n \geq 1$, then $G_n(z) = [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ has a zero of multiplicity $\geq 2n+2$ at the origin. Using the Taylor series (56) one can easily verify that the multiplicity is in fact equal to $2n+2$. This holds also in case $n = 0$. The inequality (cf. Szász [**265**])

$$[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z) > \frac{[J_{n+1}(z)]^2}{n+2}, \qquad z > 0,$$

and equation (59) imply that no other zeros exist on the real axis. In Corollary 50 we will show that no other zeros exist on the imaginary axis.

If $z^\star$ is a zero of $G_n(z)$, then (58) and (59) immediately imply that the same holds for $\overline{z^\star}$ and $-z^\star$. In other words, the zeros of $G_n(z)$ are symmetric with respect to the real axis and about the origin. Thus we may restrict our attention to one quadrant in the complex plane, for example the first quadrant.

THEOREM 49. *$G_n'(z) = \frac{2}{z} J_n(z) J_{n+2}(z)$ for all $z \neq 0$.*

PROOF. We use the recurrence relations [**77**, §7.2.8]

$$\begin{aligned} J_n(z) + J_{n+2}(z) &= \tfrac{2}{z}(n+1) J_{n+1}(z) \\ J_n(z) - J_{n+2}(z) &= 2 J_{n+1}'(z) \\ z J_{n+2}'(z) + (n+2) J_{n+2}(z) &= z J_{n+1}(z) \\ z J_n'(z) - n J_n(z) &= -z J_{n+1}(z) \end{aligned}$$

to eliminate $J_n'(z)$, $J_{n+1}(z)$, $J_{n+1}'(z)$ and $J_{n+2}'(z)$ from

$$G_n'(z) = 2 J_{n+1}(z) J_{n+1}'(z) - J_n(z) J_{n+2}'(z) - J_n'(z) J_{n+2}(z).$$

It follows that

$$\begin{aligned} G_n'(z) &= 2 \frac{z}{2(n+1)} \big(J_n(z) + J_{n+2}(z)\big) \frac{1}{2} \big(J_n(z) - J_{n+2}(z)\big) \\ &\qquad - J_n(z) \big(J_{n+1}(z) - \frac{n+2}{z} J_{n+2}(z)\big) - J_{n+2}(z) \big(\frac{n}{z} J_n(z) - J_{n+1}(z)\big) \\ &= \frac{z}{2(n+1)} \big([J_n(z)]^2 - [J_{n+2}(z)]^2\big) - J_n(z) J_{n+1}(z) + \frac{n+2}{z} J_n(z) J_{n+2}(z) \\ &\qquad - \frac{n}{z} J_n(z) J_{n+2}(z) + J_{n+1}(z) J_{n+2}(z) \\ &= \frac{z}{2(n+1)} \big([J_n(z)]^2 - [J_{n+2}(z)]^2\big) - J_n(z) \frac{z}{2(n+1)} \big(J_n(z) + J_{n+2}(z)\big) \\ &\qquad + \frac{2}{z} J_n(z) J_{n+2}(z) + \frac{z}{2(n+1)} \big(J_n(z) + J_{n+2}(z)\big) J_{n+2}(z) \\ &= \frac{2}{z} J_n(z) J_{n+2}(z). \end{aligned}$$

This proves the theorem. □

COROLLARY 50. *Except for $z = 0$, the function $[J_{n+1}(z)]^2 - J_n(z) J_{n+2}(z)$ has no zeros on the imaginary axis.*

PROOF. Define $f_n(\zeta) : \mathbb{R} \to \mathbb{R} : \zeta \mapsto G_n(i\zeta)$. Then

$$f_n'(\zeta) = i G_n'(i\zeta) = \frac{2}{\zeta} J_n(i\zeta) J_{n+2}(i\zeta).$$

Let $I_n(z)$ denote the modified Bessel function of the first kind of order n. From the Taylor series

$$I_n(z) = \left(\frac{z}{2}\right)^n \sum_{k=0}^{\infty} \frac{1}{k!\,(n+k)!} \left(\frac{z}{2}\right)^{2k}, \qquad |z| < \infty,$$

it immediately follows that $J_n(z) = i^n I_n(-iz)$. Thus $J_n(i\zeta) = i^n I_n(\zeta)$ and

$$f_n'(\zeta) = (-1)^{n+1} \frac{2}{\zeta} I_n(\zeta) I_{n+2}(\zeta).$$

Now we have to distinguish between $\zeta > 0$ and $\zeta < 0$.

- If $\zeta > 0$, then $I_n(\zeta) > 0$ and thus $\operatorname{sign} f_n'(\zeta) = (-1)^{n+1}$ independently of ζ. Thus $f_n(\zeta)$ is either strictly increasing or strictly decreasing in $(0, +\infty)$. As $f_n(0) = 0$, it follows that $f_n(\zeta)$ has no zeros in $(0, +\infty)$.

- If $\zeta < 0$ then, as $I_n(-\zeta) = (-1)^n I_n(\zeta)$, it follows that $\operatorname{sign} I_n(\zeta) = (-1)^n$ and thus $\operatorname{sign} f_n'(\zeta) = (-1)^{n+1}(-1)(-1)^n(-1)^{n+2} = (-1)^{3n+4}$, which is again independent of ζ. Thus $f_n(\zeta)$ has no zeros in $(-\infty, 0)$.

This proves the theorem. Note that $f_n'(\zeta)$ has opposite sign in $(-\infty, 0)$ and $(0, +\infty)$. This is in accordance with the fact that $f_n(\zeta)$ has a multiple zero of even multiplicity at $\zeta = 0$. □

COROLLARY 51. *The zeros of* $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ *that lie in* $\mathbb{C}_0$ *are simple.*

PROOF. Suppose that $z^\star \in \mathbb{C}_0$ is a multiple zero of $G_n(z)$,

$$G_n(z^\star) = G_n'(z^\star) = 0.$$

Then Theorem 49 implies that $J_n(z^\star) = 0$ or $J_{n+2}(z^\star) = 0$. However, this cannot hold as we have proven that $G_n(z)$ has no real zeros except at $z = 0$ whereas Bessel functions of the first kind and order > -1 have only real zeros, cf. Watson [**295**, §15.27].

Another argument goes as follows. If $G_n'(z^\star) = 0$, then we have just seen that $z^\star$ must be real. Also, $G_n(z^\star) = 0$ together with $J_n(z^\star) = 0$ or $J_{n+2}(z^\star) = 0$ implies that $J_{n+1}(z^\star) = 0$. This is impossible because of the interlacing property of the zeros of Bessel functions of the first kind and of successive nonnegative integer orders, cf. Watson [**295**, §15.22]. □

4. Numerical results

The package ZEBEC was written for the Bessel functions $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$, $H_\nu^{(2)}(z)$ and their first derivatives, but it can be easily extended to calculate zeros of any analytic function $f(z)$ whose zeros are known to be simple, provided that a Fortran 77 routine exists to evaluate the logarithmic derivative $f'(z)/f(z)$. In Section 2 and 3 we have shown that all the zeros of $F_n(z) = J_n(z) - iJ_{n+1}(z)$ and $G_n(z) = [J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$ that lie in $\mathbb{C}_0$ are simple. The zeros of these functions can therefore be calculated via ZEBEC. As

$$\frac{F_n'(z)}{F_n(z)} = \frac{J_n'(z) - iJ_{n+1}'(z)}{J_n(z) - iJ_{n+1}(z)} \qquad \text{and} \qquad \frac{G_n'(z)}{G_n(z)} = \frac{\frac{2}{z}J_n(z)J_{n+2}(z)}{[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)},$$

it seems that one has to evaluate $J_n(z)$, $J_n'(z)$, $J_{n+1}(z)$, $J_{n+1}'(z)$ and $J_n(z)$, $J_{n+1}(z)$, $J_{n+2}(z)$ to compute $F_n'(z)/F_n(z)$ and $G_n'(z)/G_n(z)$, resp. However, the recurrence relations [**77**, §7.2.8]

$$J_n'(z) = \frac{n}{z}J_n(z) - J_{n+1}(z)$$
$$J_{n+1}'(z) = J_n(z) - \frac{n+1}{z}J_{n+1}(z)$$

and

$$J_{n+2}(z) = \frac{2(n+1)}{z}J_{n+1}(z) - J_n(z),$$

respectively, imply that it is sufficient to compute $J_n(z)$ and $J_{n+1}(z)$.

We have calculated the first 30 zeros of $J_5(z) - iJ_6(z)$ that lie in the fourth quadrant. The results are shown in Table 1. The reader may compare these with the values given by MacDonald [**206**, Table 7 on page 633] and conclude that our approximations are more accurate, in particular the approximations for the smaller zeros.

9.33311962903697 - i* 0.69444271743387
12.94510369182598 - i* 0.82936671338824
16.33461779337308 - i* 0.93080334205489
19.63429282581447 - i* 1.01367717572733
22.88679602216657 - i* 1.08418924023245
26.11097880771598 - i* 1.14572183934914
29.31666723089333 - i* 1.20038033303009
32.50954914896803 - i* 1.24958398094783
35.69316831642123 - i* 1.29434274725774
38.86985731122473 - i* 1.33540436964299
42.04121867038544 - i* 1.37333958320706
45.20839196877325 - i* 1.40859475982058
48.37221090307405 - i* 1.44152605770887
51.53330011302131 - i* 1.47242245993473
54.69213722132923 - i* 1.50152182265512
57.84909391098780 - i* 1.52902235758964
61.00446389725773 - i* 1.55509103690239
64.15848244476649 - i* 1.57986986761685
67.31134027894330 - i* 1.60348065661451
70.46319369060231 - i* 1.62602868463876
73.61417199987465 - i* 1.64760557747193
76.76438315327531 - i* 1.66829157701086
79.91391797841353 - i* 1.68815735731665
83.06285345843724 - i* 1.70726549130409
86.21125528062697 - i* 1.72567164594395
89.35917984058424 - i* 1.74342556430824
92.50667583333037 - i* 1.76057187870224
95.65378552754606 - i* 1.77715078859128
98.80054579435346 - i* 1.79319862955800
101.94698894421000 - i* 1.80874835363552

TABLE 1. Approximations for the first 30 zeros of $J_5(z) - iJ_6(z)$ that lie in the fourth quadrant.

Acknowledgements

The results in this chapter were obtained in collaboration with Pierre Verlinden.

I would like to thank Michael Vrahatis for providing me with references [**204**], [**206**] and [**266**], which initiated this research. I would also like to thank Mourad Ismail, Duncan MacDonald and Costas Synolakis for interesting discussions.

Part 3

CHAPTER 7

Newton's method for multiple zeros

We propose a modification of Newton's method for computing multiple zeros of analytic mappings (in other words, multiple roots of systems of analytic equations). Under mild assumptions the iteration converges quadratically. It involves certain constants whose product is a lower bound for the multiplicity of the zero. As these constants are usually not known in advance, we devise an iteration in which not only an approximation for the zero is refined, but also approximations for these constants. Numerical examples illustrate the effectiveness of our approach.

This chapter corresponds to our paper [**184**].

1. Introduction

Consider a smooth function $f : \mathbb{C} \to \mathbb{C}$ that has a zero of multiplicity μ at the point $z^\star$. If $\mu = 1$, then Newton's method converges quadratically to $z^\star$ if the initial iterate is sufficiently close to $z^\star$. If $\mu > 1$, then the convergence is only linear. In the latter case, if μ is known in advance, quadratic convergence can be regained by considering the iteration

$$z^{(p+1)} = z^{(p)} - \mu \frac{f(z^{(p)})}{f'(z^{(p)})}, \qquad p = 0, 1, 2, \ldots . \tag{64}$$

Van de Vel [**281, 283**] devised an iteration in which not only an approximation for the zero is refined, but also an estimate of its multiplicity. King [**177**] analysed Van de Vel's method and proved that its order of convergence is 1.554. He rearranged the order of the calculations and obtained an iteration that has order of convergence 1.618. This iteration proceeds as follows:

$$\left\{ \begin{array}{lcl} \mu^{(p+1)} & = & \dfrac{u(z^{(p)})}{u(z^{(p)}) - u(z^{(p+1)})} \mu^{(p)} \\ z^{(p+2)} & = & z^{(p+1)} - \mu^{(p+1)} u(z^{(p+1)}) \end{array} \right. \tag{65}$$

for $p = 0, 1, 2, \ldots$, with initial $z^{(0)}$ and $\mu^{(0)}$, and after one preliminary quasi-Newton step $z^{(1)} = z^{(0)} - \mu^{(0)} u(z^{(0)})$, and where $u(z) := f(z)/f'(z)$.

We generalize these two results to the multivariate case. We consider systems of analytic equations, i.e., analytic mappings $f : \mathbb{C}^n \to \mathbb{C}^n$. The multidimensional version of (64) is formulated in Theorem 56. Instead of μ the iteration now involves a diagonal matrix containing certain constants $k_1, \ldots, k_n$ that are called orders. The product of these orders is a lower bound for μ (Theorem 55). In Section 4 we present an algorithm in which an approximation for the zero as well as approximations for the

orders are refined iteratively. This iteration is our multidimensional generalization of (65). A lot of numerical examples illustrate our results.

At a multiple zero the Jacobian matrix of f is singular. The set of points $z \in \mathbb{C}^n$ such that $\det f'(z) = 0$ is a codimension one smooth manifold through the zero. As soon as an iterate lies on this manifold, the iteration breaks down. We assume throughout this chapter that this does not happen. In other words, we assume that the initial iterate is such that the iteration is well defined at every step. This assumption enables us to focus entirely on the order of convergence.

The behaviour of Newton's method in case the Jacobian is singular at the zero has been analysed extensively in the literature [**65, 66, 67, 68, 69, 119, 120, 121, 122, 173, 174, 223, 245, 246, 309**]. Many sufficient conditions for its convergence have been formulated. Under certain regularity and smoothness assumptions, the existence of special regions (cones, starlike regions) about the zero $z^\star$ has been proven. The Jacobian is regular in every point of these regions except in $z^\star$. If the initial iterate lies in such a region, then the Newton iterates will remain in this region and converge (linearly) to $z^\star$. We have not investigated the existence of such regions for the iterations presented in this chapter. Indeed, the proofs for the classical cases are extremely complex and it is not a priori clear how to extend them to our iterations.

In [**68**] a modification of Newton's method is proposed that produces a sequence $\{z^{(p)}\}_{p\geq 0}$ such that the subsequence $\{z^{(2p)}\}_{p\geq 0}$ converges quadratically to the zero. However, this method works only in case the dimension of the null space of the Jacobian at the zero is equal to 1 or 2, and the projector onto this null space is known explicitely. Other modifications of Newton's method have been proposed in [**65, 174**]. These methods result in superlinear or quadratic convergence but again require rather restrictive hypotheses to be satisfied and need additional information (certain constants, projectors, ...) that is usually not available.

In [**225, 226, 227**] a 'deflation algorithm' was proposed for computing multiple roots of systems of nonlinear algebraic equations. The system to be solved is replaced by another one having the same root but with a lower multiplicity. While the deflation algorithm proceeds, the multiplicity is systematically reduced until it is equal to one and classical methods can be applied. However, this algorithm requires symbolic calculation and works only for systems of algebraic equations.

Other approaches that have been proposed include bordering methods [**146, 147, 148, 195, 210**], enlargement methods [**224, 270, 296**] and homotopy continuation methods [**215, 216, 217**].

All these methods require deciding whether the problem is singular: one should know in advance that the zero is multiple. This probably makes these methods unsuitable for general purpose use. As we will illustrate in Example 27, our method also works in case the zero is simple. Moreover, it requires no additional information, works under mild assumptions, and provides a lower bound for the multiplicity of the zero.

2. Preliminaries and notation

Let $f = f(z) : \mathbb{C}^n \to \mathbb{C}^n$ be an analytic mapping, with $z = (z_1, \ldots, z_n)$ and $f = (f_1, \ldots, f_n)$. A point $z^\star \in \mathbb{C}^n$ is called a *zero* of f if $f(z^\star) = 0$. An isolated zero $z^\star$ of f is called *simple* if the Jacobian matrix of f at $z^\star$ is regular, $\det f'(z^\star) \neq 0$.

The following material is taken from the well-known book by Aĭzenberg and Yuzhakov [**3**].

PROPOSITION 52. *If the closure of a neighbourhood $U_{z^\star}$ of a zero $z^\star$ of f does not contain other zeros of f, then there exists an $\epsilon > 0$ such that for almost all $\zeta \in \mathbb{C}^n$, $\|\zeta\|_2 < \epsilon$, the mapping*

$$z \mapsto f(z) - \zeta \tag{66}$$

has only simple zeros in $U_{z^\star}$ and their number depends neither on ζ nor on the choice of the neighbourhood $U_{z^\star}$.

The number of zeros of the mapping (66) in $U_{z^\star}$ indicated in this proposition is called the *multiplicity* of the zero $z^\star$ of f and is denoted by $\mu_{z^\star}(f)$. In other words, the multiplicity of an isolated zero of an analytic mapping is given by the number of simple zeros into which this zero desintegrates under a sufficiently small perturbation of the mapping.

The next result follows from the local invertibility of an analytic mapping at points where its Jacobian matrix is regular.

PROPOSITION 53. *The multiplicity of a simple zero is equal to* 1.

PROPOSITION 54. *If $z^\star$ is an isolated zero of f and* $\det f'(z^\star) = 0$, *then its multiplicity $\mu_{z^\star}(f)$ is larger than* 1.

This statement justifies calling an isolated zero $z^\star$ of f *multiple* if $\det f'(z^\star) = 0$.

Now let $z^\star = (z_1^\star, \ldots, z_n^\star)$ be an isolated zero of $f = (f_1, \ldots, f_n)$ such that

$$f_j(z) = \sum_{|\alpha| \geq k_j} c_{j,\alpha}(z - z^\star)^\alpha$$

for $j = 1, \ldots, n$ where α is a multi-index, $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, $|\alpha| = \alpha_1 + \cdots + \alpha_n$ and $(z - z^\star)^\alpha = (z_1 - z_1^\star)^{\alpha_1} \cdots (z_n - z_n^\star)^{\alpha_n}$. We call k_j the *order* of $z^\star$ as a zero of f_j. Define

$$P_j(z) := \sum_{|\alpha| = k_j} c_{j,\alpha}(z - z^\star)^\alpha$$

for $j = 1, \ldots, n$. The homogeneous polynomial mapping

$$P = P(z) := (P_1(z), \ldots, P_n(z)) \tag{67}$$

is called the *homogeneous principal part* of f at $z^\star$. The following theorem by Tsikh and Yuzhakov relates the multiplicity $\mu_{z^\star}(f)$ to the orders $k_1, \ldots, k_n$.

THEOREM 55. *The multiplicity of an isolated zero $z^\star$ of f is equal to the product of the orders of $z^\star$ as a zero of $f_1, \ldots, f_n$ if and only if $z^\star$ is an isolated zero of the mapping* (67). *Moreover, the inequality $\mu_{z^\star}(f) \geq k_1 \cdots k_n$ always holds.*

3. A modification of Newton's method

THEOREM 56. *Suppose that $z^{(0)}$ is such that the iteration*

$$z^{(p+1)} = z^{(p)} - [f'(z^{(p)})]^{-1}\,\mathrm{diag}(k_1,\dots,k_n)f(z^{(p)}), \qquad p = 0,1,2,\dots,$$

is well defined for every p. If $\det P'(z) \not\equiv 0$ and if $z^{(0)}$ is sufficiently close to $z^\star$, then $z^{(p)}$ converges quadratically to $z^\star$. If $\det P'(z) \equiv 0$, then the convergence is only linear.

PROOF. Define

$$\hat f_j(z) := \sum_{|\alpha|\ge k_j+1} c_{j,\alpha}(z-z^\star)^\alpha$$

for $j = 1,\dots,n$. As

$$f_j(z) = P_j(z) + \hat f_j(z) = \sum_{|\alpha|=k_j} c_{j,\alpha}(z-z^\star)^\alpha + \hat f_j(z) \tag{68}$$

for $j = 1,\dots,n$, it follows that

$$\frac{\partial f_j}{\partial z_k}(z) = \sum_{|\alpha|=k_j} \alpha_k c_{j,\alpha}(z_1-z_1^\star)^{\alpha_1}\cdots(z_k-z_k^\star)^{\alpha_k-1}\cdots(z_n-z_n^\star)^{\alpha_n} + \frac{\partial \hat f_j}{\partial z_k}(z)$$

for $j,k = 1,\dots,n$. Let $e^{(p)} = (e_{1,p},\dots,e_{n,p}) := z^{(p)} - z^\star$. Then the iteration can be written as

$$f'(z^\star + e^{(p)})e^{(p+1)} = f'(z^\star + e^{(p)})e^{(p)} - \mathrm{diag}(k_1,\dots,k_n)f(z^\star + e^{(p)}). \tag{69}$$

The jth component of the vector appearing in the right-hand side of (69) is given by

$$\begin{aligned}
g_j(e^{(p)}) &:= \sum_{k=1}^n \frac{\partial f_j}{\partial z_k}(z^\star + e^{(p)})e_{k,p} - k_j f_j(z^\star + e^{(p)}) \\
&= \sum_{k=1}^n \Big[\sum_{|\alpha|=k_j} \alpha_k c_{j,\alpha} e_{1,p}^{\alpha_1}\cdots e_{n,p}^{\alpha_n} + \frac{\partial \hat f_j}{\partial z_k}(z^\star + e^{(p)})e_{k,p}\Big] \\
&\qquad - k_j\Big[\sum_{|\alpha|=k_j} c_{j,\alpha} e_{1,p}^{\alpha_1}\cdots e_{n,p}^{\alpha_n} + \hat f_j(z^\star + e^{(p)})\Big] \\
&= \sum_{|\alpha|=k_j} \big(|\alpha| - k_j\big)c_{j,\alpha}\big[e^{(p)}\big]^\alpha + \sum_{k=1}^n \frac{\partial \hat f_j}{\partial z_k}(z^\star + e^{(p)})e_{k,p} - k_j \hat f_j(z^\star + e^{(p)}) \\
&= \sum_{k=1}^n \sum_{|\alpha|\ge k_j+1} \alpha_k c_{j,\alpha}\big[e^{(p)}\big]^\alpha - k_j \sum_{|\alpha|\ge k_j+1} c_{j,\alpha}\big[e^{(p)}\big]^\alpha \\
&= \sum_{|\alpha|\ge k_j+1} \big(|\alpha| - k_j\big)c_{j,\alpha}\big[e^{(p)}\big]^\alpha \\
&= \sum_{|\alpha|=k_j+1} c_{j,\alpha}\big[e^{(p)}\big]^\alpha + \sum_{|\alpha|>k_j+1} \big(|\alpha| - k_j\big)c_{j,\alpha}\big[e^{(p)}\big]^\alpha.
\end{aligned}$$

It follows that $|g_j(e^{(p)})| = \mathcal{O}(\,\|e^{(p)}\|^{k_j+1}\,)$ for $j = 1,\dots,n$.

One can easily verify that $\det P'(z)$ is a homogeneous polynomial in $z - z^\star$. Now there are two possibilities: either all its coefficients are equal to zero, $\det P' \equiv 0$, or $\det P'$ has degree $\sum_{j=1}^n k_j - n$. By using (68) we can write $\det f'$ as a sum of 2^n determinants, including $\det P'$, and it follows readily that

$$\det f'(z^\star + e^{(p)}) = \begin{cases} \mathcal{O}(\|e^{(p)}\|^{\sum_{j=1}^n k_j - n}) & \text{if } \det P' \not\equiv 0, \\ \mathcal{O}(\|e^{(p)}\|^{\sum_{j=1}^n k_j - n + 1}) & \text{if } \det P' \equiv 0. \end{cases} \tag{70}$$

By setting $g := (g_1, \ldots, g_n)$ we can write (69) as

$$f'(z^\star + e^{(p)}) e^{(p+1)} = g(e^{(p)}).$$

Cramer's rule implies that

$$e_j^{(p+1)} = \frac{1}{\det f'(z^\star + e^{(p)})} \times \tag{71}$$

$$\begin{vmatrix} \frac{\partial f_1}{\partial z_1}(z^\star + e^{(p)}) & \ldots & \frac{\partial f_1}{\partial z_{j-1}}(z^\star + e^{(p)}) & g_1(e^{(p)}) & \frac{\partial f_1}{\partial z_{j+1}}(z^\star + e^{(p)}) & \ldots & \frac{\partial f_1}{\partial z_n}(z^\star + e^{(p)}) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial z_1}(z^\star + e^{(p)}) & \ldots & \frac{\partial f_n}{\partial z_{j-1}}(z^\star + e^{(p)}) & g_n(e^{(p)}) & \frac{\partial f_n}{\partial z_{j+1}}(z^\star + e^{(p)}) & \ldots & \frac{\partial f_n}{\partial z_n}(z^\star + e^{(p)}) \end{vmatrix}$$

for $j = 1, \ldots, n$. The denominator in the right-hand side of (71) was examined in (70). One can easily verify that the numerator is $\mathcal{O}(\|e^{(p)}\|^\alpha)$ with $\alpha = (\sum_{j=1}^n k_j - n) - (k_j - 1) + (k_j + 1) = \sum_{j=1}^n k_j - n + 2$. This proves the theorem. □

REMARK. As already mentioned, the problem of analysing the possible structure of the set of initial iterates $z^{(0)}$ that lie in a neighbourhood of $z^\star$ and guarantee convergence of the iteration is of considerable difficulty. We therefore restrict our attention entirely to the *order* of convergence.

In the following examples we used Mathematica 2.2. All the calculations were done in multiple precision arithmetic.

EXAMPLE 21. The mapping $f = (f_1, f_2) = (z_1 \sin z_1 + z_2^3, z_2 + z_1 \sin z_2)$ has an isolated zero at $z^\star = (0, 0)$. The orders are $k_1 = 2$ and $k_2 = 1$. The homogeneous principal part of f at $z^\star$ is given by $P = P(z_1, z_2) = (z_1^2, z_2)$. It follows that $z^\star$ is an isolated zero of P and thus, according to Theorem 55, the multiplicity of $z^\star$ as a zero of f is equal to $k_1 \cdot k_2 = 2$. The Jacobian matrix of P is given by

$$P'(z_1, z_2) = \begin{bmatrix} 2z_1 & 0 \\ 0 & 1 \end{bmatrix}$$

and thus $\det P'(z_1, z_2) \not\equiv 0$. Therefore the iteration of Theorem 56 will converge quadratically. Table 1 illustrates this. The initial iterate was $z^{(0)} := (0.2, 0.2)$.

EXAMPLE 22. The mapping $f = (f_1, f_2) = (z_1 z_2 + (\sin z_1)^2 + z_2^3, \sin z_1 \sin z_2)$ has an isolated zero at $z^\star = (0, 0)$. The orders are $k_1 = 2$ and $k_2 = 2$. The homogeneous principal part of f at $z^\star$ is given by $P = P(z_1, z_2) = (z_1^2 + z_1 z_2, z_1 z_2)$. As $P(z_1, z_2) = 0$ if and only if $z_1 = 0$ and z_2 arbitrary, it follows that $z^\star$ is not an

p	$-\log_{10}\lvert z_1^{(p)}\rvert$	$-\log_{10}\lvert z_2^{(p)}\rvert$	$-\log_{10}\sqrt{\lvert z_1^{(p)}\rvert^2+\lvert z_2^{(p)}\rvert^2}$
0	0.7	0.7	0.6
1	2.0	1.5	1.5
2	2.8	3.6	2.8
3	8.4	6.4	6.4
4	11.1	14.8	11.1
5	27.1	26.0	26.0
$\vdots$	$\vdots$	$\vdots$	$\vdots$

TABLE 1. Example 21.

p	$-\log_{10}\lvert z_1^{(p)}\rvert$	$-\log_{10}\lvert z_2^{(p)}\rvert$	$-\log_{10}\sqrt{\lvert z_1^{(p)}\rvert^2+\lvert z_2^{(p)}\rvert^2}$
0	0.7	0.7	0.6
1	1.5	1.4	1.3
2	3.1	3.0	2.9
3	6.3	6.3	6.2
4	12.9	12.8	12.6
5	25.8	25.7	25.6
$\vdots$	$\vdots$	$\vdots$	$\vdots$

TABLE 2. Example 22.

isolated zero of P and thus, according to Theorem 55, the multiplicity of $z^\star$ as a zero of f is strictly larger than $k_1 \cdot k_2 = 4$. The Jacobian matrix of P is given by

$$P'(z_1, z_2) = \begin{bmatrix} 2z_1 + z_2 & z_1 \\ z_2 & z_1 \end{bmatrix}$$

and thus $\det P'(z_1, z_2) \not\equiv 0$. Therefore the iteration of Theorem 56 will converge quadratically. Table 2 illustrates this. The initial iterate was $z^{(0)} := (0.2, 0.2)$.

EXAMPLE 23. The mapping

$$f = (f_1, f_2) = (z_1 + z_2 + z_1^2 + z_1 z_2 + 2z_2^3 + (\sin z_1)^3, 2(z_1 + z_2)^3 + z_1^4)$$

has an isolated zero at $z^\star = (0, 0)$. The orders are $k_1 = 1$ and $k_2 = 3$. The homogeneous principal part of f at $z^\star$ is given by

$$P = P(z_1, z_2) = (z_1 + z_2, 2(z_1 + z_2)^3).$$

As $P(z_1, z_2) = 0$ if and only if $z_2 = -z_1$, it follows that $z^\star$ is not an isolated zero of P and thus, according to Theorem 55, the multiplicity of $z^\star$ as a zero of f is strictly larger than $k_1 \cdot k_2 = 3$. The Jacobian matrix of P is given by

$$P'(z_1, z_2) = \begin{bmatrix} 1 & 1 \\ 6(z_1 + z_2)^2 & 6(z_1 + z_2)^2 \end{bmatrix}$$

and thus $\det P'(z_1, z_2) \equiv 0$. Therefore the iteration of Theorem 56 will converge linearly. Table 3 illustrates this. The initial iterate was $z^{(0)} := (0.2, 0.2)$.

p	$-\log_{10}\lvert z_1^{(p)}\rvert$	$-\log_{10}\lvert z_2^{(p)}\rvert$	$-\log_{10}\sqrt{\lvert z_1^{(p)}\rvert^2+\lvert z_2^{(p)}\rvert^2}$
0	0.7	0.7	0.6
1	0.3	0.2	0.1
2	0.9	0.6	0.6
3	1.0	0.9	0.8
4	1.6	1.5	1.4
5	2.2	2.2	2.0
⋮	⋮	⋮	⋮
11	5.8	5.8	5.6
12	6.4	6.4	6.2
13	7.0	7.0	6.8
14	7.6	7.6	7.4
15	8.2	8.2	8.0
⋮	⋮	⋮	⋮

TABLE 3. Example 23.

REMARK. In the previous examples we did not consider the case that $\det P'(z) \equiv 0$ and $z^\star$ is an isolated zero of P. In fact, this situation cannot occur. Proposition 52 immediately implies that $\det P'(z)$ cannot be identically equal to zero near $z^\star$ if $z^\star$ is an isolated zero of P. Thus $\det P'(z) \equiv 0$ implies that $z^\star$ is not an isolated zero of P.

Let $d_1, \ldots, d_n \in \mathbb{C}_0$. Consider the iteration

$$z^{(p+1)} = z^{(p)} - [f'(z^{(p)})]^{-1} \operatorname{diag}(d_1, \ldots, d_n) f(z^{(p)}), \qquad p = 0, 1, 2, \ldots, \tag{72}$$

or, equivalently,

$$f'(z^\star + e^{(p)}) e^{(p+1)} = f'(z^\star + e^{(p)}) e^{(p)} - \operatorname{diag}(d_1, \ldots, d_n) f(z^\star + e^{(p)}), \tag{73}$$

where $e^{(p)} := z^{(p)} - z^\star$ for $p = 0, 1, 2, \ldots$. Let $g_j(e^{(p)})$ be the jth component of the vector appearing in the right-hand side of (73). Using the same reasoning as in the proof of Theorem 56, one can easily show that

$$g_j(e^{(p)}) = \sum_{|\alpha| \geq k_j} \bigl(|\alpha| - d_j\bigr)\, c_{j,\alpha} \bigl[e^{(p)}\bigr]^\alpha.$$

It follows that $|g_j(e^{(p)})| = \mathcal{O}(\|e^{(p)}\|^{k_j})$ for $j = 1, \ldots, n$ and thus the iteration (72) converges only linearly to $z^\star$ (if $\det P'(z) \not\equiv 0$ and if $z^{(0)}$ is sufficiently close to $z^\star$). The special choice $d_1 = k_1, \ldots, d_n = k_n$ gives quadratic convergence. But of course, the orders $k_1, \ldots, k_n$ are usually not known in advance.

4. Van de Vel's method

The following proposition will help us to devise an iteration for the unknown orders $k_1, \ldots, k_n$.

PROPOSITION 57. *Let $e \in \mathbb{C}^n$. Then $P'(z^\star + e)e = \operatorname{diag}(k_1, \ldots, k_n) P(z^\star + e)$.*

PROOF. Suppose $e =: (e_1, \ldots, e_n)$. Then the jth component of $P'(z^\star + e)e$ is given by

$$\sum_{k=1}^{n} \frac{\partial P_j}{\partial z_k}(z^\star + e)e_k = \sum_{|\alpha|=k_j} |\alpha| c_{j,\alpha} e^\alpha = k_j P_j(z^\star + e)$$

for $j = 1, \ldots, n$. This proves the proposition. □

Let $K := \operatorname{diag}(k_1, \ldots, k_n)$ and suppose $z^{(p)}$ is our current approximation to $z^\star$. Then, by the previous proposition,

$$P'(z^{(p)})(z^{(p)} - z^\star) = KP(z^{(p)}).$$

The next iterate $z^{(p+1)}$ is defined as

$$z^{(p+1)} := z^{(p)} - [f'(z^{(p)})]^{-1} D^{(p)} f(z^{(p)}) \tag{74}$$

where $D^{(p)} := \operatorname{diag}(d_1^{(p)}, \ldots, d_n^{(p)})$ contains our current approximations to the orders $k_1, \ldots, k_n$. Then

$$P'(z^{(p+1)})(z^{(p+1)} - z^\star) = KP(z^{(p+1)}).$$

Suppose that the matrices $P'(z^{(p)})$ and $P'(z^{(p+1)})$ are regular. Then

$$z^{(p)} - z^\star = [P'(z^{(p)})]^{-1} KP(z^{(p)}) \tag{75}$$

and

$$z^{(p+1)} - z^\star = [P'(z^{(p+1)})]^{-1} KP(z^{(p+1)}). \tag{76}$$

By subtracting (75) and (76), and using (74) we obtain that

$$[P'(z^{(p)})]^{-1} KP(z^{(p)}) - [P'(z^{(p+1)})]^{-1} KP(z^{(p+1)}) = [f'(z^{(p)})]^{-1} D^{(p)} f(z^{(p)}).$$

If we replace P by f then this relation will be satisfied only approximatively. We use the resulting equation to define our next approximation $D^{(p+1)}$ to K:

$$D^{(p+1)} f(z^{(p)}) - f'(z^{(p)})[f'(z^{(p+1)})]^{-1} D^{(p+1)} f(z^{(p+1)}) = D^{(p)} f(z^{(p)}).$$

This equation is solved for the diagonal matrix $D^{(p+1)}$ in the following way. Let

$$F(z) := \operatorname{diag}(f_1(z), \ldots, f_n(z)), \qquad d^{(p)} := \begin{bmatrix} d_1^{(p)} \\ \vdots \\ d_n^{(p)} \end{bmatrix}$$

and define $d^{(p+1)}$ in a similar way. Obviously

$$D^{(p)} f(z^{(p)}) = F(z^{(p)}) d^{(p)}, \qquad D^{(p+1)} f(z^{(p+1)}) = F(z^{(p+1)}) d^{(p+1)}, \qquad \text{etc.}$$

Therefore

$$\big[F(z^{(p)}) - f'(z^{(p)})[f'(z^{(p+1)})]^{-1} F(z^{(p+1)}) \big] d^{(p+1)} = F(z^{(p)}) d^{(p)}. \tag{77}$$

This is the formula that we will use to calculate $d^{(p+1)}$ from $d^{(p)}$, $z^{(p)}$ and $z^{(p+1)}$. If we define the matrix-valued function

$$U(z) := [f'(z)]^{-1} \operatorname{diag}(f_1(z), \ldots, f_n(z))$$

for every $z \in \mathbb{C}^n$ such that $f'(z)$ is regular, (77) can be written as

$$\big[U(z^{(p)}) - U(z^{(p+1)}) \big] d^{(p+1)} = U(z^{(p)}) d^{(p)}. \tag{78}$$

This is a multidimensional version of the iteration formula that was discovered by Van de Vel [**281, 283**]. Note the diagonal matrix in the definition of $U(z)$. If $n = 1$ then $U(z) = f(z)/f'(z)$. In the multidimensional case it is tempting to consider the vector-valued function $[f'(z)]^{-1}f(z)$ but, as we have just found out, one should replace the vector $f(z)$ by its corresponding diagonal matrix, to obtain a matrix-valued function $U(z)$. Now the iteration (74) can be written as

$$z^{(p+1)} = z^{(p)} - U(z^{(p)})d^{(p)}.$$

From the foregoing considerations we extract the following iterative procedure:

$$\begin{aligned} d^{(p+1)} &= \big[\, U(z^{(p)}) - U\big(z^{(p)} - U(z^{(p)})d^{(p)}\big)\,\big]^{-1} U(z^{(p)})d^{(p)} \\ z^{(p+1)} &= \big(z^{(p)} - U(z^{(p)})d^{(p)}\big) - U\big(z^{(p)} - U(z^{(p)})d^{(p)}\big)d^{(p+1)} \end{aligned}$$

for $p = 0, 1, 2, \ldots$, starting with initial estimates $z^{(0)}$ for the zero $z^\star$ and $d^{(0)}$ for the orders $[k_1 \quad \cdots \quad k_n]^T$. This is our generalization of Van de Vel's method. It is a two-point method with memory. An equivalent formulation is the following:

$$\begin{aligned} z^{(p+1/2)} &= z^{(p)} - U(z^{(p)})d^{(p)} \\ d^{(p+1)} &= \big[\, U(z^{(p)}) - U(z^{(p+1/2)})\,\big]^{-1} U(z^{(p)})d^{(p)} \\ z^{(p+1)} &= z^{(p+1/2)} - U(z^{(p+1/2)})d^{(p+1)} \end{aligned}$$

for $p = 0, 1, 2, \ldots$. This leads to the following algorithm.

ALGORITHM (two-point version).
input $z^{(0)}$, $d^{(0)}$
for $p = 0, 1, 2, \ldots$
1. Solve $f'(z^{(p)})\Delta z^{(p)} = -\operatorname{diag}(d^{(p)})f(z^{(p)})$
   $z^{(p+1/2)} \leftarrow z^{(p)} + \Delta z^{(p)}$
2. Solve $\big[F(z^{(p)}) - f'(z^{(p)})[f'(z^{(p+1/2)})]^{-1}F(z^{(p+1/2)})\big]d^{(p+1)} = F(z^{(p)})d^{(p)}$
3. Solve $f'(z^{(p+1/2)})\Delta z^{(p+1/2)} = -\operatorname{diag}(d^{(p+1)})f(z^{(p+1/2)})$
   $z^{(p+1)} \leftarrow z^{(p+1/2)} + \Delta z^{(p+1/2)}$

This method can be improved by noting that after step 1 is completed the first time, there is no reason ever to return to it. Instead the estimate of the orders can be improved (step 2) before each and every further quasi-Newton step (step 3). Thus the improved iteration may be written as

$$\begin{aligned} d^{(p+1)} &= \big[\, U(z^{(p)}) - U(z^{(p+1)})\,\big]^{-1} U(z^{(p)})d^{(p)} \\ z^{(p+2)} &= z^{(p+1)} - U(z^{(p+1)})d^{(p+1)} \end{aligned}$$

for $p = 0, 1, 2, \ldots$, with initial $z^{(0)}$ and $d^{(0)}$, and after one preliminary quasi-Newton step $z^{(1)} = z^{(0)} - U(z^{(0)})d^{(0)}$. This one-point method with memory corresponds to King's improvement of Van de Vel's method [**177**].

ALGORITHM (one-point version).
input $z^{(0)}$, $d^{(0)}$
Solve $f'(z^{(0)})\Delta z^{(0)} = -\operatorname{diag}(d^{(0)})f(z^{(0)})$
$z^{(1)} \leftarrow z^{(0)} + \Delta z^{(0)}$
for $p = 0, 1, 2, \ldots$

	two-point version						one-point version						
p	k_1	k_2	$\|k\|$	$z_1^\star$	$z_2^\star$	$\|z^\star\|$	k_1	k_2	$\|k\|$	$z_1^\star$	$z_2^\star$	$\|z^\star\|$	p
0				0.7	0.7	0.5				0.7	0.7	0.5	0
1/2				0.9	2.0	0.9	0.7	0.7	0.7	0.9	2.0	0.9	1
1	0.7	0.7	0.7	1.6	3.4	1.6	2.6	0.9	1.3	1.6	3.4	1.6	2
3/2				2.3	4.0	2.3	4.0	1.6	2.0	4.2	4.2	4.0	3
2	4.0	1.7	2.0	6.4	5.6	5.6	4.6	4.2	4.4	8.0	5.8	5.8	4
5/2				10.2	7.3	7.3	1.7	8.0	1.8	9.8	10.0	9.7	5
3	4.4	6.3	4.5	12.0	13.6	11.9	10.8	9.8	10.4	11.5	18.0	11.5	6
7/2				16.3	20.0	16.3	23.8	11.5	11.9	22.3	27.8	22.3	7
4	17.4	12.0	12.3	33.7	31.9	31.9	22.5	22.3	22.7	46.1	39.3	39.3	8
9/2				51.0	44.0	43.8	⋮	⋮	⋮	72.2	61.6	61.6	9
5	28.5	33.7	28.6	79.5	77.5	77.5				⋮	⋮	⋮	⋮
⋮				⋮	⋮	⋮							

TABLE 4. Example 24.

1. Solve $\big[F(z^{(p)}) - f'(z^{(p)})[f'(z^{(p+1)})]^{-1}F(z^{(p+1)})\big]d^{(p+1)} = F(z^{(p)})d^{(p)}$
2. Solve $f'(z^{(p+1)})\Delta z^{(p+1)} = -\operatorname{diag}(d^{(p+1)})f(z^{(p+1)})$
   $z^{(p+2)} \leftarrow z^{(p+1)} + \Delta z^{(p+1)}$

EXAMPLE 24. Let us reconsider the mapping of Example 21. Table 4 compares the two-point version of our algorithm with the one-point version. The columns "k_1" and "k_2" contain $-\log_{10}|d_1^{(p)} - k_1|/k_1$ and $-\log_{10}|d_2^{(p)} - k_2|/k_2$, respectively. The columns "$z_1^\star$" and "$z_2^\star$" contain $-\log_{10}|z_1^{(p)}|$ and $-\log_{10}|z_2^{(p)}|$, respectively. (Remember that $z^\star = (0,0)$.) The columns "$\|k\|$" and "$\|z^\star\|$" contain

$$-\log_{10}\sqrt{\frac{|d_1^{(p)} - k_1|^2 + |d_2^{(p)} - k_2|^2}{k_1^2 + k_2^2}} \quad \text{and} \quad -\log_{10}\sqrt{|z_1^{(p)}|^2 + |z_2^{(p)}|^2}$$

respectively. The initial iterates were $z^{(0)} := (0.2, 0.2)$ and $d^{(0)} := (1,1)$. The one-point version is superior. Intuitively this is not a surprise, of course.

EXAMPLE 25. Next we reconsider the mapping of Example 22, but shifted to the point $z^\star = (1,3)$. In other words, suppose

$$f = (f_1, f_2) = (uv + (\sin u)^2 + v^3, \sin u \sin v)$$

where $u = z_1 - 1$ and $v = z_2 - 3$. Table 5 compares both versions of our algorithm. The columns labelled "$z_1^\star$", "$z_2^\star$" and "$\|z^\star\|$" are now related to the (componentwise or normwise) *relative* errors. The initial iterates were $z^{(0)} := (1.1, 3.1)$ and $d^{(0)} := (1,1)$. Again the one-point version is superior.

EXAMPLE 26. The mapping

$$f = (f_1, f_2, f_3) =$$
$$(u^2 + u^2 \sin v + u^3 \sin w, v + uv + v^2 + u^2 \sin u, w^2 + u^3 + vw \sin w + v^4 + u^5)$$

	two-point version						one-point version						
p	k_1	k_2	$\|k\|$	$z_1^\star$	$z_2^\star$	$\|z^\star\|$	k_1	k_2	$\|k\|$	$z_1^\star$	$z_2^\star$	$\|z^\star\|$	p
0				1.0	1.5	1.3				1.0	1.5	1.3	0
1/2				1.3	1.8	1.7	1.5	2.6	1.6	1.3	1.8	1.7	1
1	1.5	2.6	1.6	2.6	3.1	2.9	2.0	3.1	2.2	2.6	3.1	2.9	2
3/2				5.6	5.5	5.5	1.8	5.6	1.9	5.5	6.5	5.9	3
2	1.8	4.2	2.0	7.0	6.8	6.9	7.3	9.1	7.4	7.4	8.4	7.9	4
5/2				8.1	8.0	8.0	9.5	15.6	9.6	14.6	15.6	15.0	5
3	6.1	6.8	6.2	13.8	13.6	13.6	16.3	25.8	16.5	24.2	25.2	24.6	6
7/2				19.2	19.0	19.0				⋮	⋮	⋮	⋮
4	12.8	17.7	13.0	⋮	⋮	⋮							

TABLE 5. Example 25.

p	k_1	k_2	k_3	$\|k\|$	$z_1^\star$	$z_2^\star$	$z_3^\star$	$\|z^\star\|$
0					0.7	1.0	1.4	1.2
1	0.6	0.4	0.5	0.5	0.9	1.7	1.6	1.5
2	1.6	0.8	1.3	1.2	1.7	2.5	2.1	2.1
3	2.7	1.7	2.2	2.1	3.2	3.3	3.4	3.4
4	3.3	2.8	3.2	3.1	5.8	5.0	5.6	5.3
5	4.9	4.6	4.9	4.8	9.1	7.7	8.8	8.1
6	7.7	7.4	7.7	7.6	14.0	12.3	13.7	12.7
7	12.3	12.0	12.3	12.2	21.7	19.7	21.4	20.1
8	19.7	19.3	19.7	19.6	34.0	32.0	33.7	32.1
⋮	⋮	⋮	⋮	⋮	54.0	51.0	53.3	51.4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

TABLE 6. Example 26.

where $u = z_1 - 1$, $v = z_2 - 2$ and $w = z_3 - 5$ has an isolated zero at $z^\star = (1, 2, 5)$. The orders are $k_1 = 2$, $k_2 = 1$ and $k_3 = 2$. The homogeneous principal part of f at $z^\star$ is given by $P = P(z_1, z_2, z_3) = ((z_1 - 1)^2, z_2 - 2, (z_3 - 5)^2)$. It follows that $z^\star$ is an isolated zero of P and thus, according to Theorem 55, the multiplicity of $z^\star$ as a zero of f is equal to $k_1 \cdot k_2 \cdot k_3 = 4$. We have used the one-point version of our algorithm. Table 6 contains minus the logarithm with base 10 of the (componentwise or normwise) relative errors. The initial iterates were $z^{(0)} := (1.2, 2.2, 5.2)$ and $d^{(0)} := (1, 1, 1)$.

EXAMPLE 27. In our last example we consider a mapping that has a simple zero. The mapping $f = (f_1, f_2, f_3) = (u + u^2 + vw + \sin u \sin w + v^3, v + uv + v^2 + vw + (\sin u)^3 + vw^2, w + uw + w^2 + u^2 \sin v + w^3)$ where $u = z_1 - 1$, $v = z_2 - 2$ and $w = z_3 - 5$ has an isolated zero at $z^\star = (1, 2, 5)$. The orders are $k_1 = 1$, $k_2 = 1$ and $k_3 = 1$. The homogeneous principal part of f at $z^\star$ is given by $P = P(z_1, z_2, z_3) = (z_1 - 1, z_2 - 2, z_3 - 5)$. It follows that $z^\star$ is an isolated zero of P

p	k_1	k_2	k_3	$\|k\|$	$z_1^\star$	$z_2^\star$	$z_3^\star$	$\|z^\star\|$
0					0.7	1.0	1.4	1.2
1	0.3	0.2	0.3	0.2	1.2	1.5	2.0	1.7
2	1.1	1.1	1.2	1.1	1.7	1.9	2.4	2.2
3	1.2	1.2	1.5	1.3	2.5	2.7	3.4	3.0
4	2.1	2.0	2.2	2.1	3.7	3.8	4.8	4.2
5	3.4	3.2	3.5	3.3	5.7	5.8	7.0	6.2
6	5.5	5.3	5.6	5.4	9.1	9.1	10.5	9.5
7	8.9	8.6	9.0	8.8	14.6	14.4	16.2	14.8
8	14.4	14.0	14.6	14.1	23.5	23.0	25.2	23.4
9	23.3	22.6	23.5	22.8	38.0	36.9	39.8	37.3
10	⋮	⋮	⋮	⋮	61.2	59.5	63.3	69.9
⋮					⋮	⋮	⋮	⋮

TABLE 7. Example 27.

and thus, according to Theorem 55, the multiplicity of $z^\star$ as a zero of f is equal to $k_1 \cdot k_2 \cdot k_3 = 1$. Therefore $\det f'(z^\star) \neq 0$. The iteration of Theorem 56 reduces to the classical Newton's method. Table 7 shows the performance of the one-point version of our algorithm. The initial iterates were $z^{(0)} := (1.2, 2.2, 5.2)$ and $d^{(0)} := (1, 1, 1)$.

REMARK. As soon as the iterates $d_1^{(p)}, \ldots, d_n^{(p)}$ are sufficiently close to integers, one can determine the correct values of the orders $k_1, \ldots, k_n$ and use the iteration of Theorem 56, of course.

REMARK. King [**177**] analysed Van de Vel's method [**281**], rearranged the order of the calculations, and gave a convergence proof for both iterations. He proved that Van de Vel's method has order of convergence 1.554, and that his modification has order of convergence 1.618. The previous examples indicate that our multidimensional generalizations of these methods have the same corresponding order of convergence. Unfortunately, we have been unable to generalize King's proof to the multidimensional setting. The main problem in the multidimensional case is that matrices occur instead of scalars and therefore commutativity gets lost.

5. Conclusions

In this chapter we have presented two iterations for computing multiple zeros of analytic mappings and studied their order of convergence.

The iteration of Theorem 56 is a multidimensional generalization of iteration (64). It can be used if the orders, which are positive integers related to the multiplicity of the zero, are known in advance. Under certain assumptions this iteration converges quadratically.

Our generalization of Van de Vel's iteration is based on (78). The algorithm requires no prior knowledge about the multiplicity of the zero, which may even be simple, and proceeds by iterating on the zero as well as on the orders. Our numerical experiments indicate that the one-point version has order of convergence 1.618.

Several challenging open problems remain:

- We gave only an informal derivation of our multidimensional generalization (78) of Van de Vel's iteration, and the question remains whether it is possible to generalize King's proof.
- Local convergence regions: what is the structure of the set of initial iterates $z^{(0)}$ that lie in a neighbourhood of the zero $z^\star$ and guarantee convergence?
- In the one-dimensional case the (discrete) dynamics of Newton's method are studied in terms of Julia and Fatou sets, Siegel disks, etc. (See, for example, [**25**] or [**38**].) The beautiful fractal-like images that illustrate these results are well-known. A global convergence analysis for the iterations presented in this chapter, in particular with respect to the role played by the singular manifold $\{ z \in \mathbb{C}^n : \det f'(z) = 0 \}$, is a very interesting (but very difficult!) challenge.
- Newton's method has been studied in terms of its continuous counterpart: a system of autonomous differential equations whose Euler discretization yields (the damped) Newton's method. (See, for example, [**211**], [**231**] or [**272**].) Is it possible to formulate a continuous-time version of the iterations presented in this chapter, in particular for our generalization of Van de Vel's method, and how do the regions of attraction of these dynamical systems relate to the convergence regions of the discrete methods?

Acknowledgements

The results in this chapter were obtained in collaboration with Ann Haegemans.

I would like to thank Andreas Griewank for pointing out a flaw in the proof of a previous version of Theorem 56. I also thank José Manuel Gutiérrez, Hugo Van de Vel and Pierre Verlinden for stimulating discussions.

Part 4

CHAPTER 8

Superfast rational interpolation

We have already encountered rational interpolation at the end of Chapter 2 where we have used it to locate clusters of zeros of analytic functions.

In Section 3 of Chapter 2 we have already given a brief overview of the main papers concerning rational interpolation. A more complete list includes the papers [**13, 14, 15, 18, 20, 21, 23, 27, 28, 29, 30, 31, 50, 59, 60, 73, 82, 100, 101, 114, 115, 116, 117, 124, 125, 126, 127, 131, 168, 207, 212, 249, 251, 252, 257, 258, 260, 274, 275, 294, 297, 298, 299, 301, 302, 303, 304**]. We will not discuss these algorithms. Instead, we will focus on the algorithm of Van Barel and Bultheel [**274**]. It forms the basis for the algorithms that we will present in this chapter. Most of the algorithms for solving rational interpolation problems are likely to suffer from instabilities when applied in floating point arithmetic. This holds also for the algorithm of Van Barel and Bultheel. Hence, stabilizing techniques are necessary. As we will see, the algorithm of Van Barel and Bultheel has the advantage that it can be stabilized quite easily. In Section 1 we will show how pivoting can be incorporated into the algorithm. In Section 2 we will digress slightly to present a matrix or 'block' version of the algorithm. In Section 3 we will use a divide and conquer approach to obtain a recursive algorithm that is stabilized by giving what we will call "difficult interpolation points" a special treatment, via iterative refinement based on a formula for the inverse of a coupled Vandermonde matrix and via downdating.

This chapter is closely related to the next chapter, in which we will present fast and superfast solvers for linear systems of equations that have Hankel or Toeplitz structure. The chapters can be read independently from each other but one can also proceed as follows. After one has read Section 1 of this chapter (on how to incorporate pivoting into the algorithm of Van Barel and Bultheel), one may go directly to Section 1 of the next chapter (concerning a fast Hankel solver). Then one may return to Section 2 of this chapter (the block version of the results presented in Section 1) and then read Section 2 of the next chapter (a fast block Hankel solver). Finally, one may read Section 3 of this chapter (a stabilized divide and conquer approach) and then continue with Sections 3 and 4 (a superfast Hankel solver and a superfast Toeplitz solver) of the next chapter.

Part of this chapter is contained in our papers [**187**], [**189**], [**278**] and [**279**].

1. The algorithm RatInt

Let n be a positive integer. Suppose that the complex numbers $s_1, \ldots, s_{2n} \in \mathbb{C}$ as well as the complex column vectors $f_1, \ldots, f_{2n} \in \mathbb{C}^{2\times 1}$ are given. Consider the

interpolation problem

$$f_k^T B(s_k) = [\,0\;\;0\,], \qquad k = 1, \ldots, 2n, \tag{79}$$

where

$$B(z) = \begin{bmatrix} n_\ell(z) & n_r(z) \\ d_\ell(z) & d_r(z) \end{bmatrix} \in \mathbb{C}[z]^{2\times 2}$$

with $\deg n_\ell(z) = n$, $\deg d_\ell(z) \le n-1$, $\deg n_r(z) \le n-1$ and $\deg d_r(z) = n$, and $n_\ell(z)$ as well as $d_r(z)$ monic, i.e., where $B(z)$ is a monic 2×2 matrix polynomial of degree n. Let us assume that this interpolation problem has a unique solution $B^\star(z)$. In the next chapter we will encounter linearized rational interpolation problems that can easily be formulated in this way.

Algorithm RatInt calculates a 2×2 matrix polynomial $B(z)$ of degree n that satisfies (79) and whose highest degree coefficient $A \in \mathbb{C}^{2\times 2}$ is nonsingular. This implies that $B(z) \equiv B^\star(z)A$. We will show below that $\det A = 1$. This algorithm is a straightforward adaptation of the algorithm given by Van Barel and Bultheel [**274**]. Therefore we state it without proof. Our version incorporates pivoting, which is an important advantage as it enhances the numerical stability.

Define the column vectors $s, L_s, R_s \in \mathbb{C}^{2n\times 1}$ as

$$s := \begin{bmatrix} s_1 \\ \vdots \\ s_{2n} \end{bmatrix}, \quad L_s := \begin{bmatrix} f_1(1) \\ \vdots \\ f_{2n}(1) \end{bmatrix} \quad \text{and} \quad R_s := \begin{bmatrix} f_1(2) \\ \vdots \\ f_{2n}(2) \end{bmatrix}.$$

function $[\,B(z)\,] \leftarrow$ RatInt$(s, L_s, R_s, 2n)$

$B(z) \leftarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
status $\leftarrow 1$
-- status $= 1$: both the left and the right choice are possible
-- status $= 2$: only the left choice is possible
-- status $= 3$: only the right choice is possible
-- flow $= 1$: L_s contains the pivot element
-- flow $= 2$: R_s contains the pivot element
for $j = 1 : 2n$
  select case (status)
    case (1)
      1.1a $[\text{maxR}, \text{pivR}] \leftarrow \max\limits_{j\le k\le 2n} \{\max\{|\operatorname{Re} R_s(k)|, |\operatorname{Im} R_s(k)|\}\}$
      1.1b $[\text{maxL}, \text{pivL}] \leftarrow \max\limits_{j\le k\le 2n} \{\max\{|\operatorname{Re} L_s(k)|, |\operatorname{Im} L_s(k)|\}\}$
        if maxR $>$ maxL **then**
          1.1c piv $\leftarrow$ pivR; flow $\leftarrow 2$; status $\leftarrow 2$
        else
          1.1d piv $\leftarrow$ pivL; flow $\leftarrow 1$; status $\leftarrow 3$
        end if
    case (2)
      1.2a $[\text{maxL}, \text{pivL}] \leftarrow \max\limits_{j\le k\le 2n} \{\max\{|\operatorname{Re} L_s(k)|, |\operatorname{Im} L_s(k)|\}\}$
      1.2b piv $\leftarrow$ pivL; flow $\leftarrow 1$; status $\leftarrow 1$

case (3)

1.3a $[\text{maxR}, \text{pivR}] \leftarrow \max_{j \le k \le 2n} \{\max\{|\operatorname{Re} R_s(k)|, |\operatorname{Im} R_s(k)|\}\}$

1.3b piv $\leftarrow$ pivR; flow $\leftarrow 2$; status $\leftarrow 1$

end select

$s(j) \leftrightarrow s(\text{piv})$; $L_s(j) \leftrightarrow L_s(\text{piv})$; $R_s(j) \leftrightarrow R_s(\text{piv})$

select case (flow)

case (1)

2.1a $\mu \leftarrow R_s(j)/L_s(j)$

2.1b $B(z) \leftarrow B(z) \begin{bmatrix} z - s_j & -\mu \\ 0 & 1 \end{bmatrix}$

for $k = j + 1 : 2n$

2.1c $L_s(k) \leftarrow (s_k - s_j) L_s(k)$

2.1d $R_s(k) \leftarrow -\mu L_s(k) + R_s(k)$

end for

case (2)

2.2a $\mu \leftarrow L_s(j)/R_s(j)$

2.2b $B(z) \leftarrow B(z) \begin{bmatrix} 1 & 0 \\ -\mu & z - s_j \end{bmatrix}$

for $k = j + 1 : 2n$

2.2c $L_s(k) \leftarrow L_s(k) - \mu R_s(k)$

2.2d $R_s(k) \leftarrow (s_k - s_j) R_s(k)$

end for

end select

end for

We have written down steps 1.1a, 1.1b, 1.2a and 1.3a using standard Matlab notation: the maxima are computed together with their location.

If the algorithm first performs a "left" step (flow $= 1$) and then a "right" step (flow $= 2$), then the highest degree coefficient (h.d.c.) of $B(z)$ is multiplied on the right by

$$\text{h. d. c.} \begin{bmatrix} z - s & -\mu \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\mu' & z - s' \end{bmatrix} = \begin{bmatrix} 1 & -\mu \\ 0 & 1 \end{bmatrix}.$$

If, on the other hand, the algorithm first performs a right step and then a left step, then the h.d.c. of $B(z)$ is multiplied on the right by

$$\text{h. d. c.} \begin{bmatrix} 1 & 0 \\ -\mu & z - s \end{bmatrix} \begin{bmatrix} z - s' & -\mu' \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\mu & 1 \end{bmatrix}.$$

Together with the initialization of $B(z)$ this implies that the determinant of the h.d.c of $B(z)$ is equal to 1.

The algorithm requires $\mathcal{O}(n^2)$ flops. We postpone a more detailed discussion of the arithmetic complexity of RatInt to Subsection 1.3 of the next chapter.

The multipliers μ that have to be computed in step 2.1a or 2.2a are well defined. Indeed, consider the set $\mathcal{S}$ of all the column vector polynomials $w(z) \in \mathbb{C}[z]^{2\times 1}$ that satisfy the interpolation conditions

$$f_k^T w(s_k) = 0, \qquad k = 1, \ldots, 2n. \tag{80}$$

If $w(z) \in \mathbb{C}[z]^{2\times 1}$ is an arbitrary vector polynomial, then the left-hand side of (80) is called the *residual* with respect to w at the interpolation point s_k. Every element of $\mathcal{S}$ is thus such that the residuals at all the interpolation points are equal to zero.

Note that initially the vectors L_s and R_s contain the residuals with respect to the left and right column vector polynomials corresponding to the initialization of $B(z)$. The algorithm proceeds by pivoting (to change the order of the interpolation points s, which results in a corresponding change in the order of the entries of L_s and R_s), by updating $B(z)$ in such a way that the residuals at the selected interpolation point become zero and by updating the residuals at the remaining interpolation points.

If after the execution of step $j < 2n$ all the residuals in R_s or L_s would be equal to zero, then the interpolation problem (80) would have a solution of degree $< n$. This is impossible, as we will now show. The set $\mathcal{S}$ forms a submodule of the $\mathbb{C}[z]$-module $\mathbb{C}[z]^{2\times 1}$. A basis for $\mathcal{S}$ always consists of exactly two elements [**275**, Theorem 3.1]. Let $\{B_1(z), B_2(z)\}$ be a basis for $\mathcal{S}$. Then every element $w(z) \in \mathcal{S}$ can be written in a unique way as $w(z) = \alpha_1(z)B_1(z) + \alpha_2(z)B_2(z)$ with $\alpha_1(z), \alpha_2(z) \in \mathbb{C}[z]$. The matrix polynomial $B(z) := [\, B_1(z)\ B_2(z)\,] \in \mathbb{C}[z]^{2\times 2}$ is called a *basis matrix*. Basis matrices can be characterized as follows.

THEOREM 58. *A matrix polynomial $C(z) = [\, C_1(z)\ C_2(z)\,] \in \mathbb{C}[z]^{2\times 2}$ is a basis matrix if and only if $C_1(z), C_2(z) \in \mathcal{S}$ and $\deg \det C(z) = 2n$.*

PROOF. This follows immediately from [**275**, Theorem 4.1]. □

Note that $B^\star(z)$ is a basis matrix.

A matrix polynomial is called *column reduced* if the highest degree coefficients of its column vector polynomials are linearly independent. Every basis matrix can be transformed into a column reduced basis matrix [**275**, p. 455]. Note that $B^\star(z)$ is column reduced.

THEOREM 59. *Let $\delta_1 := \deg B_1(z)$ and $\delta_2 := \deg B_2(z)$. If $B(z)$ is column reduced, then every element $w(z) \in \mathcal{S}$ having degree $\leq \delta$ can be written in a unique way as $w(z) = \alpha_1(z)B_1(z) + \alpha_2(z)B_2(z)$ with $\alpha_1(z) \in \mathbb{C}[z]$, $\deg \alpha_1(z) \leq \delta - \delta_1$ and $\alpha_2(z) \in \mathbb{C}[z]$, $\deg \alpha_2(z) \leq \delta - \delta_2$.*

PROOF. See [**275**, Theorem 3.2]. □

COROLLARY 60. *The interpolation problem* (80) *has no nontrivial solution of degree $< n$.*

PROOF. $B^\star(z)$ is a column reduced basis matrix. Its column degrees are equal to n. Suppose that $\delta < n$. Then $\deg \alpha_1(z) < 0$ and $\deg \alpha_2(z) < 0$ and thus $\alpha_1(z) \equiv 0$ and $\alpha_2(z) \equiv 0$. This implies that $w(z) \equiv [\,0\ \ 0\,]^T$. □

2. The algorithms BLOCKRATINT$_\lambda$ and BLOCKRATINT$_\rho$

We will now formulate a matrix or 'block' version of the results presented in the previous section.

Let n and p be positive integers. Let the complex numbers $s_1, \ldots, s_{2n} \in \mathbb{C}$ and the block column vectors $f_{\lambda,1}, \ldots, f_{\lambda,2n} \in \mathbb{C}^{2p\times p}$ be given. Consider the interpolation

problem

$$f_{\lambda,k}^T B_\lambda(s_k) = \begin{bmatrix} O_p & O_p \end{bmatrix}, \qquad k = 1, \ldots, 2n, \tag{81}$$

where

$$B_\lambda(z) = \begin{bmatrix} N_\ell(z) & N_r(z) \\ D_\ell(z) & D_r(z) \end{bmatrix} \in \mathbb{C}[z]^{2p \times 2p}$$

where $\deg N_\ell(z) = n$, $\deg D_\ell(z) \leq n-1$, $\deg N_r(z) \leq n-1$ and $\deg D_r(z) = n$, and $N_\ell(z)$ as well as $D_r(z)$ monic, i.e., where $B_\lambda(z)$ is a monic $2p \times 2p$ block matrix polynomial of degree n. Let us assume that this interpolation problem has a unique solution $B_\lambda^\star(z)$. Similarly, let the block column vectors $f_{\rho,1}, \ldots, f_{\rho,2n} \in \mathbb{C}^{2p \times p}$ be given and consider the interpolation problem

$$B_\rho(s_k) f_{\rho,k} = \begin{bmatrix} O_p \\ O_p \end{bmatrix}, \qquad k = 1, \ldots, 2n, \tag{82}$$

where

$$B_\rho(z) = \begin{bmatrix} N_u(z) & D_u(z) \\ N_\ell(z) & D_\ell(z) \end{bmatrix} \in \mathbb{C}[z]^{2p \times 2p}$$

where $\deg N_u(z) = n$, $\deg D_u(z) \leq n-1$, $\deg N_\ell(z) \leq n-1$ and $\deg D_\ell(z) = n$, and $N_u(z)$ as well as $D_\ell(z)$ monic, i.e., where $B_\rho(z)$ is a monic $2p \times 2p$ block matrix polynomial of degree n. Let us assume that this interpolation problem has a unique solution $B_\rho^\star(z)$. In the next chapter we will encounter linearized rational interpolation problems that can easily be formulated in this way.

The algorithm BLOCKRATINT$_\lambda$ calculates a $2p \times 2p$ matrix polynomial $B_\lambda(z)$ of degree n that satisfies (81) and whose highest degree coefficient $A_\lambda \in \mathbb{C}^{2p \times 2p}$ is nonsingular. This implies that $B_\lambda(z) \equiv B_\lambda^\star(z) A_\lambda$. Hence, to obtain $B_\lambda^\star(z)$ we have to multiply $B_\lambda(z)$ to the right by the inverse of A_λ. We leave it to the reader to formulate an analogous algorithm BLOCKRATINT$_\rho$ for computing a $2p \times 2p$ matrix polynomial $B_\rho(z)$ of degree n that satisfies (82) and whose highest degree coefficient $A_\rho \in \mathbb{C}^{2p \times 2p}$ is nonsingular. The latter implies that $B_\rho(z) \equiv A_\rho B_\rho^\star(z)$. Below we will prove that $\det A_\lambda = \det A_\rho = 1$. These algorithms are easy generalizations of the algorithm RATINT.

Define the column vectors s and Residuals as

$$s := \begin{bmatrix} s_1 \\ \vdots \\ s_1 \\ \vdots \\ s_{2n} \\ \vdots \\ s_{2n} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} p \\ \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} p \end{matrix} \qquad \text{and} \qquad \text{Residuals} := \begin{bmatrix} f_{\lambda,1}^T \\ \vdots \\ f_{\lambda,2n}^T \end{bmatrix}.$$

function $[B_\lambda(z)] \leftarrow$ BLOCKRATINT$_\lambda(s, \text{Residuals}, 2n, p)$

$$B_\lambda(z) \leftarrow \begin{bmatrix} I_p & 0_p \\ 0_p & I_p \end{bmatrix}$$

$\mathcal{C} \leftarrow \{1, \ldots, 2p\}$
for $j = 1 : 2pn$

1. Determine piv $\in \{j, \ldots, 2pn\}$ and col $\in \mathcal{C}$ such that
   $\max\{|\operatorname{Re}\text{Residuals}(\text{piv}, \text{col})|, |\operatorname{Im}\text{Residuals}(\text{piv}, \text{col})|\}$
   $= \max_{\substack{j \le k \le 2pn \\ l \in \mathcal{C}}} \{\max\{|\operatorname{Re}\text{Residuals}(k, l)|, |\operatorname{Im}\text{Residuals}(k, l)|\}\}.$
2. $s(j) \leftrightarrow s(\text{piv})$; $\text{Residuals}(j, :) \leftrightarrow \text{Residuals}(\text{piv}, :)$
3. **for** l **in** $\{1, \ldots, 2p\} \setminus \{\text{col}\}$
   3.1 $\mu(l) \leftarrow \text{Residuals}(j, l)/\text{Residuals}(j, \text{col})$
   end for
   3.2 $B_\lambda(z) \leftarrow B_\lambda(z) \times$
$$\begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ -\mu(1) & \cdots & -\mu(\text{col}-1) & z - s(j) & -\mu(\text{col}+1) & \cdots & -\mu(2p) \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 \end{bmatrix}$$
   for $k = j + 1 : 2pn$
   for l **in** $\{1, \ldots, 2p\} \setminus \{\text{col}\}$
   3.3a $\text{Residuals}(k, l) \leftarrow \text{Residuals}(k, l) - \mu(l)\text{Residuals}(k, \text{col})$
   end for
   3.3b $\text{Residuals}(k, \text{col}) \leftarrow (s(k) - s(j))\text{Residuals}(k, \text{col})$
   end for
4. $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\text{col}\}$; **if** $\mathcal{C} == \emptyset$ **then** $\mathcal{C} \leftarrow \{1, \ldots, 2p\}$

end for

Step 3.2 implies that $\det B_\lambda(z) \leftarrow \det B_\lambda(z)(z - s(j))$ for $j = 1 : 2pn$. Thus

$$\det B_\lambda(z) \equiv (z - s_1)^p \cdots (z - s_{2n})^p,$$

a monic polynomial of degree $2pn$. As $\det B_\lambda(z) \equiv \det B_\lambda^\star(z) \det A_\lambda$ and $\det B_\lambda^\star(z)$ is also a monic polynomial of degree $2pn$, we may conclude that $\det A_\lambda = 1$. A similar argument shows that $\det A_\rho = 1$.

The algorithms BLOCKRATINT$_\lambda$ and BLOCKRATINT$_\rho$ each require $\mathcal{O}(p^3 n^2)$ flops. We postpone a more detailed discussion of the arithmetic complexity to Subsection 2.3 of the next chapter.

The multipliers $\mu(l)$ that have to be computed in step 3.1 are well defined. Indeed, consider the set $\mathcal{S}$ of all the column vector polynomials $w(z) \in \mathbb{C}[z]^{2p \times 1}$ that satisfy the interpolation conditions

$$f_{\lambda,k}^T w(s_k) = O_{p \times 1}, \qquad k = 1, \ldots, 2n. \tag{83}$$

If $w(z) \in \mathbb{C}[z]^{2p \times 1}$ is an arbitrary vector polynomial, then the left-hand side of (83) is called the *residual* with respect to w at the interpolation point s_k. Every element of $\mathcal{S}$ is thus such that the residuals at all the interpolation points are equal to zero.

If after the execution of step $j < 2pn$ all the residuals would be equal to zero, then the interpolation problem (83) would have a solution of degree $< n$. This is impossible, as we will now show. The set $\mathcal{S}$ forms a submodule of the $\mathbb{C}[z]$-module $\mathbb{C}[z]^{2p\times 1}$. A basis for $\mathcal{S}$ always consists of exactly $2p$ elements, i.e., the dimension of $\mathcal{S}$ is equal to $2p$ [**275**, Theorem 3.1]. Let $\{B_k(z)\}_{k=1}^{2p}$ be a basis for $\mathcal{S}$. Then every element $w(z) \in \mathcal{S}$ can be written in a unique way as $w(z) = \sum_{k=1}^{2p} \alpha_k(z) B_k(z)$ with $\alpha_k(z) \in \mathbb{C}[z]$ for $k = 1, \ldots, 2p$. The matrix polynomial $B(z) := [\, B_1(z)\ B_2(z)\ \cdots\ B_{2p}(z)\,] \in \mathbb{C}[z]^{2p\times 2p}$ is called a *basis matrix*. Basis matrices can be characterized as follows.

THEOREM 61. *A matrix polynomial* $C(z) = [\, C_1(z)\ C_2(z)\ \cdots\ C_{2p}(z)\,] \in \mathbb{C}[z]^{2p\times 2p}$ *is a basis matrix if and only if* $C_k(z) \in \mathcal{S}$ *for* $k = 1, \ldots, 2p$ *and* $\deg\det C(z) = 2pn$.

PROOF. This follows immediately from [**275**, Theorem 4.1]. □

Note that $B_\lambda^\star(z)$ is a basis matrix.

A matrix polynomial is called *column reduced* if the highest degree coefficients of its column vector polynomials are linearly independent. Every basis matrix can be transformed into a column reduced basis matrix [**275**, p. 455]. Note that $B_\lambda^\star(z)$ is column reduced.

THEOREM 62. *Let* $\delta_k := \deg B_k(z)$ *for* $k = 1, \ldots, 2p$. *If* $B(z)$ *is column reduced, then every element* $w(z) \in \mathcal{S}$ *having degree* $\leq \delta$ *can be written in a unique way as* $w(z) = \sum_{k=1}^{2p} \alpha_k(z) B_k(z)$ *with* $\alpha_k(z) \in \mathbb{C}[z]$ *and* $\deg \alpha_k(z) \leq \delta - \delta_k$ *for* $k = 1, \ldots, 2p$.

PROOF. See [**275**, Theorem 3.2]. □

COROLLARY 63. *The interpolation problem* (83) *has no nontrivial solution of degree* $< n$.

PROOF. $B_\lambda^\star(z)$ is a column reduced basis matrix. Its column degrees are equal to n. Suppose that $\delta < n$. Then $\deg \alpha_k(z) < 0$ and thus $\alpha_k(z) \equiv 0$ for $k = 1, \ldots, 2p$. This implies that $w(z) \equiv O_{2p\times 1}$. □

3. A stabilized divide and conquer approach

We will now return to the setting of Section 1. Recall that the set $\mathcal{S}$ has been defined as the set of all the column vector polynomials $w(z) \in \mathbb{C}[z]^{2\times 1}$ that satisfy the interpolation conditions

$$f_k^T w(s_k) = 0, \qquad k = 1, \ldots, 2n.$$

Within the submodule $\mathcal{S}$ we want to be able to consider solutions $w(z)$ that satisfy additional conditions concerning their degree-structure. To describe the degree-structure of a vector polynomial, we use the concept of τ-degree [**275**]. Let $\tau \in \mathbb{Z}$. The *τ-degree* of a vector polynomial $w(z) = [\, w_1(z)\ w_2(z)\,]^T \in \mathbb{C}[z]^{2\times 1}$ is defined as a generalization of the classical degree:

$$\tau\text{-}\deg w(z) := \max\{\, \deg w_1(z), \deg w_2(z) - \tau \,\}$$

with $\tau\text{-}\deg 0 := -\infty$. The τ-highest degree coefficient of a vector polynomial $[\, w_1(z)\ w_2(z)\,]^T$ with τ-degree δ is defined as the vector $[\, \omega_1\ \omega_2\,]^T$ with ω_1 the coefficient of z^δ in $w_1(z)$ and ω_2 the coefficient of $z^{\delta+\tau}$ in $w_2(z)$. A set of vector

polynomials in $\mathbb{C}[z]^{2\times 1}$ is called *τ-reduced* if the τ-highest degree coefficients are linearly independent. Every basis of $\mathcal{S}$ can be transformed into a τ-reduced one. For details, we refer to [**275**]. Once we have a basis in τ-reduced form, the elements of $\mathcal{S}$ can be parametrized as follows.

THEOREM 64. *Let $\{ B_1(z), B_2(z) \}$ be a τ-reduced basis for $\mathcal{S}$. Define $\delta_1 := \tau$-$\deg B_1(z)$ and $\delta_2 := \tau$-$\deg B_2(z)$. Then every element $w(z) \in \mathcal{S}$ having τ-degree $\leq \delta$ can be written in a unique way as*

$$w(z) = \alpha_1(z)B_1(z) + \alpha_2(z)B_2(z)$$

with $\alpha_1(z), \alpha_2(z) \in \mathbb{C}[z]$, $\deg \alpha_1(z) \leq \delta - \delta_1$ and $\deg \alpha_2(z) \leq \delta - \delta_2$.

PROOF. See Van Barel and Bultheel [**275**, Theorem 3.2]. □

The following theorem will enable us to devise an interpolation algorithm that is based on a *divide and conquer* approach. It shows how basis matrices can be coupled in case the degree-structure is important.

THEOREM 65. *Suppose K is a positive integer. Let $\sigma_1, \ldots, \sigma_K \in \mathbb{C}$ be mutually distinct and let $\phi_1, \ldots, \phi_K \in \mathbb{C}^{2\times 1}$. Suppose that $\phi_k \neq [\ 0 \quad 0\ ]^T$ for $k = 1, \ldots, K$. Let $1 \leq \kappa \leq K$. Let $\tau_K \in \mathbb{Z}$. Suppose that $B_\kappa(z) \in \mathbb{C}[z]^{2\times 2}$ is a τ_K-reduced basis matrix with basis vectors having τ_K-degree δ_1 and δ_2 respectively, corresponding to the interpolation data*

$$\{ (\sigma_k, \phi_k) \ : \ k = 1, \ldots, \kappa \}.$$

Let $\tau_{\kappa\to K} := \delta_1 - \delta_2$. Let $B_{\kappa\to K}(z) \in \mathbb{C}[z]^{2\times 2}$ be a $\tau_{\kappa\to K}$-reduced basis matrix corresponding to the interpolation data

$$\{ (\sigma_k, B_\kappa^T(\sigma_k)\phi_k) \ : \ k = \kappa+1, \ldots, K \}.$$

Then $B_K(z) := B_\kappa(z)B_{\kappa\to K}(z)$ is a τ_K-reduced basis matrix corresponding to the interpolation data

$$\{ (\sigma_k, \phi_k) \ : \ k = 1, \ldots, K \}.$$

PROOF. See Van Barel and Bultheel [**276**, Theorem 3]. □

The following algorithm implements this theorem. We start with $N_s \geq 2$ interpolation points, where N_s is a power of 2. These interpolation points are split into two equal parts. As N_s is a power of 2, this process can be repeated. At the lowest level the interpolation problems are solved by our fast solver RATINT.

recursive function $[\, B(t)\,] \leftarrow$ RECRATINT(s, L_s, R_s, N_s, τ)
-- $\tau \in \mathbb{Z}$
-- $N_s = 2^{p+1}$ for some $p \in \mathbb{N}$: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s\times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s\times 1}$: the initial left and right residual vectors
-- $B(t) \in \mathbb{C}[t]^{2\times 2}$: a τ-reduced basis matrix corresponding to
the given interpolation data
if $N_s > 2^{\text{limit}}$ **then**
$[\, s_1, L_{s_1}, R_{s_1}, s_2, L_{s_2}, R_{s_2}\,] \leftarrow$ SPLIT(s, L_s, R_s)
$[\, B_1(t)\,] \leftarrow$ RECRATINT$(s_1, L_{s_1}, R_{s_1}, N_s/2, \tau)$
for $k = 1(1)N_s/2$

$[\tilde{L}_{s_2}(k)\ \ \tilde{R}_{s_2}(k)] \leftarrow [L_{s_2}(k)\ \ R_{s_2}(k)] \cdot B_1(s_2(k))$
end for
$\tilde{\tau} \leftarrow$ the difference between the left and right τ-degrees of $B_1(t)$
$[\tilde{B}_2(t)] \leftarrow \textsc{RecRatInt}(s_2, \tilde{L}_{s_2}, \tilde{R}_{s_2}, N_s/2, \tilde{\tau})$
$B(t) \leftarrow B_1(t) \cdot \tilde{B}_2(t)$
else
$[B(t)] \leftarrow \textsc{RatInt}(s, L_s, R_s, N_s, \tau)$
end if

Note that RecRatInt calls RatInt with the parameter τ. This corresponds to an initialization that is different from the one used in the version that we have presented in Section 1. The difference is that instead of the classical degree the τ-degree is used. We omit the details.

We now have a divide and conquer algorithm for rational interpolation. This algorithm can be stabilized in three ways: by giving "difficult points" an adequate treatment, via iterative refinement and via downdating.

3.1. Difficult points. At the lowest level each interpolation problem consists of a set of interpolation conditions that are to be satisfied. Our fast algorithm RatInt constructs a solution iteratively by adding interpolation points one by one. Pivoting is used to enhance the numerical stability. Some interpolation points may have residuals with respect to the subproblem solution that are very small: the subproblem may be close to degenerate. If we process these interpolation points, then the accuracy is likely to decrease. These points are therefore marked as "difficult." They are not added at the subproblem level. Instead, they are put aside and handled only at the very end, after RecRatInt has finished, via the fast-only algorithm RatInt. If at this stage the corresponding transformed residuals are still small, this indicates that the problem is ill-conditioned. The overall complexity of our algorithm will be $\mathcal{O}(n \log^2 n)$ as long as the number of difficult points is not too large.

3.2. Iterative improvement. RecRatInt computes a basis matrix $B(z)$ that corresponds to the N_s interpolation points s having initial left and right residual vectors L_s and R_s. If there are no difficult points, then $B(z)$ has the following degree-structure:

$$\begin{bmatrix} = N_s/2 & < N_s/2 \\ < N_s/2 & = N_s/2 \end{bmatrix}.$$

In general, however, RecRatInt will discover difficult points. If the number of difficult points is equal to n_{bad}, then $B(z)$ has the degree-structure

$$\begin{bmatrix} = \alpha & < \alpha \\ < \beta & = \beta \end{bmatrix}$$

with $\alpha + \beta + n_{\text{bad}} = N_s$. Therefore $B(z)$ is not only τ-(column)reduced but also row reduced. Its row highest degree coefficient is equal to the identity matrix.

Let us introduce the following notations. The component polynomials of $B(z)$ are denoted as $a(z)$, $b(z)$, $c(z)$ and $d(z)$,

$$B(z) =: \begin{bmatrix} a(z) & c(z) \\ b(z) & d(z) \end{bmatrix}.$$

Let s^+ be a complex column vector containing all the "easy" points, i.e., all the interpolation points except the difficult ones. Let L_{s^+} and R_{s^+} be the corresponding parts of L_s and R_s. Note that these vectors have size $\alpha + \beta$. We denote their components by s_i^+, $L_{s^+,i}$ and $R_{s^+,i}$ respectively ($i = 1, \ldots, \alpha + \beta$).

Let p be a complex column vector of size $\#p$ with components p_i, $i = 1, \ldots, \#p$. Let L_p and R_p be complex column vectors of size $\#p$. Let $k, l \in \mathbb{N}$. Then $V_k(L_p, p)$ is defined as the scaled Vandermonde matrix

$$V_k(L_p, p) := \operatorname{diag}(L_p) \begin{bmatrix} p_i^j \end{bmatrix}_{i=1,\ldots,\#p}^{j=0,1,\ldots,k}.$$

Note that $V_k(L_p, p)$ is a matrix of size $\#p \times (k+1)$. The last column of $V_k(L_p, p)$ is denoted as $v_k(L_p, p)$. The *coupled Vandermonde matrix* $V_{k,l}(L_p, R_p, p)$ is defined as

$$V_{k,l}(L_p, R_p, p) := \begin{bmatrix} V_k(L_p, p) & V_l(R_p, p) \end{bmatrix}.$$

It is a matrix of size $\#p \times (k + l + 2)$.

Let $p(z) \in \mathbb{C}[z]$ be a polynomial and let $\delta \in \mathbb{N}$ be an arbitrary upper bound for its degree. Then the *stacking vector* $\hat{p}_\delta \in \mathbb{C}^{\delta+1}$ of $p(z)$ with respect to δ is defined by the equation

$$\begin{bmatrix} 1 & z & \cdots & z^\delta \end{bmatrix} \hat{p}_\delta = p(z).$$

The vector $\hat{p}'_\delta \in \mathbb{C}^\delta$ is obtained by deleting the last component of $\hat{p}_\delta$. Let p_i be the coefficient of z^i in $p(z)$ for $i = 0, 1, 2, \ldots$. Of course, $p_i = 0$ if $i > \deg p(z)$.

The stacking vectors $\hat{a}_\alpha$, $\hat{b}_\beta$, $\hat{c}_\alpha$ and $\hat{d}_\beta$ satisfy the following linear system of homogeneous equations:

$$V_{\alpha,\beta}(L_{s^+}, R_{s^+}, s^+) \begin{bmatrix} \hat{a}_\alpha & \hat{c}_\alpha \\ \hat{b}_\beta & \hat{d}_\beta \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

or

$$V_{\alpha-1,\beta-1}(L_{s^+}, R_{s^+}, s^+) \begin{bmatrix} \hat{a}'_\alpha & \hat{c}'_\alpha \\ \hat{b}'_\beta & \hat{d}'_\beta \end{bmatrix} = \begin{bmatrix} -v_\alpha(L_{s^+}, s^+) & -v_\beta(R_{s^+}, s^+) \end{bmatrix}. \tag{84}$$

Note that $V_{\alpha-1,\beta-1}(L_{s^+}, R_{s^+}, s^+)$ is a square matrix. To enhance the accuracy of the computed approximations for $\hat{a}'_\alpha$, $\hat{b}'_\beta$, $\hat{c}'_\alpha$ and $\hat{d}'_\beta$ we will use iterative refinement based on an inversion formula for this matrix.

The following theorem provides us with the parameters for an inversion formula for coupled Vandermonde matrices.

THEOREM 66. *Let $V_{\alpha-1,\beta-1}(L_{s^+}, R_{s^+}, s^+)$ be nonsingular and*

$$B(z) = \begin{bmatrix} a(z) & c(z) \\ b(z) & d(z) \end{bmatrix}$$

as defined by (84). *Then the* 3×2 *system of linear equations*

$$\begin{cases} B(s_i^+)h_i = \begin{bmatrix} a(s_i^+) & c(s_i^+) \\ b(s_i^+) & d(s_i^+) \end{bmatrix} \begin{bmatrix} h_{i,1} \\ h_{i,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} L_{s^+,i} & R_{s^+,i} \end{bmatrix} B'(s_i^+)h_i = 1 \end{cases}$$

has a unique solution $h_i = \begin{bmatrix} h_{i,1} \\ h_{i,2} \end{bmatrix} \in \mathbb{C}^{2\times 1}$ *for every* $i \in \{1, \ldots, \alpha+\beta\}$.

PROOF. See Heinig [**136**, Proposition 4.2]. □

THEOREM 67. *Let* $V := V_{\alpha-1,\beta-1}(L_{s^+}, R_{s^+}, s^+)$ *be nonsingular and*

$$B(z) = \begin{bmatrix} a(z) & c(z) \\ b(z) & d(z) \end{bmatrix}$$

as defined by (84). *Then the inverse of* V *is given by*

$$V^{-1} = W\, V_{\alpha-1,\beta-1}(H_1, H_2, s^+)^T$$

with

$$W := \begin{bmatrix} A & C \\ B & D \end{bmatrix} \in \mathbb{C}^{(\alpha+\beta)\times(\alpha+\beta)}$$

where

$$A := \begin{bmatrix} a_1 & \cdots & a_\alpha \\ \vdots & \iddots & \\ a_\alpha & & \end{bmatrix} \in \mathbb{C}^{\alpha\times\alpha}, \qquad B := \begin{bmatrix} b_1 & \cdots & \cdots & b_\alpha \\ \vdots & \iddots & & \\ b_\beta & & & \end{bmatrix} \in \mathbb{C}^{\beta\times\alpha},$$

$$C := \begin{bmatrix} c_1 & \cdots & c_\beta \\ \vdots & \iddots & \\ \vdots & & \\ c_\alpha & & \end{bmatrix} \in \mathbb{C}^{\alpha\times\beta}, \qquad D := \begin{bmatrix} d_1 & \cdots & d_\beta \\ \vdots & \iddots & \\ d_\beta & & \end{bmatrix} \in \mathbb{C}^{\beta\times\beta}$$

are upper triangular Hankel matrices. The complex column vectors $H_1, H_2 \in \mathbb{C}^{\alpha+\beta}$ *are defined as*

$$H_1 = \begin{bmatrix} h_{1,1} \\ \vdots \\ h_{\alpha+\beta,1} \end{bmatrix} \qquad \textit{and} \qquad H_2 = \begin{bmatrix} h_{1,2} \\ \vdots \\ h_{\alpha+\beta,2} \end{bmatrix}$$

where $h_{i,1}$ *and* $h_{i,2}$, $i = 1, \ldots, \alpha+\beta$, *are as in the previous theorem.*

PROOF. See Heinig [**136**, Theorem 4.1]. □

In the setting that we will consider in the next chapter, the interpolation points s^+ together with the difficult points will form a set of N_s points that are equally spaced on the unit circle. The computation of the inversion parameters H_1 and H_2 can therefore be done in $\mathcal{O}(N_s \log N_s)$ flops. Each iterative refinement step based on the inversion formula of Theorem 67 also involves $\mathcal{O}(N_s \log N_s)$ flops. Note that iterative refinement can be applied at one or more intermediate levels.

3.3. Downdating. Finite precision arithmetic can lead to a situation where

$$f_k^T B(s_k) \not\approx \begin{bmatrix} 0 & 0 \end{bmatrix}$$

for one or more interpolation points s_k. As the matrix $B(s_k)$ is singular, there exists a vector $v \in \mathbb{C}^2$ such that

$$B(s_k)v = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Define

$$B(t) =: \begin{bmatrix} B_L(t) & B_R(t) \end{bmatrix}, \qquad v =: \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$

and let $\alpha_L := \tau\text{-deg}B_L(t)$ and $\alpha_R := \tau\text{-deg}B_R(t)$. If $\alpha_L \geq \alpha_R$ and $v_L \neq 0$, then we replace $B_L(t)$ by

$$B_L(t) \leftarrow B(t)v/(t - s_k).$$

If, on the other hand, $\alpha_L < \alpha_R$ and $v_R \neq 0$, then we replace $B_R(t)$ by

$$B_R(t) \leftarrow B(t)v/(t - s_k).$$

If $v_L = 0$, then $B_R(t)$ is divisible by $t - s_k$. Similarly, if $v_R = 0$, then $B_L(t)$ is divisible by $t - s_k$. These considerations lead to the following algorithm.

function $[B(t), s_{\text{bad}}] \leftarrow$ DOWNDATING$(B(t), s, L_s, R_s, N_s)$
-- N_s: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(t) \in \mathbb{C}[t]^{2\times 2}$
-- on input: a basis matrix corresponding to the given interpolation data
-- on output: the corresponding downdated basis matrix
-- s_{bad}: a complex column vector containing the interpolation points that have been downdated

$s_{\text{bad}} \leftarrow \emptyset$
for $k = 1(1)N_s$
  if $\| \begin{bmatrix} L_s(k) & R_s(k) \end{bmatrix} \| > \eta$ **then**
    Choose $v \in \mathbb{C}^2$ such that $B(s(k))v = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and $\|v\| = 1$
    -- Let $B(t) =: \begin{bmatrix} B_L(t) & B_R(t) \end{bmatrix}$ and $v =: \begin{bmatrix} v_L & v_R \end{bmatrix}^T$
    $\alpha_L \leftarrow \tau\text{-deg}B_L(t)$
    $\alpha_R \leftarrow \tau\text{-deg}B_R(t)$
    if $\alpha_L \geq \alpha_R$ **then**
      if $v_L \neq 0$ **then**
        $B_L(t) \leftarrow B(t)v/(t - s(k))$
      else
        $B_R(t) \leftarrow B_R(t)/(t - s(k))$
      end if
    else
      if $v_R \neq 0$ **then**
        $B_R(t) \leftarrow B(t)v/(t - s(k))$
      else
        $B_L(t) \leftarrow B_L(t)/(t - s(k))$
      end if

end if
$s_{\mathrm{bad}} \leftarrow s_{\mathrm{bad}} \oplus s(k)$
end if
end for

We are now ready to formulate the stabilized version of our algorithm.

recursive function $[\, B(t), s_{\mathrm{bad}} \,] \leftarrow \textsc{RecRatInt}(s, L_s, R_s, N_s, \tau)$
-- $\tau \in \mathbb{Z}$
-- $N_s = 2^{p+1}$ for some $p \in \mathbb{N}$: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(t) \in \mathbb{C}[t]^{2\times 2}$: a τ-reduced basis matrix corresponding to the given interpolation data
-- s_{bad}: a complex column vector containing the difficult interpolation points
if $N_s > 2^{\mathrm{limit}}$ **then**
$[\, s_1, L_{s_1}, R_{s_1}, s_2, L_{s_2}, R_{s_2} \,] \leftarrow \textsc{Split}(s, L_s, R_s)$
$[\, B_1(t), s_{\mathrm{bad},1} \,] \leftarrow \textsc{RecRatInt}(s_1, L_{s_1}, R_{s_1}, N_s/2, \tau)$
for $k = 1(1)N_s/2$
$[\, \tilde{L}_{s_2}(k) \;\; \tilde{R}_{s_2}(k) \,] \leftarrow [\, L_{s_2}(k) \;\; R_{s_2}(k) \,] \cdot B_1(s_2(k))$
end for
$\tilde{\tau} \leftarrow$ the difference between the left and right τ-degrees of $B_1(t)$
$[\, \tilde{B}_2(t), \tilde{s}_{\mathrm{bad},2} \,] \leftarrow \textsc{RecRatInt}(s_2, \tilde{L}_{s_2}, \tilde{R}_{s_2}, N_s/2, \tilde{\tau})$
$B(t) \leftarrow B_1(t) \cdot \tilde{B}_2(t)$
$s_{\mathrm{bad}} \leftarrow s_{\mathrm{bad},1} \oplus \tilde{s}_{\mathrm{bad},2}$
else
$[\, B(t), s_{\mathrm{bad}} \,] \leftarrow \textsc{RatInt}(s, L_s, R_s, N_s, \tau)$
end if
if $N_s = 2^{\mathrm{downdating}}$ **then**
$s^+ \leftarrow s \ominus s_{\mathrm{bad}}$
$[\, B(t), s_{\mathrm{bad},3} \,] \leftarrow \textsc{Downdating}(s^+, L_{s^+}, R_{s^+}, N_s)$
$s_{\mathrm{bad}} \leftarrow s_{\mathrm{bad}} \oplus s_{\mathrm{bad},3}$
end if
if $N_s = 2^{\mathrm{reflimit}}$ **then**
$s^+ \leftarrow s \ominus s_{\mathrm{bad}}$
$[\, B(t) \,] \leftarrow \textsc{ItRef}(B(t), s^+, L_{s^+}, R_{s^+}, N_s, N_{\mathrm{ref}})$
end if

function $[\, B(t) \,] \leftarrow \textsc{RatIntAll}(s, L_s, R_s, N_s, \tau)$
-- $\tau \in \mathbb{Z}$
-- $N_s = 2^{p+1}$ for some $p \in \mathbb{N}$: the number of interpolation conditions
-- $s \in \mathbb{C}^{N_s \times 1}$: the (mutually distinct) interpolation points
-- $L_s, R_s \in \mathbb{C}^{N_s \times 1}$: the initial left and right residual vectors
-- $B(t) \in \mathbb{C}[t]^{2\times 2}$: a τ-reduced basis matrix corresponding to the given interpolation data
$[\, B^+(t), s_{\mathrm{bad}} \,] \leftarrow \textsc{RecRatInt}(s, L_s, R_s, N_s, \tau)$

$N_{\text{bad}} \leftarrow \textsc{Size}(s_{\text{bad}})$
if $N_{\text{bad}} > 0$ **then**
  calculate L_{bad} and R_{bad}
  $\tau^- \leftarrow$ the difference between the left and right τ-degrees of $B^+(t)$
  $[\,B^-(t)\,] \leftarrow \textsc{RatInt}(s_{\text{bad}}, L_{\text{bad}}, R_{\text{bad}}, N_{\text{bad}}, \tau^-)$
  $B(t) \leftarrow B^+(t) \cdot B^-(t)$
end if
return

Acknowledgements

The results in this chapter were obtained in close collaboration with Marc Van Barel.

CHAPTER 9

Superfast Hankel and Toeplitz solvers

In a Toeplitz matrix the entries are the same along each diagonal whereas in a Hankel matrix the entries are the same along each antidiagonal. These matrices occur in many applications, for example in signal processing or Markov chains [**34, 213**]. They also play a central role in the theory of orthogonal polynomials and Padé approximation. We have already encountered Hankel matrices in the first part of this thesis. A Toeplitz system of linear equations can be transformed into a Hankel system of linear equations by reversing the order of the unknowns and in this sense Toeplitz and Hankel systems are equivalent. Gaussian elimination requires $\mathcal{O}(n^3)$ floating point operations to solve an $n \times n$ Toeplitz or Hankel system. However, Toeplitz and Hankel matrices depend on $2n-1$ instead of n^2 entries and hence one can try to exploit this structure to obtain algorithms that require less floating point operations than general purpose algorithms. An algorithm for solving Toeplitz or Hankel systems is called *fast* if its arithmetic complexity is $\mathcal{O}(n^2)$. *Superfast* algorithms require only $\mathcal{O}(n \log^2 n)$ flops.

Several fast algorithms exist. They compute a triangular factorization of the matrix or of its inverse in a recursive way. More precisely, in step $k \leq n$ a factorization of the $k \times k$ leading principal submatrix (section) is constructed based on a factorization of the $(k-1) \times (k-1)$ section. These algorithms only work if all these sections are nonsingular, i.e., if the matrix is strongly nonsingular. There exists algorithms that overcome this difficulty by jumping over exactly singular sections. These approaches result in a block triangular factorization of the matrix. Nearly singular sections are handled in the same way as nonsingular sections. Hence it is safe to apply these methods in exact arithmetic. However, in floating point arithmetic stability problems are likely to occur in case nearly singular sections are involved. We refer the reader to [**35, 170**] and the references cited therein.

Look-ahead algorithms try to overcome this instability by looking ahead from one well-conditioned section to the next one and by jumping over the ill-conditioned sections that lie in between. If there are only a few such ill-conditioned sections, then the jumps can be small and the algorithm remains fast. However, if all the sections of the matrix except the last one (the matrix itself) are singular, then look-ahead algorithms will require $\mathcal{O}(n^3)$ flops. Also, a look-ahead criterion and a condition estimator that are reliable as well as cheap are difficult to construct. One has to balance between a severe look-ahead criterion that can lead to big jumps and a less stringent criterion that can give less accurate results. For more details about look-ahead algorithms for Toeplitz systems we refer to [**57, 58, 84, 85, 89, 113, 123, 129, 132, 133, 150**]. Look-ahead algorithms for Hankel systems can be

found in [**39, 51, 86**]. For the block Toeplitz and the block Hankel case, the reader may consult [**277**]. In [**88**] a look-ahead Schur algorithm was designed for Hermitian block Toeplitz matrices. Several high performance algorithms for Toeplitz and block Toeplitz matrices are described in [**90**], including two look-ahead Schur algorithms for symmetric indefinite block Toeplitz matrices. In [**250**] a look-ahead block Schur algorithm for Toeplitz-like matrices was presented.

Good look-ahead strategies are difficult to design. Only recently a completely different approach has been considered. Gaussian elimination relies on (partial or complete) pivoting to enhance the numerical stability. Unfortunately, pivoting destroys the structure of a Hankel or a Toeplitz matrix. Other classes of matrices maintain their structure after pivoting. For example, a matrix $M = [m_{kl}]$ is called *Cauchy-like* if there exist certain numbers y_k and z_l such that the rank of the matrix $[(y_k - z_l)m_{kl}]$ is small compared to the order of M. Pivoting does not destroy the Cauchy-like structure and hence fast as well as numerically stable algorithms for solving Cauchy-like systems can be designed [**104, 105, 106, 134, 140, 170**]. Heinig [**134**] was one of the first who proposed to *transform* structured matrices from one class into another and to use pivoting strategies to enhance the numerical stability. For an overview of different transformation techniques and algorithms we refer the reader to [**90, 137, 138, 139**] and the references cited therein.

Pivoting for structured matrices has a good record of numerical performance in several applications. For example, what is known as the Leja ordering is the equivalent of partial pivoting for Vandermonde-related structures, see [**143, 247**]. Rational Leja ordering (i.e., for Cauchy matrices) is discussed in [**40**]. In the context of interpolation, pivoting is numerically shown to be a very succesful technique in [**109**] and [**169**]. Numerical experiments in [**107**] and [**169**] showed the practicality of this approach for shift-invariant structures (such as Toeplitz or Toeplitz plus Hankel matrices). For a review about pivoting on structured matrices, we refer to [**110**].

In [**135**] Heinig transformed a Toeplitz system into a paired Vandermonde system, which is then solved as a tangential Lagrange interpolation problem. When two steps of his algorithm are combined, one obtains a block-step algorithm [**137**]. In this chapter we will present a fast Hankel solver, a fast block Hankel solver and a superfast Hankel solver. Our approach is related to Heinig's in the sense that they both proceed by first transforming the system and they both involve the solution of interpolation problems. We will also present a superfast Toeplitz solver that is based on a different transformation approach. In Section 1 we will show how an $n \times n$ Hankel system $Hx = b$ can be transformed into a Loewner system $Lx' = b'$. An explicit formula for L^{-1} enables us to calculate x' as $L^{-1}b'$. The solution x' is then transformed back into x. These steps require only $\mathcal{O}(n \log n)$ flops. The inversion formula for Loewner matrices involves certain parameters that can be computed by solving two linearized rational interpolation problems. By using the $\mathcal{O}(n^2)$ algorithm RatInt that we have presented in the previous chapter, we obtain a fast Hankel solver. Note that RatInt incorporates pivoting. The corresponding algorithm for block Hankel matrices will be presented in Section 2.

The first superfast algorithms were designed by Sugiyama et al. [**262**], Bitmead and Anderson [**37**], Brent, Gustavson and Yun [**43**] and also Morf [**214**]. More recent algorithms can be found in [**4, 5, 6, 49, 64, 112, 194, 221, 261**]. For the block case we refer to [**102, 196**]. The main disadvantage of these algorithms is that they cannot handle nearly singular leading principal submatrices. To overcome this problem, Gutknecht [**129**] and Gutknecht and Hochbruck [**132, 133**] developed an algorithm that combines the look-ahead idea with divide and conquer techniques. Because in most practical problems the look-ahead step will be small compared to the order of the system that is to be solved, the algorithm is *generically* superfast. In the previous chapter we have used a divide and conquer approach to obtain a superfast algorithm for rational interpolation. By using this algorithm instead of RatInt we obtain a stabilized generically superfast solver for indefinite Hankel systems, although only in case the order n is a power of 2. This will be discussed in Section 3. In Section 4 we will present a stabilized generically superfast solver for indefinite Toeplitz systems. Our approach relies on an inversion formula for Toeplitz matrices and is valid for arbitrary n. This algorithm is faster than our superfast Hankel solver. The inversion formula contains certain parameters that are computed by solving linearized rational interpolation problems. Numerical examples will illustrate the effectiveness of our approach.

The algorithms that we will present in this chapter have been implemented in Fortran 90 as well as in Matlab. Our software is available at

`http://www.cs.kuleuven.ac.be/~marc/hankel/`

Part of this chapter is contained in our papers [**187**], [**189**], [**278**] and [**279**].

1. A fast Hankel solver

Let n be a positive integer, let $H = H_n := [h_{k+l}]_{k,l=0}^{n-1}$ be a nonsingular $n \times n$ complex Hankel matrix and let $b \in \mathbb{C}^n$. We consider the problem of computing $x := H^{-1}b$.

1.1. Transformation into a Loewner system. Let $y_1, \ldots, y_n, z_1, \ldots, z_n$ be $2n$ mutually distinct complex numbers and define $\mathbf{y} := (y_1, \ldots, y_n)$ and $\mathbf{z} := (z_1, \ldots, z_n)$. Let $\mathcal{L}(\mathbf{y}, \mathbf{z})$ be the class of matrices

$$\mathcal{L}(\mathbf{y}, \mathbf{z}) := \left\{ \left[\frac{c_k - d_l}{y_k - z_l} \right]_{k,l=1}^{n} \mid c_1, \ldots, c_n, d_1, \ldots, d_n \in \mathbb{C} \right\}.$$

The elements of $\mathcal{L}(\mathbf{y}, \mathbf{z})$ are called *Loewner matrices*. They bear the name of Karl Loewner who studied them in the context of rational interpolation and monotone matrix functions [**202**].

The set $\mathcal{L}(\mathbf{y}, \mathbf{z})$ is a linear space over $\mathbb{C}$ and a subspace of the linear space of all the $n \times n$ complex matrices. Since addition of a constant to all the $2n$ parameters c_k, d_l leads to the same Loewner matrix, its dimension is $2n-1$. The set of all the $n \times n$ complex Hankel matrices also forms a linear subspace of dimension $2n-1$. Hankel and Loewner matrices are even more closely related. According to Fiedler [**79**] every Hankel matrix can be transformed into a Loewner matrix and vice versa. We will now formulate this theorem and discuss our $\mathcal{O}(n \log n)$ implementation of the transformation.

First we have to deal with some preliminaries concerning Vandermonde matrices. Let $t_1, \ldots, t_n$ be n complex numbers and define $\mathbf{t} := (t_1, \ldots, t_n)$. The Vandermonde matrix with nodes $t_1, \ldots, t_n$ is given by

$$V(\mathbf{t}) = V(t_1, \ldots, t_n) := \begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_n & \cdots & t_n^{n-1} \end{bmatrix}.$$

Let $f_{\mathbf{t}}(z)$ be the monic polynomial of degree n that has zeros $t_1, \ldots, t_n$,

$$f_{\mathbf{t}}(z) := (z - t_1) \cdots (z - t_n),$$

and define

$$f_{\mathbf{t},j}(z) := \prod_{k \neq j} (z - t_k), \qquad j = 1, \ldots, n.$$

Note that $f_{\mathbf{t},j}(z)$ is a monic polynomial of degree $n-1$ for $j = 1, \ldots, n$. Define the $n \times n$ matrix $W(\mathbf{t})$ by the equation

$$\begin{bmatrix} f_{\mathbf{t},1}(z) \\ f_{\mathbf{t},2}(z) \\ \vdots \\ f_{\mathbf{t},n}(z) \end{bmatrix} = W(\mathbf{t}) \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{n-1} \end{bmatrix}. \tag{85}$$

This means that the jth row of $W(\mathbf{t})$ contains the coefficients of $f_{\mathbf{t},j}(z)$ when written in terms of the standard monomial basis $\{1, z, \ldots, z^{n-1}\}$. Then

$$W(\mathbf{t})\,[V(\mathbf{t})]^T = \operatorname{diag}\big(\, f_{\mathbf{t},1}(t_1), \ldots, f_{\mathbf{t},n}(t_n) \,\big) =: D(\mathbf{t}). \tag{86}$$

The Vandermonde matrix $V(\mathbf{t})$ is nonsingular if and only if its nodes $t_1, \ldots, t_n$ are mutually distinct. In that case (86) implies that $W(\mathbf{t})$ is nonsingular and also that

$$[V(\mathbf{t})]^{-1} = [W(\mathbf{t})]^T\,[D(\mathbf{t})]^{-1}. \tag{87}$$

For an early reference to this formula see, for example, the book by Kowalewski [**181**].

Let $V(\mathbf{y}, \mathbf{z})$ be the $2n \times 2n$ Vandermonde matrix with nodes $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ and similarly for $W(\mathbf{y}, \mathbf{z})$.

THEOREM 68. *The matrix $L := W(\mathbf{y})\,H\,[W(\mathbf{z})]^T$ is a Loewner matrix in $\mathcal{L}(\mathbf{y}, \mathbf{z})$ whose parameters $c_1, \ldots, c_n, d_1, \ldots, d_n$ are given by (up to an arbitrary additive constant $\xi \in \mathbb{C}$)*

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = W(\mathbf{y}, \mathbf{z}) \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{2n-2} \\ \xi \end{bmatrix}.$$

PROOF. See Fiedler [**79**, Theorem 12]. □

Note that L is nonsingular.

A judicious choice of the points $\mathbf{y}$ and $\mathbf{z}$ enables us to transform the Hankel system $Hx = b$ into the Loewner system $Lx' = b'$ in $\mathcal{O}(n \log n)$ flops. Let $\omega := \exp(2\pi\, i/n)$ and suppose from now on that $y_k = \omega^{k-1}$ for $k = 1, \ldots, n$. That is, let $\mathbf{y} = (1, \omega, \ldots, \omega^{n-1})$. Let $\zeta := \exp(\pi\, i/n)$ and suppose from now on that $z_k = \zeta y_k$ for $k = 1, \ldots, n$. That is, let $\mathbf{z} = (\zeta, \zeta\omega, \ldots, \zeta\omega^{n-1})$. Note that

$$\mathbf{y} = (1, \zeta^2, \ldots, \zeta^{2n-2}) \qquad \text{and} \qquad \mathbf{z} = (\zeta, \zeta^3, \ldots, \zeta^{2n-1}).$$

Let Ω_n be the $n \times n$ Fourier matrix,

$$\Omega_n := \frac{1}{\sqrt{n}} V(1, \omega, \ldots, \omega^{n-1}). \tag{88}$$

Matrix-vector products involving Ω_n (Ω_n^H) amount to a(n) (inverse) discrete Fourier transform (DFT) and can be evaluated via the celebrated (inverse) fast Fourier transform (FFT) in $\mathcal{O}(n \log n)$ flops. Finally, let $D_{n,\omega}$ and $D_{n,\zeta}$ be the $n \times n$ diagonal matrices $D_{n,\omega} := \mathrm{diag}(1, \omega, \ldots, \omega^{n-1})$ and $D_{n,\zeta} := \mathrm{diag}(1, \zeta, \ldots, \zeta^{n-1})$.

THEOREM 69. *The solution to the Hankel system $Hx = b$ is given by*

$$x = \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}}\, x'$$

where $x' := L^{-1}b'$ with $b' := \sqrt{n}\, \overline{D_{n,\omega}\Omega_n}\, b$.

PROOF. Theorem 68 implies that $x = H^{-1}b$ is given by $x = [W(\mathbf{z})]^T x'$ where $x' := L^{-1}b'$ with $b' := W(\mathbf{y})b$. Equation (86) and (88) and the fact that Ω_n is unitary imply that

$$W(\mathbf{y}) = W(1, \omega, \ldots, \omega^{n-1}) = \frac{1}{\sqrt{n}} D(1, \omega, \ldots, \omega^{n-1})\, \overline{\Omega_n}. \tag{89}$$

An easy calculation shows that $D(1, \omega, \ldots, \omega^{n-1}) = n\, \overline{D_{n,\omega}}$ and thus $b' = W(\mathbf{y})b = \sqrt{n}\, \overline{D_{n,\omega}\Omega_n}\, b$.

Equation (87) tells us that

$$[W(\mathbf{z})]^T = [V(\mathbf{z})]^{-1} D(\mathbf{z}).$$

Since $V(\mathbf{z}) = V(1, \omega, \ldots, \omega^{n-1}) D_{n,\zeta}$ and $D(\mathbf{z}) = n\zeta^{n-1} \overline{D_{n,\omega}}$, it follows that

$$\begin{aligned} x = [W(\mathbf{z})]^T x' &= n\zeta^{n-1} D_{n,\zeta}^{-1} [V(1, \omega, \ldots, \omega^{n-1})]^{-1} \overline{D_{n,\omega}}\, x' \\ &= \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}}\, x'. \end{aligned}$$

This proves the theorem. □

Let P be the permutation matrix defined by

$$P \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} 1 \\ \zeta \\ \vdots \\ \zeta^{2n-1} \end{bmatrix},$$

let Ω_{2n} be the $2n \times 2n$ Fourier matrix,

$$\Omega_{2n} := \frac{1}{\sqrt{2n}} V(1, \zeta, \ldots, \zeta^{2n-1}),$$

and let $D_{2n,\zeta} := \mathrm{diag}(1, \zeta, \ldots, \zeta^{2n-1})$.

THEOREM 70. *The parameters $c_1, \ldots, c_n, d_1, \ldots, d_n$ of the Loewner matrix L are given by (up to an arbitrary additive constant $\xi \in \mathbb{C}$)*

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = \sqrt{2n}\, P^T \overline{D_{2n,\zeta}\, \Omega_{2n}} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{2n-2} \\ \xi \end{bmatrix}.$$

PROOF. Equation (85) and (89) imply that

$$P\, W(\mathbf{y}, \mathbf{z}) = W(1, \zeta, \ldots, \zeta^{2n-1}) = \frac{1}{\sqrt{2n}} D(1, \zeta, \ldots, \zeta^{2n-1})\, \overline{\Omega_{2n}}.$$

An easy calculation shows that $D(1, \zeta, \ldots, \zeta^{2n-1}) = 2n\, \overline{D_{2n,\zeta}}$ and thus

$$W(\mathbf{y}, \mathbf{z}) = \sqrt{2n}\, P^T \overline{D_{2n,\zeta}\, \Omega_{2n}}.$$

The result then follows immediately from Theorem 68. □

NOTE. In the proof of Theorem 69 we have shown that

$$W(\mathbf{y}) = \sqrt{n}\, \overline{D_{n,\omega} \Omega_n} \qquad \text{and} \qquad [W(\mathbf{z})]^T = \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}}.$$

These formulae imply that $W(\mathbf{y})/\sqrt{n}$ and $[W(\mathbf{z})]^T/\sqrt{n}$ are unitary. Therefore $\|L\|_2 = n\|H\|_2$ and $\|L^{-1}\|_2 = n^{-1}\|H^{-1}\|_2$ and thus $\kappa_2(L) = \kappa_2(H)$. In other words, the spectral condition number is left unchanged.

1.2. An inversion formula for Loewner matrices.

THEOREM 71. *Let $p_k, u_k, \tilde{p}_k$ and $\tilde{u}_k$ for $k = 1, \ldots, n$ be defined by the equations*

$$\begin{bmatrix} p_1 & \cdots & p_n \end{bmatrix} L = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$$
$$\begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} L = \begin{bmatrix} d_1 & \cdots & d_n \end{bmatrix}$$
$$L \begin{bmatrix} \tilde{p}_1 \\ \vdots \\ \tilde{p}_n \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$
$$L \begin{bmatrix} \tilde{u}_1 \\ \vdots \\ \tilde{u}_n \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

Then the inverse of L is given by

$$L^{-1} = \left[\frac{\tilde{u}_k\, p_l - \tilde{p}_k\, u_l}{y_l - z_k} \right]_{k,l=1}^{n}. \tag{90}$$

PROOF. See Van Barel and Vavřín [**280**, Theorem 2.2]. □

A Loewner matrix is a special kind of Cauchy-like matrix. The inversion formula given in the previous theorem can therefore be seen as a special case of the formula that appears in the book written by Heinig and Rost [**140**, p. 161].

The parameters that appear in the inversion formula (90) can be computed by solving the following two linearized rational interpolation problems.

THEOREM 72. *Define*

$$\begin{aligned} u(z) &:= f_{\mathbf{y}}(z) - \sum_{k=1}^{n} u_k \, f_{\mathbf{y},k}(z) \\ v(z) &:= -\sum_{k=1}^{n} c_k \, u_k \, f_{\mathbf{y},k}(z) \\ \tilde{u}(z) &:= f_{\mathbf{z}}(z) - \sum_{k=1}^{n} \tilde{u}_k \, f_{\mathbf{z},k}(z) \\ \tilde{v}(z) &:= -\sum_{k=1}^{n} d_k \, \tilde{u}_k \, f_{\mathbf{z},k}(z). \end{aligned}$$

Then the polynomial pair $(v(z), u(z))$ *is the only pair such that*

- $u(z) \in \mathbb{C}[z]$ *and* $\deg u(z) = n$, $v(z) \in \mathbb{C}[z]$ *and* $\deg v(z) < n$, $u(z)$ *is monic*
- $v(y_k) = c_k \, u(y_k)$ *and* $v(z_k) = d_k \, u(z_k)$ *for* $k = 1, \ldots, n$.

The polynomial pair $(\tilde{v}(z), \tilde{u}(z))$ *satisfies exactly the same properties and thus* $v(z) \equiv \tilde{v}(z)$ *and* $u(z) \equiv \tilde{u}(z)$.

PROOF. See Van Barel and Vavřín [**280**, Theorem 3.1]. Compare also with Vavřín [**284**, Theorem 2.1]. □

THEOREM 73. *Define*

$$\begin{aligned} p(z) &:= \sum_{k=1}^{n} p_k \, f_{\mathbf{y},k}(z) \\ q(z) &:= f_{\mathbf{y}}(z) + \sum_{k=1}^{n} c_k \, p_k \, f_{\mathbf{y},k}(z) \\ \tilde{p}(z) &:= \sum_{k=1}^{n} \tilde{p}_k \, f_{\mathbf{z},k}(z) \\ \tilde{q}(z) &:= f_{\mathbf{z}}(z) + \sum_{k=1}^{n} d_k \, \tilde{p}_k \, f_{\mathbf{z},k}(z). \end{aligned}$$

Then the polynomial pair $(q(z), p(z))$ *is the only pair such that*

- $q(z) \in \mathbb{C}[z]$ *and* $\deg q(z) = n$, $p(z) \in \mathbb{C}[z]$ *and* $\deg p(z) < n$, $q(z)$ *is monic*
- $q(y_k) = c_k \, p(y_k)$ *and* $q(z_k) = d_k \, p(z_k)$ *for* $k = 1, \ldots, n$.

The polynomial pair $(\tilde{q}(z), \tilde{p}(z))$ *satisfies exactly the same properties and thus* $q(z) \equiv \tilde{q}(z)$ *and* $p(z) \equiv \tilde{p}(z)$.

PROOF. See Van Barel and Vavřín [**280**, Theorem 3.2]. □

Note that if $p(y_k)$ and $p(z_k)$, $k = 1, \ldots, n$, are different from zero, then the rational function $q(z)/p(z)$ satisfies the proper rational interpolation conditions

$$\frac{q(y_k)}{p(y_k)} = c_k \quad \text{and} \quad \frac{q(z_k)}{p(z_k)} = d_k, \qquad k = 1, \ldots, n.$$

Similarly, if $u(y_k)$ and $u(z_k)$, $k = 1, \ldots, n$, are different from zero, then the rational function $v(z)/u(z)$ satisfies the proper rational interpolation conditions

$$\frac{v(y_k)}{u(y_k)} = c_k \quad \text{and} \quad \frac{v(z_k)}{u(z_k)} = d_k, \qquad k = 1, \ldots, n.$$

The rational functions $q(z)/p(z)$ and $v(z)/u(z)$ are different because their degree structure is different.

An easy calculation reveals the following connections with the parameters that appear in the inversion formula (90):

$$\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = -\frac{1}{n} D_{n,\omega} \begin{bmatrix} u(y_1) \\ \vdots \\ u(y_n) \end{bmatrix}, \quad \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} = \frac{1}{n} D_{n,\omega} \begin{bmatrix} p(y_1) \\ \vdots \\ p(y_n) \end{bmatrix},$$

$$\begin{bmatrix} \tilde{u}_1 \\ \vdots \\ \tilde{u}_n \end{bmatrix} = -\frac{1}{n\overline{\zeta^{n-1}}} D_{n,\omega} \begin{bmatrix} u(z_1) \\ \vdots \\ u(z_n) \end{bmatrix}, \quad \begin{bmatrix} \tilde{p}_1 \\ \vdots \\ \tilde{p}_n \end{bmatrix} = \frac{1}{n\overline{\zeta^{n-1}}} D_{n,\omega} \begin{bmatrix} p(z_1) \\ \vdots \\ p(z_n) \end{bmatrix}.$$

As the points $y_1, \ldots, y_n, z_1, \ldots, z_n$ are (up to a permutation) equal to the $2n$th roots of unity $1, \zeta, \ldots, \zeta^{2n-1}$ we can use the FFT to evaluate the polynomials $u(z)$ and $p(z)$ at these points.

Once these inversion parameters have been computed, the matrix-vector product $x' = L^{-1}b'$ can be calculated in $\mathcal{O}(n \log n)$ flops. Indeed, L^{-1} can be written as

(91)
$$L^{-1} = \operatorname{diag}(\tilde{p}_1, \ldots, \tilde{p}_n)\, C\, \operatorname{diag}(u_1, \ldots, u_n) - \operatorname{diag}(\tilde{u}_1, \ldots, \tilde{u}_n)\, C\, \operatorname{diag}(p_1, \ldots, p_n)$$

if C is given by the Cauchy matrix

(92)
$$C := \left[\frac{1}{z_k - y_l} \right]_{k,l=1}^{n}.$$

By applying Proposition 3.2 in [**80**] to the special case of the roots of unity, one obtains that the Cauchy matrix C can be factorized as

$$C = -\frac{n}{2}\, \Omega_n\, D_{n,\zeta}\, \Omega_n^H\, \overline{D_{n,\omega}}.$$

This implies that the product of C with a vector in $\mathbb{C}^n$ can be evaluated in $\mathcal{O}(n \log n)$ flops and thus, because of (91), the same holds for the product $L^{-1}b'$.

1.3. The algorithm RATINT. Let $s_k := y_k$ and $s_{n+k} := z_k$ for $k = 1, \ldots, n$. Define the column vectors $f_1, \ldots, f_{2n} \in \mathbb{C}^{2\times 1}$ as

$$f_k := \begin{bmatrix} 1 \\ -c_k \end{bmatrix}, \quad f_{n+k} := \begin{bmatrix} 1 \\ -d_k \end{bmatrix}, \qquad k = 1, \ldots, n.$$

The linearized rational interpolation problems that appear in Theorems 72 and 73 can be formulated as

$$f_k^T B(s_k) = [0 \;\; 0], \qquad k = 1, \ldots, 2n, \tag{93}$$

where

$$B(z) = \begin{bmatrix} n_\ell(z) & n_r(z) \\ d_\ell(z) & d_r(z) \end{bmatrix} \in \mathbb{C}[z]^{2\times 2}$$

with $\deg n_\ell(z) = n$, $\deg d_\ell(z) \le n-1$, $\deg n_r(z) \le n-1$ and $\deg d_r(z) = n$, and $n_\ell(z)$ as well as $d_r(z)$ monic, i.e., where $B(z)$ is a monic 2×2 matrix polynomial of degree n. This corresponds to the interpolation problem that we have considered in Section 1 of Chapter 8. Theorems 72 and 73 assert that this problem has a unique solution given by

$$B^\star(z) := \begin{bmatrix} q(z) & v(z) \\ p(z) & u(z) \end{bmatrix}.$$

Algorithm RATINT calculates a 2×2 matrix polynomial $B(z)$ of degree n that satisfies (93) and whose highest degree coefficient $A \in \mathbb{C}^{2\times 2}$ is nonsingular. This implies that $B(z) \equiv B^\star(z)A$. In Chapter 8 we have shown that $\det A = 1$.

Note that only the second row of $B^\star(z)$ is needed to compute the inversion parameters. If we compute the inversion parameters from the polynomials that appear in the second row of $B(z)$, then we still obtain the correct value of L^{-1} because $\det A = 1$. Indeed, suppose that

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and let $p'(z) := ap(z) + cu(z)$ and $u'(z) := bp(z) + du(z)$. Define

$$\begin{bmatrix} u'_1 \\ \vdots \\ u'_n \end{bmatrix} = -\frac{1}{n} D_{n,\omega} \begin{bmatrix} u'(y_1) \\ \vdots \\ u'(y_n) \end{bmatrix}, \quad \begin{bmatrix} p'_1 \\ \vdots \\ p'_n \end{bmatrix} = \frac{1}{n} D_{n,\omega} \begin{bmatrix} p'(y_1) \\ \vdots \\ p'(y_n) \end{bmatrix},$$

$$\begin{bmatrix} \tilde{u}'_1 \\ \vdots \\ \tilde{u}'_n \end{bmatrix} = -\frac{1}{n} \overline{\zeta^{n-1}}\, D_{n,\omega} \begin{bmatrix} u'(z_1) \\ \vdots \\ u'(z_n) \end{bmatrix}, \quad \begin{bmatrix} \tilde{p}'_1 \\ \vdots \\ \tilde{p}'_n \end{bmatrix} = \frac{1}{n} \overline{\zeta^{n-1}}\, D_{n,\omega} \begin{bmatrix} p'(z_1) \\ \vdots \\ p'(z_n) \end{bmatrix}.$$

Then

$$u'_k = -bp_k + du_k, \qquad p'_k = ap_k - cu_k,$$
$$\tilde{u}'_k = -b\tilde{p}_k + d\tilde{u}_k, \qquad \tilde{p}'_k = a\tilde{p}_k - c\tilde{u}_k$$

for $k = 1, \ldots, n$. This implies that

$$\left[\frac{\tilde{u}'_k\, p'_l - \tilde{p}'_k\, u'_l}{y_l - z_k} \right]_{k,l=1}^{n} = \left[\frac{\tilde{u}_k\, p_l - \tilde{p}_k\, u_l}{y_l - z_k} \right]_{k,l=1}^{n}$$

because

$$\begin{aligned}
\tilde{u}'_k\, p'_l - \tilde{p}'_k\, u'_l &= \det \begin{bmatrix} \tilde{u}'_k & u'_l \\ \tilde{p}'_k & p'_l \end{bmatrix} \\
&= \det\Big(\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} \tilde{u}_k & u_l \\ \tilde{p}_k & p_l \end{bmatrix} \Big) \\
&= \det \begin{bmatrix} \tilde{u}_k & u_l \\ \tilde{p}_k & p_l \end{bmatrix} = \tilde{u}_k\, p_l - \tilde{p}_k\, u_l
\end{aligned}$$

for $k, l = 1, \ldots, n$ since

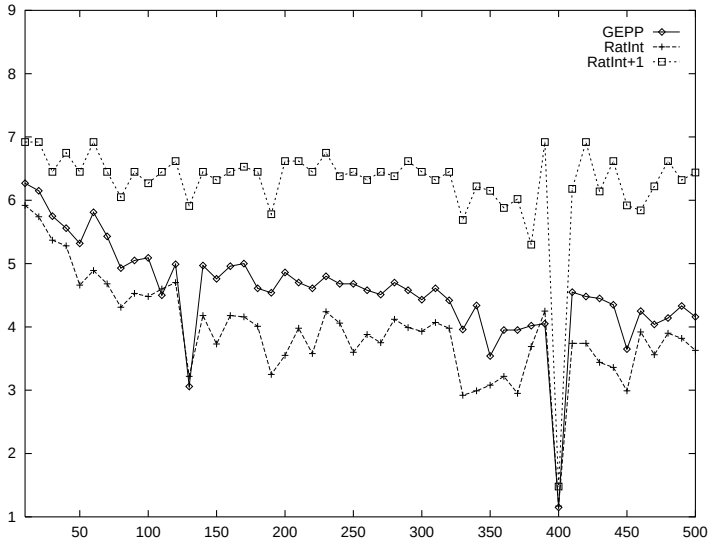
$$\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \det A & 0 \\ 0 & \det A \end{bmatrix} = I_2.$$

What is the arithmetic complexity of this approach? It takes $2n^2+\mathcal{O}(n)$ complex multiplications and $2n^2+\mathcal{O}(n)$ complex additions to compute the second row of $B(z)$. The updating of the residuals costs $4n^2 + \mathcal{O}(n)$ complex multiplications and $4n^2 + \mathcal{O}(n)$ complex additions. Thus the overall cost is $6n^2+\mathcal{O}(n)$ complex multiplications and $6n^2+\mathcal{O}(n)$ complex additions. Therefore, by solving the interpolation problems via RatInt, we obtain a fast Hankel solver, i.e., a Hankel solver that has arithmetic complexity $\mathcal{O}(n^2)$. Indeed, it takes only $\mathcal{O}(n \log n)$ flops to transform the Hankel system $Hx = b$ into the Loewner system $Lx' = b'$ and once the solutions to the two linearized rational interpolation problems are known, the parameters that appear in the inversion formula for L can be computed in $\mathcal{O}(n \log n)$ flops. A Fortran 90 as well as a Matlab implementation are available. In the Fortran version the FFTs are calculated via FFTPACK. In the next subsection we will discuss a few numerical experiments. They show the effectiveness of our approach.

1.4. Numerical examples. As a matrix-vector product involving a Hankel matrix amounts to a convolution of two vectors or, equivalently, the product of two polynomials, the residue $r := b - H\hat{x}$ can be calculated via FFT in $\mathcal{O}(n \log n)$ flops [**35**]. This implies that improving an approximation for x iteratively does not add to the $\mathcal{O}(n^2)$ complexity of our fast Hankel solver.

In the following examples we used the Fortran 90 version of our package. The calculations were done by an IBM SP2 with machine epsilon $\epsilon \approx 0.12\, 10^{-6}$ in single precision and $\epsilon \approx 0.22\, 10^{-15}$ in double precision.

Example 28. We consider single precision $n \times n$ real Hankel matrices H_n whose entries are random uniformly distributed in $[-1, 1]$ for $n = 10\,(10)\,500$. The right-hand sides $b_n \in \mathbb{R}^n$ are calculated in double precision such that $x_n := H_n^{-1} b_n = [1 \ \cdots \ 1]^T$. Figure 1 shows the results obtained by our algorithm (before and after one step of iterative improvement in which the residue was calculated in double precision) and the results obtained via Gaussian elimination with partial pivoting (GEPP) using the LAPACK routines CGETRF and CGETRS. We have plotted $-\log_{10}(\|\hat{x}_n - x_n\|_\infty / \|x_n\|_\infty)$. Also shown is $\log_{10} \kappa_\infty(H_n)$. The condition number was calculated via LAPACKs routine CGECON. Note that after one step of iterative improvement the relative accuracy of the solution is of the order of the machine precision except for very ill-conditioned Hankel matrices.



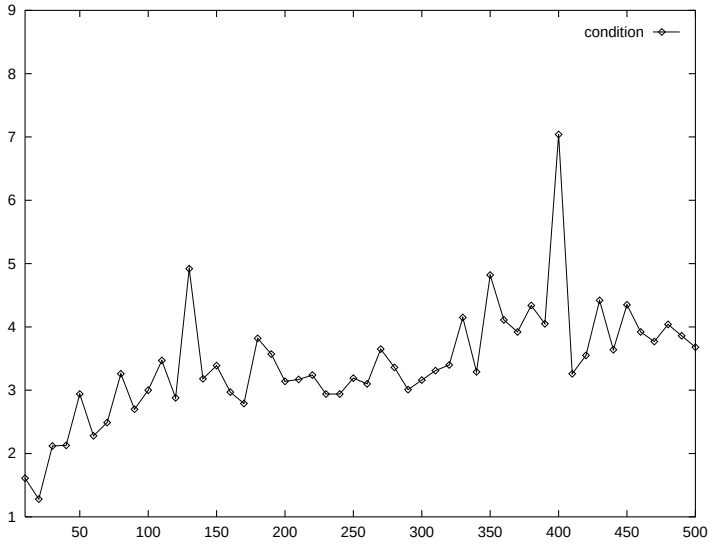


FIGURE 1. $-\log_{10} \dfrac{\|\hat{x}_n - x_n\|_\infty}{\|x_n\|_\infty}$ and $\log_{10} \kappa_\infty(H_n)$ versus n

EXAMPLE 29. In this test we consider the Hankel matrix

$$H := \begin{bmatrix} (\frac{1}{2})^{n-1} & \cdots & \cdots & \frac{1}{2} & \epsilon \\ (\frac{1}{2})^{n-2} & & & \iddots & \frac{1}{2} \\ \vdots & & \iddots & & \vdots \\ \frac{1}{2} & \iddots & & & (\frac{1}{2})^{n-2} \\ \epsilon & \frac{1}{2} & \cdots & (\frac{1}{2})^{n-2} & (\frac{1}{2})^{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

The corresponding Toeplitz matrix belongs to the class of the Kac-Murdock-Szegő matrices (with $\rho = \frac{1}{2}$) [**167**]. For $n = 3k+1$, $k = 1, 2, \ldots$, this matrix is singular if $\epsilon = 0$ and, consequently, ill-conditioned if ϵ is set to a small number. We take $n = 1000$ and let $\epsilon = 10^{-q}$ for $q = 0, 1, \ldots, 15$. We define the right-hand side $b := [\ b_1 \quad \cdots \quad b_n\ ]^T \in \mathbb{R}^n$ as

$$b_k := 2 + \epsilon - (\frac{1}{2})^{k-1} - (\frac{1}{2})^{n-k}, \qquad k = 1, \ldots, n,$$

because then $x := H^{-1}b$ is equal to $[\ 1 \quad \cdots \quad 1\ ]^T$ as one can easily verify. Figure 2 shows the results obtained by our algorithm (with up to three steps of iterative improvement). The calculations were done in double precision. We have plotted $-\log_{10}(\|\hat{x} - x\|_\infty / \|x\|_\infty)$ versus $q = -\log_{10}\epsilon$. Also shown is $\log_{10}\kappa_\infty(H)$. In Figure 3 we have plotted $-\log_{10}(\|b - H\hat{x}\|_\infty / \|b\|_\infty)$. Note that in case the condition number is not too large, the relative error for the residue after one (or more) step(s) of iterative improvement is of the order of the machine precision.

EXAMPLE 30. We consider the Hankel matrix

$$H_n := \begin{bmatrix} 1 & 2 & \cdots & n \\ 2 & & \iddots & 0 \\ \vdots & \iddots & \iddots & \vdots \\ n & 0 & \cdots & 0 \end{bmatrix}$$

for $n = 1000\,(100)\,10000$. One can easily verify that $\|H_n^{-1}\|_\infty = \mathcal{O}(1/n)$. As $\|H_n\|_\infty = n(n+1)/2$, this implies that $\kappa_\infty(H_n) = \mathcal{O}(n)$. We define the right-hand side $b_n = [\ b_1^{(n)} \quad \cdots \quad b_n^{(n)}\ ]^T$ as

$$b_k^{(n)} := \frac{1}{2}n(n+1) - \frac{1}{2}(k-1)k, \qquad k = 1, \ldots, n.$$

Then the solution to $H_n x_n = b_n$ is given by $x_n = [\ 1 \quad \cdots \quad 1\ ]^T$ as one can easily verify. Figure 4 shows the results obtained by our algorithm (with up to two steps of iterative improvement). The calculations were done in double precision.

2. A fast block Hankel solver

We will now show how the results presented in the previous section can be generalized to the block case. This will lead to a fast solver for block Hankel systems. Let n and p be positive integers. Consider a sequence $H_0, \ldots, H_{2n-2}$ in $\mathbb{C}^{p \times p}$ such that the $np \times np$ block Hankel matrix $H := [H_{k+l}]_{k,l=0}^{n-1}$ is nonsingular. Let $b \in \mathbb{C}^{np}$. We consider the problem of computing $x := H^{-1}b$.

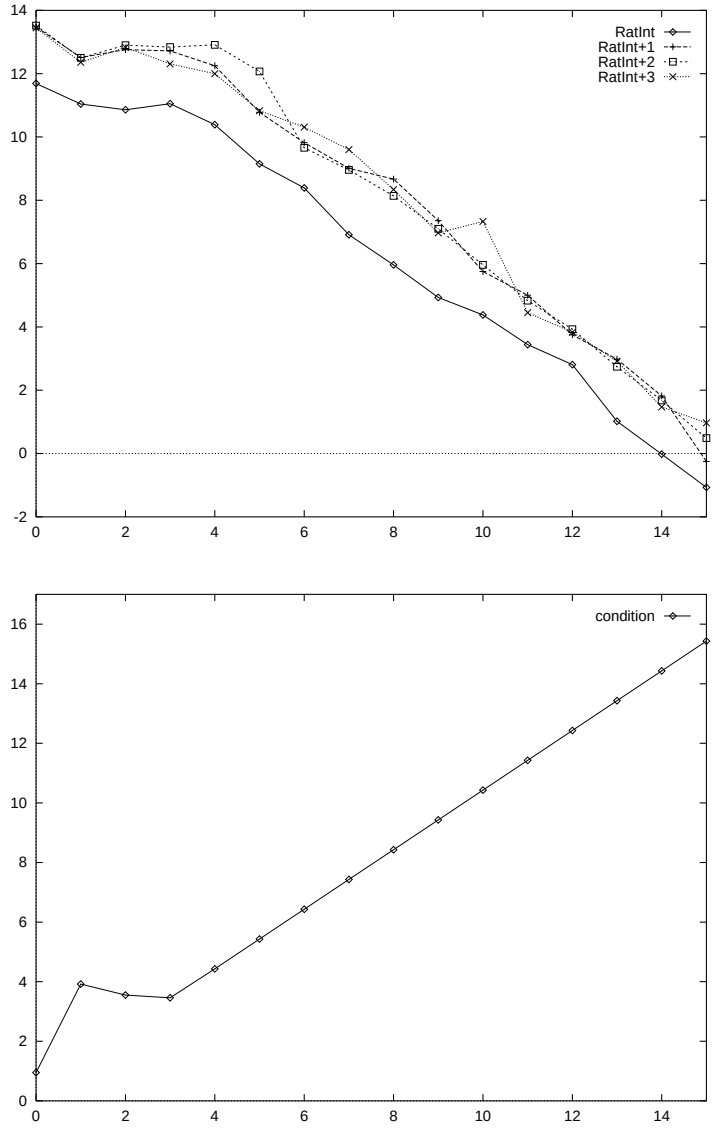


FIGURE 2. $-\log_{10} \dfrac{\|\hat{x} - x\|_\infty}{\|x\|_\infty}$ and $\log_{10} \kappa_\infty(H)$ versus $-\log_{10} \epsilon$

2.1. Transformation into a block Loewner system. Let $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ be $2n$ mutually distinct complex numbers and define $\mathbf{y} := (y_1, \ldots, y_n)$ and $\mathbf{z} := (z_1, \ldots, z_n)$. Let $\mathcal{L}_p(\mathbf{y}, \mathbf{z})$ be the class of $np \times np$ block matrices

$$\mathcal{L}_p(\mathbf{y}, \mathbf{z}) := \left\{ \left[\frac{C_k - D_l}{y_k - z_l} \right]_{k,l=1}^{n} \;\middle|\; C_1, \ldots, C_n, D_1, \ldots, D_n \in \mathbb{C}^{p \times p} \right\}.$$

The elements of $\mathcal{L}_p(\mathbf{y}, \mathbf{z})$ are called *block Loewner matrices*.

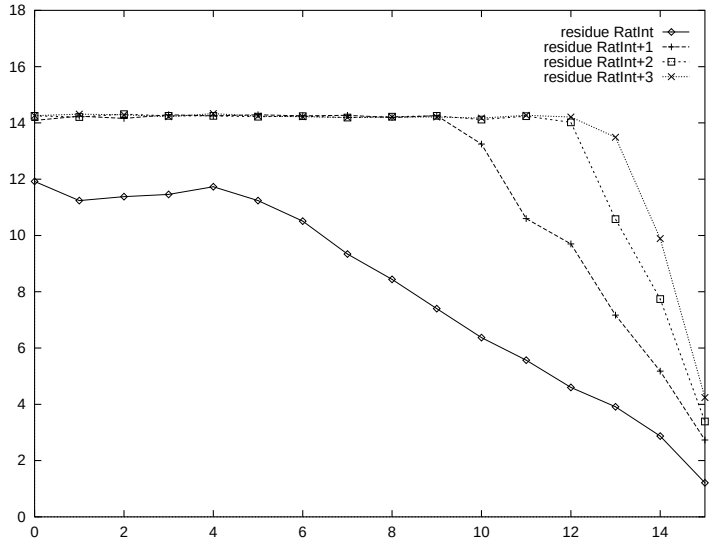


FIGURE 3. $-\log_{10} \dfrac{\|b - H\hat{x}\|_\infty}{\|b\|_\infty}$ versus $-\log_{10} \epsilon$

The set $\mathcal{L}_p(\mathbf{y}, \mathbf{z})$ is a $\mathbb{C}^{p\times p}$-module and a submodule of the $\mathbb{C}^{p\times p}$-module of all the complex $np \times np$ block matrices having $p \times p$ blocks. Since addition of a constant matrix to all the $2n$ $p \times p$ matrices C_k, D_l leads to the same block Loewner matrix, its dimension is $2n - 1$. The set of all the complex $np \times np$ block Hankel matrices having $p \times p$ blocks also forms a submodule of dimension $2n - 1$. Block Hankel and block Loewner matrices are even more closely related. We have already seen how in the scalar case ($p = 1$) every Hankel matrix can be transformed into a Loewner matrix and vice versa, cf. Theorem 68. We will now formulate this result for the block case and discuss our $\mathcal{O}(p^2 n \log n)$ implementation of the transformation.

Recall that the *Kronecker product* of two matrices $A = [a_{kl}]_{k,l=1}^{\alpha} \in \mathbb{C}^{\alpha\times\alpha}$ and $B \in \mathbb{C}^{\beta\times\beta}$ is defined as the block matrix

$$A \otimes B := [a_{kl}B]_{k,l=1}^{\alpha} \in \mathbb{C}^{\alpha\beta\times\alpha\beta}.$$

Let the matrices $W(\mathbf{y})$ and $W(\mathbf{z})$ be defined as in Subsection 1.1.

THEOREM 74. *The matrix* $L := [W(\mathbf{y}) \otimes I_p]\, H\, [W(\mathbf{z}) \otimes I_p]^T$ *is a block Loewner matrix in* $\mathcal{L}_p(\mathbf{y}, \mathbf{z})$ *whose* $p \times p$ *matrices* $C_1, \ldots, C_n, D_1, \ldots, D_n$ *are given by (up to an arbitrary additive matrix* $\xi \in \mathbb{C}^{p\times p}$*)*

$$\begin{bmatrix} C_1 \\ \vdots \\ C_n \\ D_1 \\ \vdots \\ D_n \end{bmatrix} = [W(\mathbf{y}, \mathbf{z}) \otimes I_p] \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{2n-2} \\ \xi \end{bmatrix}.$$

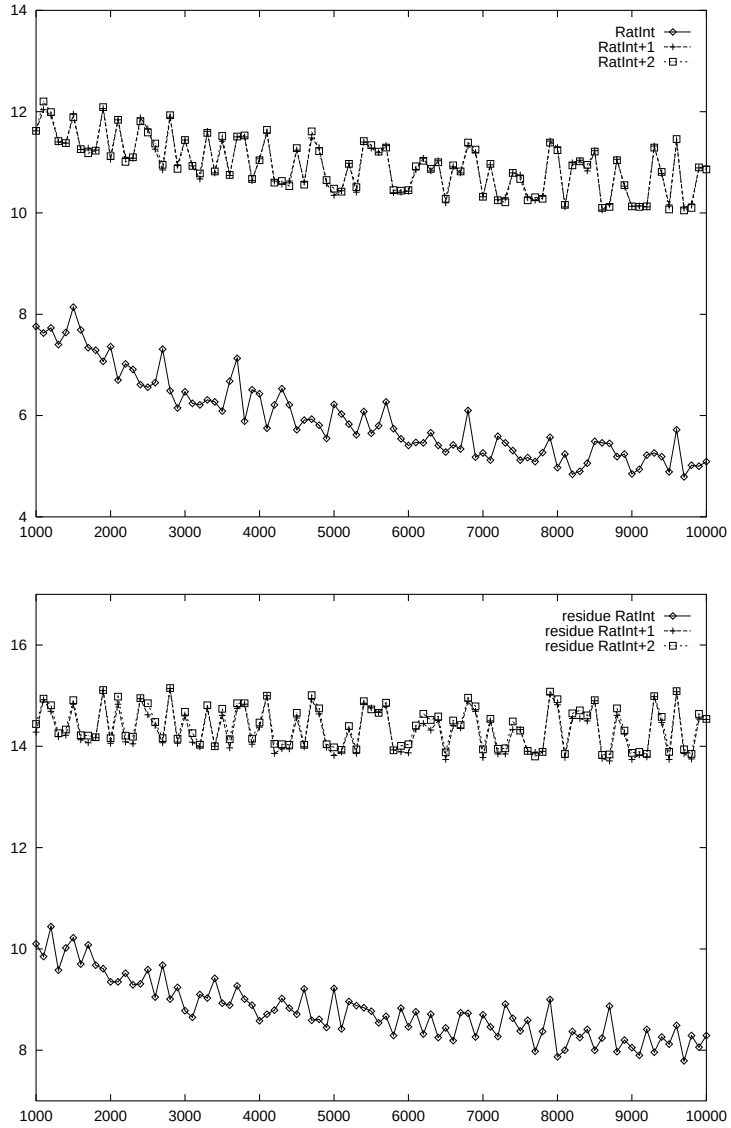


FIGURE 4. $-\log_{10} \dfrac{\|\hat{x}_n - x_n\|_\infty}{\|x_n\|_\infty}$ and $-\log_{10} \dfrac{\|b_n - H_n \hat{x}_n\|_\infty}{\|b_n\|_\infty}$ versus n

PROOF. The scalar case ($p = 1$) was proven by Fiedler [**79**], cf. Theorem 68. The generalization to the block case is straightforward. □

The Kronecker product $A \otimes B$ is nonsingular if A and B are both nonsingular. This implies that $W(\mathbf{y}) \otimes I_p$ and $W(\mathbf{z}) \otimes I_p$ are nonsingular and thus L is nonsingular.

A judicious choice of the points $\mathbf{y}$ and $\mathbf{z}$ enables us to transform the block Hankel system $Hx = b$ into a block Loewner system $Lx' = b'$ in $\mathcal{O}(p^2 n \log n)$ flops.

Let $\omega := \exp(2\pi\, i/n)$ and suppose from now on that $y_k = \omega^{k-1}$ for $k = 1, \ldots, n$. Also, let $\zeta := \exp(\pi\, i/n)$ and suppose from now on that $z_k = \zeta y_k$ for $k = 1, \ldots, n$. Let the matrices Ω_n, $D_{n,\omega}$ and $D_{n,\zeta}$ be defined as in Subsection 1.1.

THEOREM 75. *The solution to the block Hankel system $Hx = b$ is given by*

$$x = \left[\, \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}} \otimes I_p \,\right] x'$$

where $x' := L^{-1} b'$ with $b' := \left[\, \sqrt{n}\, \overline{D_{n,\omega} \Omega_n} \otimes I_p \,\right] b$.

PROOF. Theorem 74 implies that $x = H^{-1} b$ is given by $x = [W(\mathbf{z}) \otimes I_p]^T x'$ where $x' := L^{-1} b'$ with $b' := [W(\mathbf{y}) \otimes I_p] b$. In the proof of Theorem 69 it is shown that $W(\mathbf{y}) = \sqrt{n}\, \overline{D_{n,\omega} \Omega_n}$ and $[W(\mathbf{z})]^T = \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}}$. As $[W(\mathbf{z}) \otimes I_p]^T = [W(\mathbf{z})]^T \otimes I_p$, the result follows. □

It follows that

$$\begin{bmatrix} x_r \\ x_{p+r} \\ \vdots \\ x_{(n-1)p+r} \end{bmatrix} = \sqrt{n}\, \zeta^{n-1} \overline{D_{n,\zeta}}\, \Omega_n^H\, \overline{D_{n,\omega}} \begin{bmatrix} x'_r \\ x'_{p+r} \\ \vdots \\ x'_{(n-1)p+r} \end{bmatrix}$$

and

$$\begin{bmatrix} b'_r \\ b'_{p+r} \\ \vdots \\ b'_{(n-1)p+r} \end{bmatrix} = \sqrt{n}\, \overline{D_{n,\omega} \Omega_n} \begin{bmatrix} b_r \\ b_{p+r} \\ \vdots \\ b_{(n-1)p+r} \end{bmatrix}$$

for $r = 1, \ldots, p$. Thus the transformations $b \mapsto b'$ and $x' \mapsto x$ can be done in $\mathcal{O}(pn \log n)$ floating point operations.

Let the matrices P, Ω_{2n} and $D_{2n,\zeta}$ be defined as in Subsection 1.1.

THEOREM 76. *The $p \times p$ matrices $C_1, \ldots, C_n, D_1, \ldots, D_n$ of the block Loewner matrix L are given by (up to an arbitrary additive matrix $\xi \in \mathbb{C}^{p \times p}$)*

$$\begin{bmatrix} C_1 \\ \vdots \\ C_n \\ D_1 \\ \vdots \\ D_n \end{bmatrix} = \left[\, \sqrt{2n}\, P^T \overline{D_{2n,\zeta}\, \Omega_{2n}} \otimes I_p \,\right] \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{2n-2} \\ \xi \end{bmatrix}.$$

PROOF. In Theorem 70 it is shown that $W(\mathbf{y}, \mathbf{z}) = \sqrt{2n}\, P^T \overline{D_{2n,\zeta}\, \Omega_{2n}}$. The result then follows immediately from Theorem 74. □

The previous theorem implies that

$$\begin{bmatrix} (C_1)_{k,l} \\ \vdots \\ (C_n)_{k,l} \\ (D_1)_{k,l} \\ \vdots \\ (D_n)_{k,l} \end{bmatrix} = \sqrt{2n}\, P^T \overline{D_{2n,\zeta}\, \Omega_{2n}} \begin{bmatrix} (H_0)_{k,l} \\ (H_1)_{k,l} \\ \vdots \\ (H_{2n-2})_{k,l} \\ (\xi)_{k,l} \end{bmatrix}$$

for $k,l = 1,\ldots,p$. Thus H can be transformed into L in $\mathcal{O}(p^2 n \log n)$ flops.

NOTE. As we have already mentioned in Subsection 1.1, the matrices $W(\mathbf{y})/\sqrt{n}$ and $[W(\mathbf{z})]^T/\sqrt{n}$ are unitary. The Kronecker product of two unitary matrices is a unitary matrix. As

$$L = n\,\big[\frac{1}{\sqrt{n}} W(\mathbf{y}) \otimes I_p\big]\; H\; \big[\frac{1}{\sqrt{n}} [W(\mathbf{z})]^T \otimes I_p\big],$$

it follows that $\|L\|_2 = n\|H\|_2$ and $\|L^{-1}\|_2 = n^{-1}\|H^{-1}\|_2$ and thus $\kappa_2(L) = \kappa_2(H)$. In other words, the spectral condition number is left unchanged.

2.2. An inversion formula for block Loewner matrices.

THEOREM 77. *Let P_k, U_k, $\tilde{P}_k$ and $\tilde{U}_k \in \mathbb{C}^{p\times p}$ for $k = 1,\ldots,n$ be defined by the equations*

$$\begin{aligned} \begin{bmatrix} P_1 & \cdots & P_n \end{bmatrix} L &= \begin{bmatrix} I_p & \cdots & I_p \end{bmatrix} \\ \begin{bmatrix} U_1 & \cdots & U_n \end{bmatrix} L &= \begin{bmatrix} D_1 & \cdots & D_n \end{bmatrix} \\ L \begin{bmatrix} \tilde{P}_1 \\ \vdots \\ \tilde{P}_n \end{bmatrix} &= \begin{bmatrix} I_p \\ \vdots \\ I_p \end{bmatrix} \\ L \begin{bmatrix} \tilde{U}_1 \\ \vdots \\ \tilde{U}_n \end{bmatrix} &= \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}. \end{aligned}$$

Then the inverse of L is given by

$$L^{-1} = \Big[\frac{\tilde{U}_k\, P_l - \tilde{P}_k\, U_l}{y_l - z_k}\Big]_{k,l=1}^{n}. \tag{94}$$

PROOF. See Van Barel and Vavřín [**280**, Theorem 2.2]. □

The $p \times p$ matrices that appear in the inversion formula (94) can be computed by solving the following linearized rational interpolation problems.

THEOREM 78. *Define*

$$\begin{aligned} U(z) &:= f_{\mathbf{y}}(z)I_p - \sum_{k=1}^{n} f_{\mathbf{y},k}(z)U_k \\ V(z) &:= -\sum_{k=1}^{n} f_{\mathbf{y},k}(z)U_k C_k \\ \tilde{U}(z) &:= f_{\mathbf{z}}(z)I_p - \sum_{k=1}^{n} f_{\mathbf{z},k}(z)\tilde{U}_k \\ \tilde{V}(z) &:= -\sum_{k=1}^{n} f_{\mathbf{z},k}(z)D_k\tilde{U}_k. \end{aligned}$$

Then the pair $\big(V(z), U(z)\big)$ *is the only pair such that*

- $U(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg U(z) = n$, $V(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg V(z) < n$, $U(z)$ *is monic*
- $V(y_k) = U(y_k)C_k$ *and* $V(z_k) = U(z_k)D_k$ *for* $k = 1, \ldots, n$.

The pair $\big(\tilde{V}(z), \tilde{U}(z)\big)$ *is the only pair such that*

- $\tilde{U}(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg \tilde{U}(z) = n$, $\tilde{V}(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg \tilde{V}(z) < n$, $\tilde{U}(z)$ *is monic*
- $\tilde{V}(y_k) = C_k\tilde{U}(y_k)$ *and* $\tilde{V}(z_k) = D_k\tilde{U}(z_k)$ *for* $k = 1, \ldots, n$.

Moreover, these pairs represent the same matrix rational function,

$$[U(z)]^{-1}V(z) \equiv \tilde{V}(z)[\tilde{U}(z)]^{-1}.$$

PROOF. See Van Barel and Vavřín [**280**, Theorem 3.1]. □

As $U(z)$ is monic, $\det U(z) = z^{pn}$ + lower order terms and hence $\det U(z) \neq 0$ a.e. In other words, $U(z)$ is invertible a.e. The same holds for $\tilde{U}(z)$.

THEOREM 79. *Define*

$$\begin{aligned} P(z) &:= \sum_{k=1}^{n} f_{\mathbf{y},k}(z)P_k \\ Q(z) &:= f_{\mathbf{y}}(z)I_p + \sum_{k=1}^{n} f_{\mathbf{y},k}(z)P_k C_k \\ \tilde{P}(z) &:= \sum_{k=1}^{n} f_{\mathbf{z},k}(z)\tilde{P}_k \\ \tilde{Q}(z) &:= f_{\mathbf{z}}(z)I_p + \sum_{k=1}^{n} f_{\mathbf{z},k}(z)D_k\tilde{P}_k. \end{aligned}$$

Then the pair $\big(Q(z), P(z)\big)$ *is the only pair such that*

- $Q(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg Q(z) = n$, $P(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg P(z) < n$, $Q(z)$ *is monic*
- $Q(y_k) = P(y_k)C_k$ *and* $Q(z_k) = P(z_k)D_k$ *for* $k = 1, \ldots, n$.

The pair $\big(\tilde{Q}(z), \tilde{P}(z)\big)$ *is the only pair such that*

- $\tilde{Q}(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg \tilde{Q}(z) = n$, $\tilde{P}(z) \in \mathbb{C}[z]^{p\times p}$ *and* $\deg \tilde{P}(z) < n$, $\tilde{Q}(z)$ *is monic*
- $\tilde{Q}(y_k) = C_k \tilde{P}(y_k)$ *and* $\tilde{Q}(z_k) = D_k \tilde{P}(z_k)$ *for* $k = 1, \ldots, n$.

Moreover, these pairs represent the same matrix rational function,

$$[Q(z)]^{-1} P(z) \equiv \tilde{P}(z)[\tilde{Q}(z)]^{-1}.$$

PROOF. See Van Barel and Vavřín [**280**, Theorem 3.2]. □

Observe that $Q(z)$ and $\tilde{Q}(z)$ are monic and thus invertible a.e.

Note that if $U(y_k)$ and $U(z_k)$, $k = 1, \ldots, n$, are nonsingular, then the matrix rational function $[U(z)]^{-1}V(z)$ satisfies the proper rational interpolation conditions

$$[U(y_k)]^{-1}V(y_k) = C_k \quad \text{and} \quad [U(z_k)]^{-1}V(z_k) = D_k$$

for $k = 1, \ldots, n$. Similarly, if $P(y_k)$ and $P(z_k)$, $k = 1, \ldots, n$, are nonsingular, then the matrix rational function $[P(z)]^{-1}Q(z)$ satisfies the proper rational interpolation conditions

$$[P(y_k)]^{-1}Q(y_k) = C_k \quad \text{and} \quad [P(z_k)]^{-1}Q(z_k) = D_k$$

for $k = 1, \ldots, n$. The matrix rational functions $[U(z)]^{-1}V(z)$ and $[P(z)]^{-1}Q(z)$ are different because their degree structure is different. The same conclusions hold for $\tilde{V}(z)[\tilde{U}(z)]^{-1}$ and $\tilde{Q}(z)[\tilde{P}(z)]^{-1}$.

An easy calculation reveals the following connections with the $p \times p$ matrices that appear in the inversion formula (94):

$$\begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix} = -\frac{1}{n}[D_{n,\omega} \otimes I_p] \begin{bmatrix} U(y_1) \\ \vdots \\ U(y_n) \end{bmatrix}, \quad \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} = \frac{1}{n}[D_{n,\omega} \otimes I_p] \begin{bmatrix} P(y_1) \\ \vdots \\ P(y_n) \end{bmatrix},$$

$$\begin{bmatrix} \tilde{U}_1 \\ \vdots \\ \tilde{U}_n \end{bmatrix} = -\frac{1}{n}\overline{\zeta^{n-1}}\,[D_{n,\omega} \otimes I_p] \begin{bmatrix} \tilde{U}(z_1) \\ \vdots \\ \tilde{U}(z_n) \end{bmatrix}, \quad \begin{bmatrix} \tilde{P}_1 \\ \vdots \\ \tilde{P}_n \end{bmatrix} = \frac{1}{n}\overline{\zeta^{n-1}}\,[D_{n,\omega} \otimes I_p] \begin{bmatrix} \tilde{P}(z_1) \\ \vdots \\ \tilde{P}(z_n) \end{bmatrix}.$$

As the points $\mathbf{y}$ are the nth roots of unity and $\mathbf{z} = \zeta\mathbf{y}$, the right-hand sides of these formulae can be evaluated in $\mathcal{O}(p^2 n \log n)$ flops. The product $x' = L^{-1}b'$ can then be calculated in $4np^2 + \mathcal{O}(pn \log n)$ flops. Indeed, the matrix L^{-1} can be written as

$$\begin{aligned} L^{-1} \quad = \quad & \operatorname{diag}(\tilde{P}_1, \ldots, \tilde{P}_n)\, C_p \operatorname{diag}(U_1, \ldots, U_n) \\ & - \operatorname{diag}(\tilde{U}_1, \ldots, \tilde{U}_n)\, C_p \operatorname{diag}(P_1, \ldots, P_n) \end{aligned} \tag{95}$$

if C_p is given by the $np \times np$ block Cauchy matrix

$$C_p := \left[\frac{I_p}{z_k - y_l}\right]_{k,l=1}^{n}.$$

Note that $C_p = C \otimes I_p$ if $C = C_1$ is the corresponding $n \times n$ Cauchy matrix

$$C = C_1 := \left[\frac{1}{z_k - y_l}\right]_{k,l=1}^{n},$$

cf. Equation (92). As we have already seen in Subsection 1.2, the product of C with a vector in $\mathbb{C}^n$ can be evaluated in $\mathcal{O}(n \log n)$ flops. Hence, because of (95), the product $L^{-1}b'$ can indeed be evaluated in $4np^2 + \mathcal{O}(pn \log n)$ flops.

2.3. The algorithms BLOCKRATINT$_\lambda$ **and** BLOCKRATINT$_\rho$**.** Define $s_k := y_k$ and $s_{n+k} := z_k$ for $k = 1, \ldots, n$. Define the block column vectors $f_{\lambda,1}, \ldots, f_{\lambda,2n} \in \mathbb{C}^{2p \times p}$ as

$$f_{\lambda,k}^T := \begin{bmatrix} I_p & -C_k \end{bmatrix}, \quad f_{\lambda,n+k}^T := \begin{bmatrix} I_p & -D_k \end{bmatrix}, \qquad k = 1, \ldots, n.$$

The interpolation problems that appear in the second part of Theorems 78 and 79 can be formulated as

$$f_{\lambda,k}^T B_\lambda(s_k) = \begin{bmatrix} O_p & O_p \end{bmatrix}, \qquad k = 1, \ldots, 2n, \tag{96}$$

where

$$B_\lambda(z) = \begin{bmatrix} N_\ell(z) & N_r(z) \\ D_\ell(z) & D_r(z) \end{bmatrix} \in \mathbb{C}[z]^{2p \times 2p}$$

where $\deg N_\ell(z) = n$, $\deg D_\ell(z) \leq n-1$, $\deg N_r(z) \leq n-1$ and $\deg D_r(z) = n$, and $N_\ell(z)$ as well as $D_r(z)$ monic, i.e., where $B_\lambda(z)$ is a monic $2p \times 2p$ block matrix polynomial of degree n. Theorems 78 and 79 assert that this problem has a unique solution given by

$$B_\lambda^\star(z) := \begin{bmatrix} \tilde{Q}(z) & \tilde{V}(z) \\ \tilde{P}(z) & \tilde{U}(z) \end{bmatrix}.$$

Similarly, define the block column vectors $f_{\rho,1}, \ldots, f_{\rho,2n} \in \mathbb{C}^{2p \times p}$ as

$$f_{\rho,k} := \begin{bmatrix} I_p \\ -C_k \end{bmatrix}, \quad f_{\rho,n+k} := \begin{bmatrix} I_p \\ -D_k \end{bmatrix}, \qquad k = 1, \ldots, n.$$

The interpolation problems that appear in the first part of Theorems 78 and 79 can now be formulated as

$$B_\rho(s_k) f_{\rho,k} = \begin{bmatrix} O_p \\ O_p \end{bmatrix}, \qquad k = 1, \ldots, 2n, \tag{97}$$

where

$$B_\rho(z) = \begin{bmatrix} N_u(z) & D_u(z) \\ N_\ell(z) & D_\ell(z) \end{bmatrix} \in \mathbb{C}[z]^{2p \times 2p}$$

where $\deg N_u(z) = n$, $\deg D_u(z) \leq n-1$, $\deg N_\ell(z) \leq n-1$ and $\deg D_\ell(z) = n$, and $N_u(z)$ as well as $D_\ell(z)$ monic, i.e., where $B_\rho(z)$ is a monic $2p \times 2p$ block matrix polynomial of degree n. Theorems 78 and 79 assert that this problem has a unique solution given by

$$B_\rho^\star(z) := \begin{bmatrix} Q(z) & P(z) \\ V(z) & U(z) \end{bmatrix}.$$

This corresponds to the interpolation problems that we have considered in Section 2 of Chapter 8.

The algorithms BLOCKRATINT$_\lambda$ and BLOCKRATINT$_\rho$ enable us to compute $B_\lambda^\star(z)$ and $B_\rho^\star(z)$, respectively. Note that only the second block row of $B_\lambda^\star(z)$ and the second block column of $B_\rho^\star(z)$ are needed to compute the inversion parameters.

What is the arithmetic complexity of this approach? It takes $2p^3n^2 + \mathcal{O}(n)$ complex multiplications and $2p^3n^2 + \mathcal{O}(n)$ complex additions to compute the second block row of $B^\star_\lambda(z)$. The updating of the residuals costs $4p^3n^2 + \mathcal{O}(n)$ complex multiplications and $4p^3n^2 + \mathcal{O}(n)$ complex additions. Thus the overall cost is $6p^3n^2 + \mathcal{O}(n)$ complex multiplications and $6p^3n^2 + \mathcal{O}(n)$ complex additions. Similarly we need $6p^3n^2 + \mathcal{O}(n)$ complex multiplications and $6p^3n^2 + \mathcal{O}(n)$ complex additions to compute the second block column of $B^\star_\rho(z)$. Therefore, by solving the interpolation problems via BLOCKRATINT$_\lambda$ and BLOCKRATINT$_\rho$ we obtain a fast block Hankel solver that has arithmetic complexity $\mathcal{O}(p^3n^2)$. Indeed, the transformation $H \mapsto L$ requires $\mathcal{O}(p^2 n \log n)$ flops whereas the transformations $b \mapsto b'$ and $x' \mapsto x$ both take $\mathcal{O}(pn \log n)$ flops. Once the interpolation problems have been solved, the parameters that appear in the inversion formula for L can be computed in $\mathcal{O}(p^2 n \log n)$ flops and the product $L^{-1}b$ can be evaluated in $4np^2 + \mathcal{O}(pn \log n)$ flops. A Fortran 90 as well as a Matlab implementation are available. In the Fortran version the FFTs are calculated via FFTPACK. In the next subsection we will discuss a few numerical experiments. They show the effectiveness of our approach.

2.4. Numerical examples. A matrix-vector product involving an $n \times n$ Hankel matrix amounts to a convolution of two vectors or, equivalently, the product of two polynomials and can thus be calculated via FFTs in $\mathcal{O}(n \log n)$ flops [**35**]. By writing an approximation $\hat{x}$ to $x = H^{-1}b$ as

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ 0 \\ \vdots \\ 0 \\ \hline \vdots \\ \hline \hat{x}_{(n-1)p+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{x}_2 \\ 0 \\ \vdots \\ 0 \\ \hline \vdots \\ \hline 0 \\ \hat{x}_{(n-1)p+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hat{x}_p \\ \hline \vdots \\ \hline 0 \\ \vdots \\ 0 \\ \hat{x}_{np} \end{bmatrix},$$

it follows that the residue $r := b - H\hat{x}$ can be calculated in $p^2 n + \mathcal{O}(p^2 n \log n)$ flops. This implies that improving an approximation for x iteratively does not add to the $\mathcal{O}(p^3n^2)$ complexity of our algorithm.

In the following examples we used the Fortran 90 version of our package. The calculations were done by an IBM SP2 with machine epsilon $\epsilon \approx 0.12\,10^{-6}$ in single precision and $\epsilon \approx 0.22\,10^{-15}$ in double precision.

EXAMPLE 31. We consider single precision $np \times np$ real block Hankel matrices $H_{n,p}$ having $p \times p$ blocks whose entries are random uniformly distributed in $[-1, 1]$. The right-hand sides $b_{n,p} \in \mathbb{R}^{np}$ are calculated in double precision such that

$$x_{n,p} := H^{-1}_{n,p} b_{n,p} = [\ 1 \quad \cdots \quad 1\ ]^T.$$

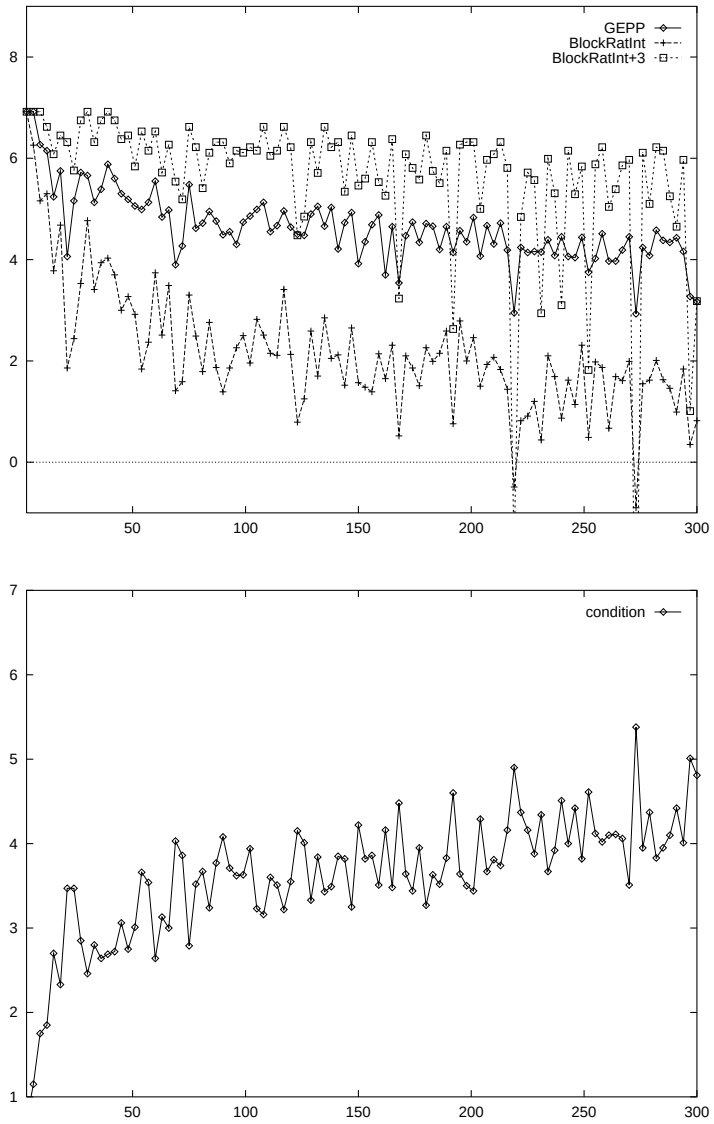


FIGURE 5. $-\log_{10} \dfrac{\|\hat{x}_{n,p} - x_{n,p}\|_\infty}{\|x_{n,p}\|_\infty}$ and $\log_{10} \kappa_\infty(H_{n,p})$ versus np

We fix the block size $p = 3$ and let $n = 1(1)100$. Figure 5 shows the results obtained by our algorithm (before and after three steps of iterative improvement in which the residue was calculated in double precision) and the results obtained via Gaussian elimination with partial pivoting (GEPP) using the LAPACK routines CGETRF and CGETRS. We have plotted $-\log_{10}(\|\hat{x}_{n,p} - x_{n,p}\|_\infty / \|x_{n,p}\|_\infty)$. Also shown is $\log_{10} \kappa_\infty(H_{n,p})$. The condition number was calculated via the routine CGECON.

EXAMPLE 32. We consider the block Hankel matrix

$$H_{2n} := \begin{bmatrix} 1 & 2 & \cdots & n \\ 2 & & \iddots & 0 \\ \vdots & \iddots & \iddots & \vdots \\ n & 0 & \cdots & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$$

for $n = 1(1)500$. We define the right-hand side $b_{2n} = [\ b_1^{(2n)} \quad \cdots \quad b_{2n}^{(2n)}\ ]^T$ as

$$b_{2k-1}^{(2n)} = 0, \qquad b_{2k}^{(2n)} = n(n+1) - k(k-1), \qquad k = 1, \ldots, n.$$

Then the solution to $H_{2n}x_{2n} = b_{2n}$ is given by $x_{2n} = [\ 1 \quad \cdots \quad 1\ ]^T$ as one can easily verify. Figure 6 shows the results obtained by our algorithm (before and after one step of iterative improvement). The calculations were done in double precision. Also shown is $\log_{10} \kappa_\infty(H_{2n})$. The condition number was calculated via LAPACKs routine CGECON.

3. A superfast Hankel solver

We will now return to the setting of Section 1. Let n be a positive integer, let $H = H_n := [h_{k+l}]_{k,l=0}^{n-1}$ be a nonsingular $n \times n$ complex Hankel matrix and let $b \in \mathbb{C}^n$. We consider the problem of computing $x := H^{-1}b$. The fast Hankel solver that we have presented in Section 1 consists of the following components:

1. The Hankel system $Hx = b$ is transformed into a Loewner system $Lx' = b'$. The solution x' is transformed back into x.
2. An explicit formula for L^{-1} is available. The solution x' is found as the product $L^{-1}b'$.
3. This inversion formula for L contains certain parameters. These are calculated by solving two linearized rational interpolation problems.

By considering Loewner matrices that are based on roots of unity we can use FFTs to perform part 1 and 2 with arithmetic complexity $\mathcal{O}(n \log n)$ flops. In Section 1 we have used our $\mathcal{O}(n^2)$ algorithm RATINT to solve the interpolation problems of part 3 and hence the Hankel solver that we have presented there has an overall complexity of $\mathcal{O}(n^2)$ flops. However, if we suppose that n is a power of 2, then we can solve the interpolation problems via our divide and conquer algorithm RATINTALL. This immediately leads to a superfast Hankel solver, although only for systems whose size is a power of 2.

Superfast solvers are notoriously unstable when applied to indefinite systems. We have stabilized our algorithm in several ways. Transforming the Hankel system into an interpolation problem allows us to use pivoting at the lowest "fast" level of the interpolation algorithm. If the pivots are too small, then the corresponding interpolation conditions are handled only after the recursive "superfast" part of the algorithm. This leads to a "generically superfast" solver for indefinite Hankel systems, i.e., superfast in case the number of difficult points is small and fast otherwise. The stability can also be enhanced via iterative refinement and downdating. Iterative refinement can be applied at (several or even each) intermediate level(s), based on an inversion formula for coupled Vandermonde matrices, or at the very end, based on an inversion formula for Loewner matrices. The implementation and combination

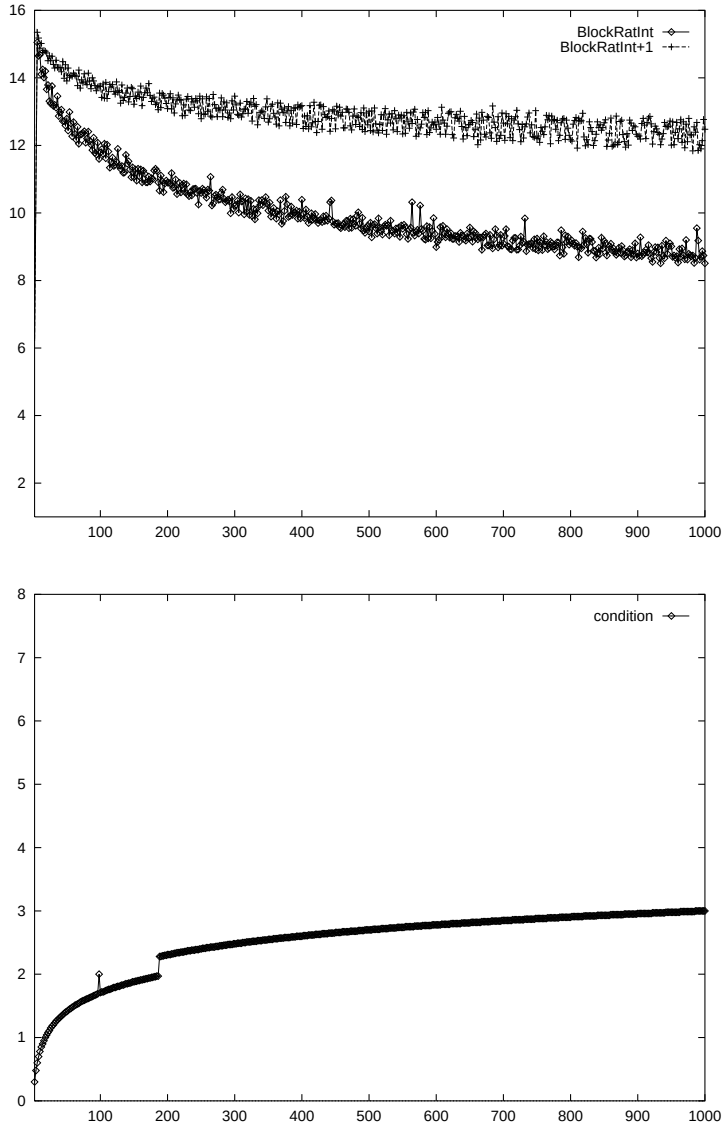


FIGURE 6. $-\log_{10} \dfrac{\|\hat{x}_{2n} - x_{2n}\|_\infty}{\|x_{2n}\|_\infty}$ and $\log_{10} \kappa_\infty(H_{2n})$ versus $2n$

of these tools requires the determination of several parameters that have a mutual effect on each other. How to make a good choice for these parameters is still an open problem. In our numerical experiments we tuned these parameters manually. This led to very satisfactory results.

3.1. Numerical experiments. We consider double precision $n \times n$ real Hankel matrices H_n whose entries are random uniformly distributed in $[0, 1]$ with $n = 2^k$ for

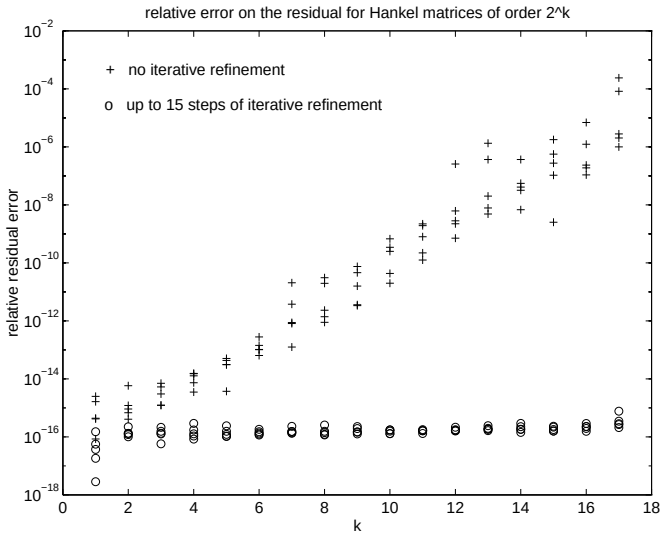


FIGURE 7. $\frac{\|b_n - H_n \hat{x}_n\|_1}{\|b_n\|_1}$ versus $k = \log_2 n$ for $k = 1, \ldots, 17$.

$k = 1, \ldots, 17$. Note that $2^{17} = 131072$. The right-hand sides $b_n \in \mathbb{R}^n$ are calculated such that $x_n := H_n^{-1} b_n = [1 \ \cdots \ 1]^T$. The calculations were done by an IBM SP2 with machine precision $\approx 0.22\,10^{-15}$ in double precision.

Figure 7 shows the results obtained by our algorithm in case no iterative refinement was applied and in case up to 15 steps of iterative refinement were applied to enhance the accuracy of the computed solution to the Hankel system. As a matrix-vector product involving a Hankel matrix amounts to a convolution of two vectors or, equivalently, the product of two polynomials, the residual $r_n := b_n - H_n \hat{x}_n$ can be calculated via FFT in $\mathcal{O}(n \log n)$ flops. Therefore improving an approximation for x_n iteratively does not add substantially to the $\mathcal{O}(n \log^2 n)$ complexity of our algorithm. For each value of k, five Hankel matrices were considered. Let $\hat{x}_n^{(l)}$ be the approximation to the solution x_n after l steps of iterative improvement. We made the algorithm stop as soon as the impact of iterative improvement stagnated,

$$\frac{\|b_n - H_n \hat{x}_n^{(l)}\|_1}{\|b_n\|_1} \geq \frac{1}{2} \frac{\|b_n - H_n \hat{x}_n^{(l-1)}\|_1}{\|b_n\|_1}.$$

Interpolation problems of size less or equal than 2^6 were solved by our fast-only algorithm RATINT. We experimented to find an 'optimal' subproblem size. Our algorithm performed equally well with subproblem sizes of 2^4, 2^5, 2^7 or even 2^8. Its performance seems to be rather insensitive to the exact subproblem size, as long as this is 'near' 2^6.

Our next figures represent timings. As on computer systems measurements of very small execution times are rather inaccurate, we limit the k-axis to that part where the results are meaningful. This is why in the following figures k does not start at one but at a larger value.

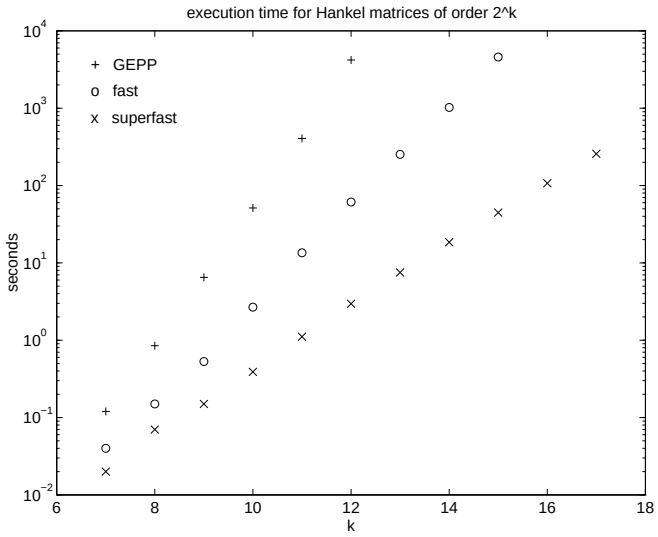


FIGURE 8. Execution time in seconds.

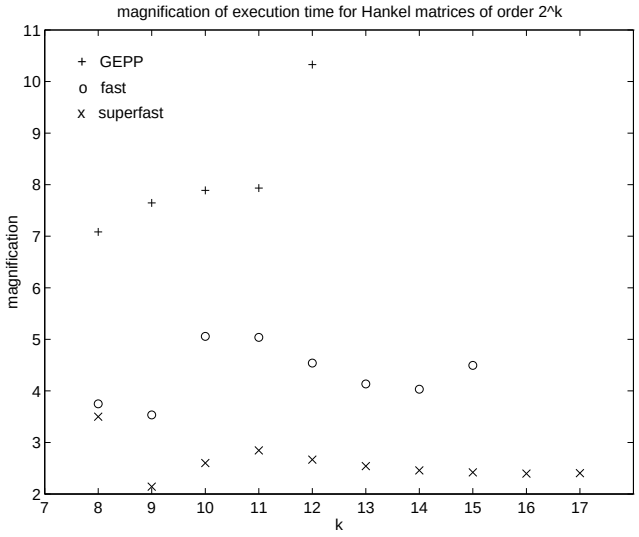


FIGURE 9. Magnification of the execution time.

Figure 8 shows the execution time for Gaussian elimination with partial pivoting (GEPP) (these results were calculated via the LAPACK routines ZGETRF and ZGETRS), our fast algorithm and our superfast algorithm.

Figure 9 presents the results shown in Figure 8 in a different way. It gives the magnification of the execution time. For each k, it tells us by which factor the execution time is to be multiplied if we go from $k-1$ to k.

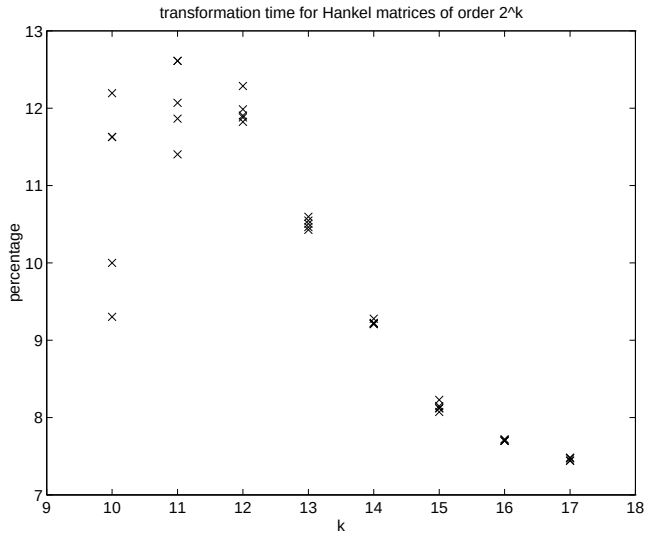


FIGURE 10. Transformation time as percentage of the execution time in case no iterative refinement was applied.

Figure 10 shows that our algorithm spends only a small part of its execution time on the transformation from Hankel to Loewner and back. The solution of the rational interpolation problems is (obviously) the most time-consuming part of the algorithm. For each value of k, five Hankel matrices were considered.

The computed solution to the Hankel system can be refined iteratively. Figure 11 shows how much execution time was spent on iterative refinement as percentage of the execution time in case no iterative refinement was applied. We considered one, two, three or four steps of iterative refinement. For each value of k and each number of iterative refinement steps, five Hankel matrices were considered.

4. A superfast Toeplitz solver

In the previous section we have presented a superfast solver for indefinite Hankel systems. A Hankel system is first transformed into a Loewner system and then an explicit formula for the inverse of a Loewner matrix is used. This inversion formula involves certain parameters that can be computed by solving two linearized rational interpolation problems. We will now consider Toeplitz systems instead of Hankel systems and we will use an explicit formula for the inverse of a Toeplitz matrix. In Subsection 4.1 we will write this inversion formula in such a way that it can be applied by using FFTs. This formula contains certain polynomials. In Subsection 4.2 we will show how these can be computed by solving two linearized rational interpolation problems. Numerical examples will be given in Subsection 4.3. Note that in our superfast Hankel solver the size of the Hankel matrix is limited to a power of 2 whereas the superfast Toeplitz solver that we will present in this section can handle Toeplitz systems of arbitrary size.

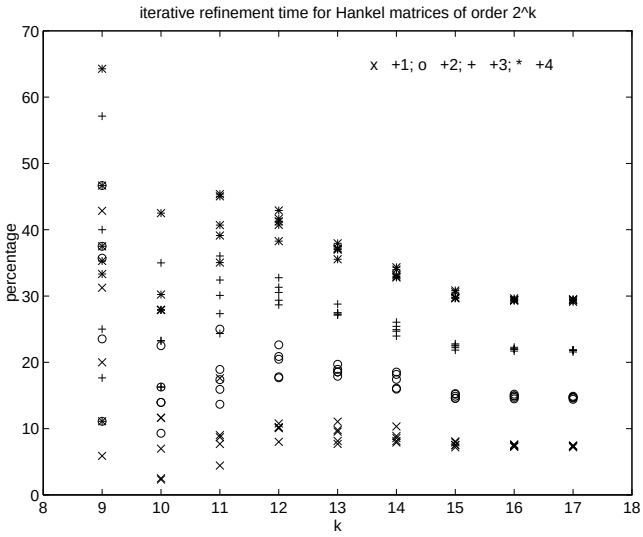


FIGURE 11. Time spent on iterative refinement as percentage of the execution time in case no iterative refinement was applied.

4.1. An inversion formula for Toeplitz matrices. Let n be a positive integer, let $T = T_n := [a_{k-l}]_{k,l=0}^{n-1}$ be a nonsingular $n \times n$ complex Toeplitz matrix and let $b \in \mathbb{C}^n$. We consider the problem of computing $x := T^{-1}b$.

Let us introduce the following notations. To each column vector $u = [u_k]_{k=0}^{n} \in \mathbb{C}^{n+1}$ we associate the polynomial $u(z) := \sum_{k=0}^{n} u_k z^k \in \mathbb{C}[z]$. The column vector $\hat{u}$ is defined as $\hat{u} := [u_{n-k}]_{k=0}^{n}$. Thus, $\hat{u}(z) = z^n u(z^{-1})$.

Let $a_{-n} \in \mathbb{C}$ be arbitrary and let $\tilde{T} = \tilde{T}_n$ be given by the $n \times (n+1)$ matrix

$$\tilde{T} := [a_{k-l}]_{k,l=0}^{n-1,n} = \left[\begin{array}{cccc|c} a_0 & a_{-1} & \cdots & a_{-n+1} & a_{-n} \\ a_1 & a_0 & \ddots & & a_{-n+1} \\ \vdots & & \ddots & \ddots & \vdots \\ a_{n-1} & \cdots & \cdots & a_0 & a_{-1} \end{array}\right] = \left[\begin{array}{c|c} & a_{-n} \\ T & \vdots \\ & a_{-1} \end{array}\right].$$

The polynomials $u(z)$ and $v(z)$ are called the *canonical fundamental system* of T if

- $\tilde{T}u = e_1$ and $u_n = 0$, where $e_1 := [\ 1 \quad 0 \quad \cdots \quad 0\ ]^T$,
- $\tilde{T}v = 0$ and $v_n = 1$.

In other words, $u(z)$ is a polynomial of degree $n-1$ while $v(z)$ is a monic polynomial of degree n such that

$$\left[\begin{array}{cccc} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & & \ddots & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_0 \end{array}\right] \left[\begin{array}{c} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{array}\right] = \left[\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array}\right]$$

and

$$\begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & & \ddots & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = - \begin{bmatrix} a_{-n} \\ a_{-n+1} \\ \vdots \\ a_{-1} \end{bmatrix}.$$

As T is assumed to be nonsingular, these polynomials exist. Moreover, $u(z)$ is unique whereas $v(z)$ is uniquely determined given a particular value of a_{-n}. For our purposes the specific value of a_{-n} is immaterial and thus $v(z)$ is in fact unique up to a linear combination with $u(z)$. Note that by cancelling the last (zero) component of u, one obtains the first column of T^{-1}.

REMARK. Let $w = [\ w_0 \ \cdots \ w_{n-1}\ ]^T \in \mathbb{C}^n$ be the last column of T_n^{-1}. If $T_{n-1} := [a_{k-l}]_{k,l=0}^{n-2}$ is also nonsingular, then Cramer's rule implies that $w_{n-1} \neq 0$ and one may choose $v(z)$ as $v(z) = zw(z)/w_{n-1}$. This choice determines the value of a_{-n}. Also, if T_n is symmetric, then $w(z) = \hat{u}(z)$.

The *generating function* $M(t,s)$ of a matrix $M = [m_{k,l}]_{k,l=0}^{p,q}$ is defined as

$$M(t,s) := \sum_{k=0}^{p} \sum_{l=0}^{q} m_{k,l} t^k s^l.$$

THEOREM 80. *The generating function of T^{-1} is given by*

$$T^{-1}(t,s) = \frac{u(t)\hat{v}(s) - v(t)\hat{u}(s)}{1 - ts}. \tag{98}$$

PROOF. See Heinig and Rost [**140**, p. 32]. □

The matrix whose generating function is given by the right-hand side of (98) is called the *Toeplitz Bezoutian* of the polynomials $u(z)$ and $v(z)$.

Let $N \geq n$ be a power of 2. From the previous theorem we will now derive a formula for T^{-1} that will enable us to calculate the matrix-vector product $T^{-1}b$ in $\mathcal{O}(N \log N)$ floating point operations.

Define $\omega_0, \ldots, \omega_{2N-1}$ as the $2N$th roots of unity,

$$\omega_k := \exp\Big(\frac{2\pi i}{2N} k\Big), \qquad k = 0, 1, \ldots, 2N-1,$$

and let $\omega_k^+ := \omega_{2k}$ and $\omega_k^- := \omega_{2k+1}$ for $k = 0, 1, \ldots, N-1$. Note that $\omega_0^+, \ldots, \omega_{N-1}^+$ are the Nth roots of unity whereas $\omega_0^-, \ldots, \omega_{N-1}^-$ are the Nth roots of -1. Let $\eta := \exp(\pi i/N)$. Define the matrices $\mathcal{F}_+$ and $\mathcal{F}_-$ as

$$\mathcal{F}_+ := [\ (\omega_k^+)^l\ ]_{k,l=0}^{N-1} \quad \text{and} \quad \mathcal{F}_- := [\ (\omega_k^-)^l\ ]_{k,l=0}^{N-1}.$$

Then $\mathcal{F}_+/\sqrt{N}$ is unitary and $\mathcal{F}_- = \mathcal{F}_+ \operatorname{diag}(1, \eta, \ldots, \eta^{N-1})$. Matrix-vector products involving $\mathcal{F}_+$ or $\mathcal{F}_-$ can be evaluated with arithmetic complexity $\mathcal{O}(N \log N)$ via the celebrated FFT.

Let $[T^{-1}]_N$ denote the $N \times N$ matrix

$$[T^{-1}]_N := \begin{bmatrix} T^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{N \times N}.$$

Define $D := \mathrm{diag}\big((\omega_0^+)^{-n}, \ldots, (\omega_{N-1}^+)^{-n} \big)$, $D_\pm(u) := \mathrm{diag}\big(u(\omega_0^\pm), \ldots, u(\omega_{N-1}^\pm) \big)$ and similar for the matrices $D_\pm(v)$.

THEOREM 81. *The matrix $[T^{-1}]_N$ can be expressed as*

$$[T^{-1}]_N = \frac{1}{2}\mathcal{F}_-^{-1}\big[D_-(u)\mathcal{F}_-\mathcal{F}_+^{-1}D_+(v) - D_-(v)\mathcal{F}_-\mathcal{F}_+^{-1}D_+(u)\big] D\mathcal{F}_+. \tag{99}$$

PROOF. As $\mathcal{F}_+^{-1} = \frac{1}{N}\mathcal{F}_+^H$, it follows that

$$\mathcal{F}_-[T^{-1}]_N\mathcal{F}_+^{-1} =: [c_{k,l}]_{k,l=0}^{N-1}$$

where

$$\begin{aligned} c_{k,l} &= \frac{1}{N}T^{-1}(\omega_k^-, 1/\omega_l^+) \\ &= \frac{1}{N}\frac{u(\omega_k^-)\hat{v}(1/\omega_l^+) - v(\omega_k^-)\hat{u}(1/\omega_l^+)}{1-\omega_k^-/\omega_l^+} \\ &= \frac{1}{N}\frac{u(\omega_k^-)v(\omega_l^+) - v(\omega_k^-)u(\omega_l^+)}{1-\omega_k^-/\omega_l^+}[\omega_l^+]^{-n}. \end{aligned}$$

One can easily verify that

$$\mathcal{F}_-\mathcal{F}_+^{-1} = \frac{2}{N}\left[\frac{1}{1-\omega_k^-/\omega_l^+} \right]_{k,l=0}^{N-1}.$$

The expression for $[T^{-1}]_N$ then follows immediately. □

The formula in the previous theorem allows us to compute the product $x = T^{-1}b$ in $\mathcal{O}(N\log N)$ flops provided that the polynomials $u(z)$ and $v(z)$ are known. Indeed,

$$\begin{bmatrix} x \\ 0 \end{bmatrix} = [T^{-1}]_N \begin{bmatrix} b \\ 0 \end{bmatrix}$$

and the multiplication by $[T^{-1}]_N$ can be done via six N-point (inverse) FFTs and $\mathcal{O}(N)$ flops. For preprocessing one has to compute the values of $u(\omega_k)$ and $v(\omega_k)$ for $k = 0, 1, \ldots, 2N-1$, which amounts to two $2N$-points FFTs.

4.2. Interpolation interpretation. The *symbol* of T is defined as the function

$$a : \mathbb{C}_0 \to \mathbb{C} : z \mapsto a(z) := \frac{a_{-n+1}}{z^{n-1}} + \cdots + \frac{a_{-1}}{z} + a_0 + a_1 z + \cdots + a_{n-1}z^{n-1}.$$

THEOREM 82. *Suppose $a_{-n} = 0$. Then the polynomials $u(z)$ and $v(z)$ are the canonical fundamental system of T if and only if the following linearized rational interpolation conditions are satisfied:*

$$\omega_k^n\hat{r}_u(\omega_k) - a(\omega_k)u(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg\hat{r}_u(z) \le 2N-n$, $\hat{r}_{u,2N-n} = 1$ *and* $\deg u(z) \le n$, $u_n = 0$ *and*

$$\omega_k^n\hat{r}_v(\omega_k) - a(\omega_k)v(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg\hat{r}_v(z) \le 2N-n$, $\hat{r}_{v,2N-n} = 0$ *and* $\deg v(z) \le n$, $v_n = 1$.

PROOF. Let $\delta \in \mathbb{C}$. Let $w(z)$ be a polynomial of degree $\leq n$ such that $\tilde{T}w = \delta e_1$. The case $\delta = 1$ corresponds to $w(z) = u(z)$ whereas $\delta = 0$ corresponds to $w(z) = v(z)$. The condition $\tilde{T}w = \delta e_1$ is equivalent to the existence of polynomials $r_-(z)$ and $r_+(z)$ of degree $\leq n-1$ with $r_{-,0} = \delta$ such that

$$a(z)w(z) = r_-(1/z) + z^n r_+(z).$$

It follows that

$$\begin{aligned} a(\omega_k)w(\omega_k) &= r_-(\omega_k^{-1}) + \omega_k^n r_+(\omega_k) \\ &= r_-(\omega_k^{-1}) + \omega_k^{n-2N} r_+(\omega_k) \end{aligned}$$

for $k = 0, 1, \ldots, 2N-1$. Define

$$r(z) := r_-(z) + z^{2N-n} r_+(1/z).$$

Then $r(z)$ is a polynomial of degree $\leq 2N-n$ and $r_0 = \delta$. Thus

$$\begin{aligned} a(\omega_k)w(\omega_k) &= r(\omega_k^{-1}) \\ &= \omega_k^n \, [\omega_k^{2N-n} r(\omega_k^{-1})] \\ &= \omega_k^n \hat{r}(\omega_k) \end{aligned}$$

for $k = 0, 1, \ldots, 2N-1$. In other words,

$$\omega_k^n \hat{r}(\omega_k) - a(\omega_k)w(\omega_k) = 0, \qquad k = 0, 1, \ldots, 2N-1,$$

where $\deg \hat{r}(z) \leq 2N-n$, $\hat{r}_{2N-n} = \delta$ and $\deg w(z) \leq n$. This proves the theorem. □

These interpolation conditions can also be written as follows:

$$\left[\begin{array}{cc} \omega_k^n & -a(\omega_k) \end{array} \right] B(\omega_k) = \left[\begin{array}{cc} 0 & 0 \end{array} \right], \qquad k = 0, 1, \ldots, 2N-1,$$

where

$$B(z) := \left[\begin{array}{cc} \hat{r}_u(z) & \hat{r}_v(z) \\ u(z) & v(z) \end{array} \right] \in \mathbb{C}[z]^{2\times 2}$$

is a 2×2 matrix polynomial. The degree of the first row of $B(z)$ is equal to $2N-n$ whereas the degree of the second row of $B(z)$ is equal to n. Note that if $\tau := 2(n-N)$, then the τ-highest degree coefficient (cf. Section 3 of Chapter 8) of $B(z)$ is equal to I_2. The matrix polynomial $B(z)$ can be computed via the stabilized superfast algorithm RECRATINT. The second row of $B(z)$ leads to the inversion parameters that are needed in Equation (99).

4.3. Numerical experiments. We consider double precision $n \times n$ Toeplitz matrices T_n whose entries are real and random uniformly distributed in $[0, 1]$ with $n = 2^k$ for $k = 1, \ldots, 18$. Note that $2^{18} = 262144$. The right-hand sides $b_n \in \mathbb{R}^n$ are calculated such that $x_n := T_n^{-1} b_n = [1 \ \cdots \ 1]^T$. The calculations were done by an IBM SP2 with machine precision $\approx 0.22\, 10^{-15}$ in double precision.

Figures 12 and 13 show the results obtained by our algorithm in case no iterative refinement is applied (the symbols '+') and in case up to 10 steps of iterative refinement are applied (the symbols 'o') to enhance the accuracy of the computed solution to the Toeplitz system. Interpolation problems of size less than or equal to 2^8 are solved by our fast-only algorithm. For each value of k we consider five (random) Toeplitz matrices.

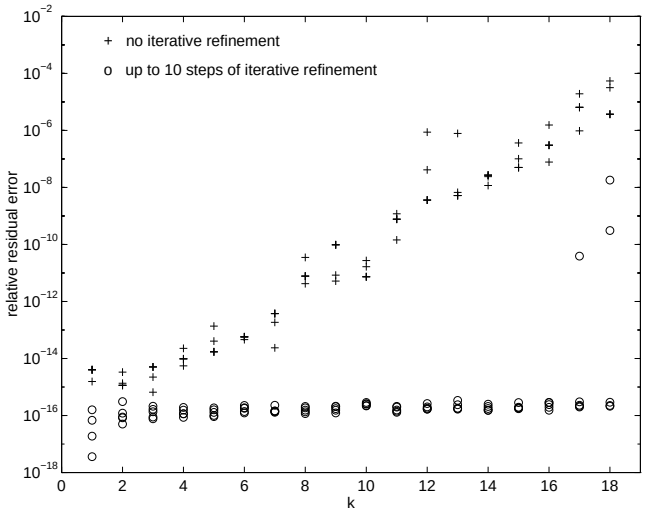


FIGURE 12. $\frac{\|b_n - T_n \hat{x}_n\|_1}{\|b_n\|_1}$ versus $k = \log_2 n$ for $k = 1, \ldots, 18$.

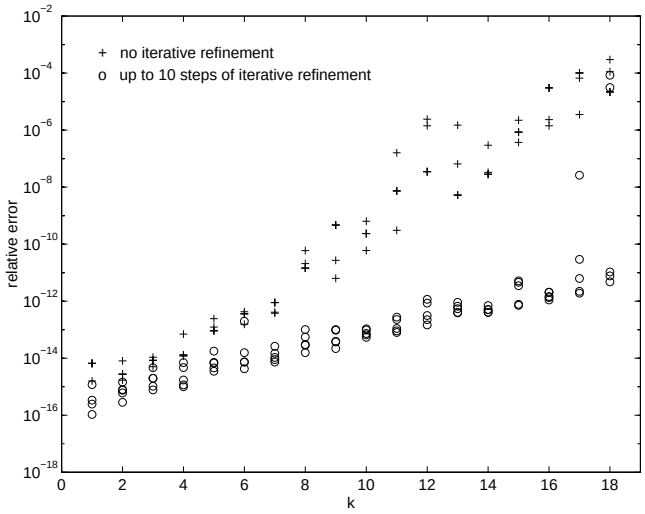


FIGURE 13. $\frac{\|\hat{x}_n - x_n\|_1}{\|x_n\|_1}$ versus $k = \log_2 n$ for $k = 1, \ldots, 18$.

Our next figures represent timings. As on our computer system measurements of execution times are done in units of 0.01 seconds, we limit the k-axis to that part where the results are meaningful. This is why in the following figures k does not start at one but at a larger value.

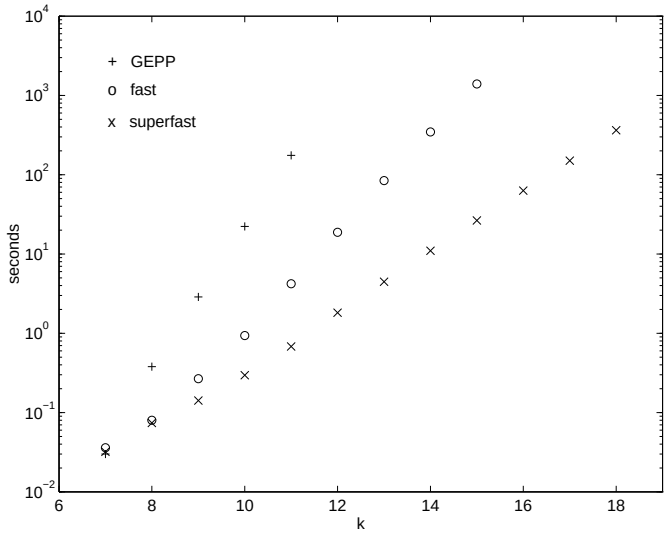


FIGURE 14. Execution time in seconds.

Figure 14 shows the execution time (in seconds) for Gaussian elimination with partial pivoting (these results were calculated via the LAPACK routines ZGETRF and ZGETRS), our fast algorithm and our superfast algorithm in case no iterative refinement is applied. The results are indicated with the symbols '+', 'o' and 'x', respectively.

Figure 15 presents the results shown in Figure 14 in a different way. It gives the magnification of the execution time. For each k, it tells us by which factor the execution time is to be multiplied if we go from $k-1$ to k.

Figures 16 and 17 are related to our superfast solver. For each value of k we consider five (random) Toeplitz matrices. No iterative refinement is applied. Figure 16 shows the percentage of the execution time spent to compute the input data for the interpolation problem formulated in Theorem 82, i.e., the time needed to evaluate the $a(\omega_k)$'s. Figure 17 shows the percentage of the execution time spent to apply the inversion formula given in Theorem 81 once the interpolation problem has been solved.

We also consider matrices of size $n = 10000(5000)100000$. The entries are again real and random uniformly distributed in $[0,1]$ and the right-hand sides are again calculated such that all the entries of the solution vector are equal to one. For each value of n we consider five matrices. Figure 18 shows the execution time (in seconds). The results are indicated with the symbol 'x'. The symbols 'o' correspond to the case where n is a power of two.

One expects that for n in the range $2^{k-1} < n \leq 2^k$ the execution time is more or less equal to that of the system of size 2^k. In practice, the execution time is less. This can be explained as follows. At the lowest level some of the first interpolation problems can be solved via polynomial interpolation, i.e., by applying FFT.

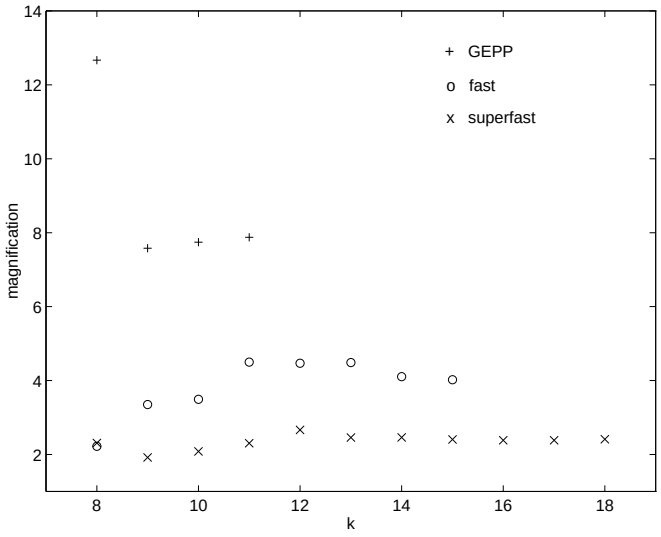


FIGURE 15. Magnification of the execution time.

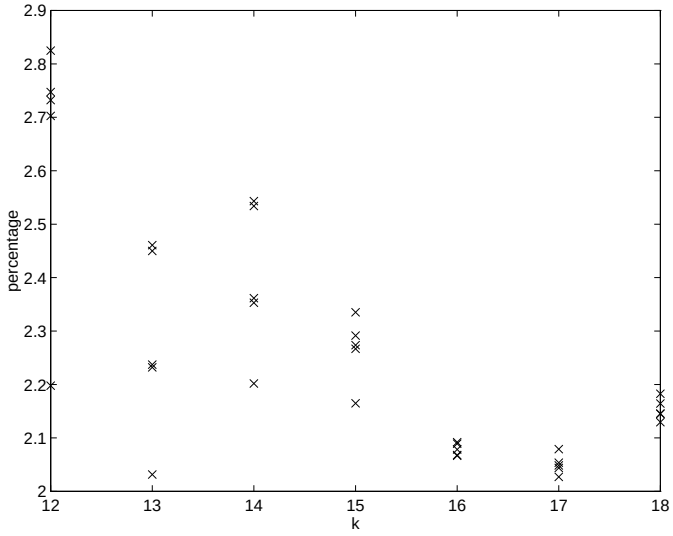


FIGURE 16. Percentage of the execution time spent to compute the input data for the interpolation problem.

The computed solution can be refined iteratively. Figure 19 shows how much execution time is spent on iterative refinement as percentage of the execution time in case no iterative refinement is applied. We consider one, two, three or four steps of iterative refinement. The results are represented by the symbols 'x', 'o', '+' and

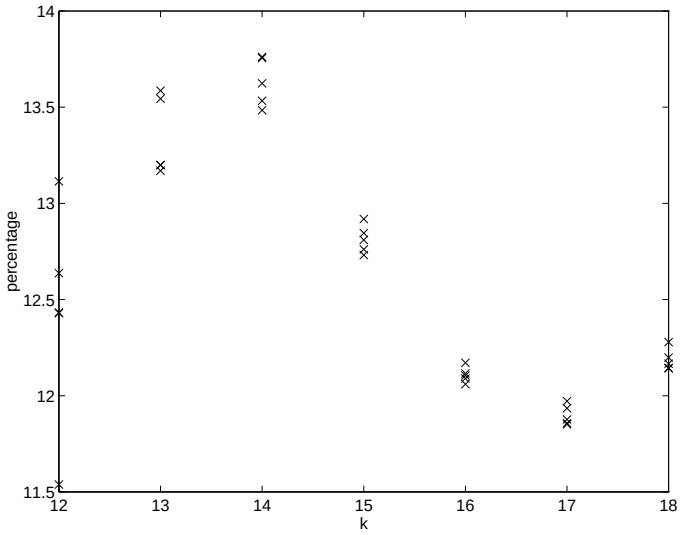


FIGURE 17. Percentage of the execution time spent to apply the inversion formula.

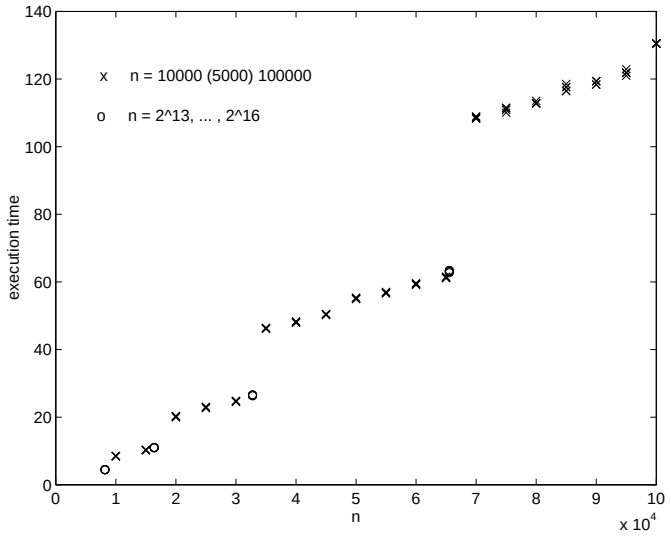


FIGURE 18. Execution time for $n = 10000(5000)100000$ and $n = 2^{13}, \ldots, 2^{16}$.

'$\star$', respectively. For each value of k and each number of iterative refinement steps, five Toeplitz matrices are considered.

So far we have only considered iterative refinement at the Toeplitz level, i.e., we have refined the computed solution to the Toeplitz system iteratively. Iterative refinement can also be applied at the interpolation level, cf. the inversion formula

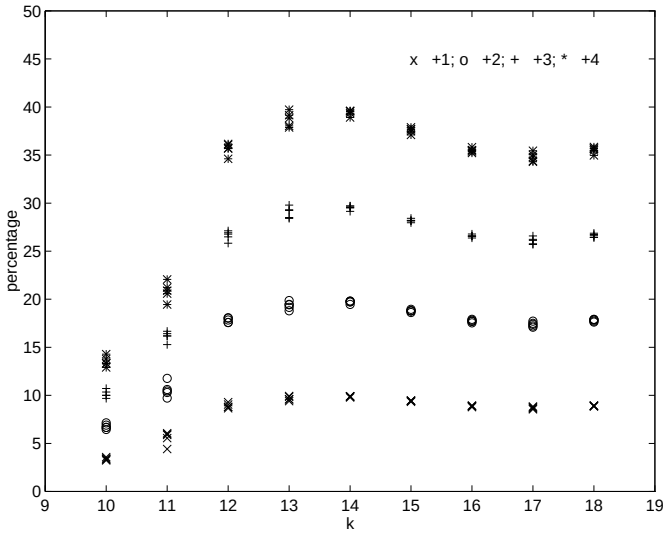


FIGURE 19. Time spent on iterative refinement as percentage of the execution time in case no iterative refinement is applied.

for coupled Vandermonde matrices given in Theorem 67. In our next experiment we apply up to four steps of iterative refinement at the highest interpolation level. The timing results are shown in Figure 20. We compare the execution time spent on this kind of iterative refinement to the total execution time in case no iterative refinement whatsoever is applied. Observe that this kind of iterative refinement is much more expensive than iterative refinement applied at the Toeplitz level.

Iterative refinement at an interpolation level may be preceded by downdating. Numerical experiments indicate that the time needed to search the interpolation points that have to be downdated is approximately 45% of the time needed for one step of iterative refinement.

The following example illustrates how important it is to find the proper combination of the stabilizing techniques that we have developed. For a certain matrix of size 2^{18} whose entries are random uniformly distributed in the interval $[0, 1]$, we have observed the following. By applying at most 10 steps of iterative refinement on the interpolation problems of size 2^{18} (this is the one but highest interpolation level; remember that a matrix of size 2^{18} corresponds to 2^{19} interpolation conditions), by considering 85 difficult points and by applying iterative refinement at the Toeplitz level, we obtain an approximation for the solution whose relative residual error is $\mathcal{O}(10^{-15})$. If we do not consider difficult points and do not use iterative refinement at the interpolation level, then the computed approximation is so bad that iterative refinement at the Toeplitz level fails. The same holds if we only apply iterative refinement on the interpolation problems of size 2^{19}. This clearly shows the importance of combining our stabilizing tools in the correct way. One can of course apply iterative refinement on each interpolation level, but this is very costly. One

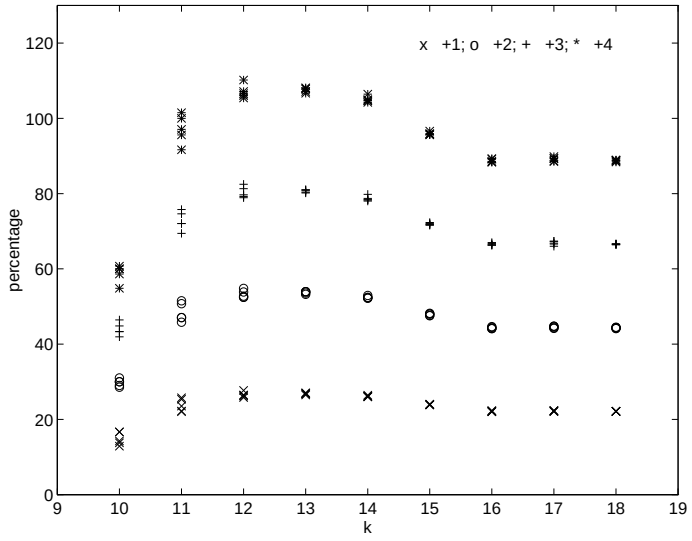


FIGURE 20. Up to four steps of iterative refinement at the highest interpolation level. We compare the corresponding execution time to the total execution time in case no iterative refinement at all is applied.

has to find the correct balance between accuracy and cost. This will be the subject of future research.

Acknowledgements

The results in this chapter were obtained in close collaboration with Marc Van Barel. We thank Georg Heinig for providing us with references [**80, 134, 135, 137, 138, 139**] and for suggesting to use the inversion formula (98).

Bibliography

1. L. F. Abd-Elall, L. M. Delves, and J. D. Reid, *A numerical method for locating the zeros and poles of a meromorphic function*, Numerical Methods for Nonlinear Algebraic Equations (P. Rabinowitz, ed.), Gordon and Breach, 1970, pp. 47–59. {**78**}
2. M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, Dover, 1970. {**98, 102**}
3. I. A. Aĭzenberg and A. P. Yuzhakov, *Integral representations and residues in multidimensional complex analysis*, Translations of Mathematical Monographs, vol. 58, American Mathematical Society, Providence, Rhode Island, 1983. {**xxv, 83, 84, 86, 88, 125**}
4. G. S. Ammar and W. B. Gragg, *The generalized Schur algorithm for the superfast solution of Toeplitz systems*, Rational Approximation and its Applications in Mathematics and Physics (J. Gilewicz, M. Pindor, and W. Siemaszko, eds.), Lecture Notes in Mathematics, vol. 1237, Springer, 1987, pp. 315–330. {**155**}
5. ———, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl. **9** (1988), no. 1, 61–76. {**155**}
6. ———, *Numerical experience with a superfast real Toeplitz solver*, Linear Algebr. Appl. **121** (1989), 185–206. {**155**}
7. E. G. Anastasselou, *A formal comparison of the Delves-Lyness and Burniston-Siewert methods for locating the zeros of analytic functions*, IMA J. Numer. Anal. **6** (1986), 337–341. {**13**}
8. E. G. Anastasselou and N. I. Ioakimidis, *Application of the Cauchy theorem to the location of zeros of sectionally analytic functions*, J. Appl. Math. Phys. **35** (1984), 705–711. {**13**}
9. ———, *A generalization of the Siewert-Burniston method for the determination of zeros of analytic functions*, J. Math. Phys. **25** (1984), no. 8, 2422–2425. {**13**}
10. ———, *A new method for obtaining exact analytical formulae for the roots of transcendental functions*, Lett. Math. Phys. **8** (1984), 135–143. {**13**}
11. ———, *A new approach to the derivation of exact analytical formulae for the zeros of sectionally analytic functions*, J. Math. Anal. Appl. **112** (1985), no. 1, 104–109. {**13**}
12. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK users' guide*, SIAM, 1994. {**37, 92**}
13. A. C. Antoulas, *On the scalar rational interpolation problem*, IMA J. Math. Control Inf. **3** (1986), 61–88. {**68, 139**}
14. ———, *Rational interpolation and the Euclidean algorithm*, Linear Algebr. Appl. **108** (1988), 157–171. {**68, 139**}
15. A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems, *On the solution of the minimal rational interpolation problem*, Linear Algebr. Appl. **137/138** (1990), 511–573. {**68, 139**}
16. L. Atanassova, *On the simultaneous determination of the zeros of an analytic function inside a simple smooth closed contour in the complex plane*, J. Comput. Appl. Math. **50** (1994), 99–107. {**13**}

17. V. M. Badkov, *Uniform asymptotic representations of orthogonal polynomials*, Proceedings of the Steklov Institute of Mathematics **2** (1985), 5–41. {**115**}
18. B. Beckermann, *The structure of the singular solution table of the M-Padé approximation problem*, J. Comput. Appl. Math. **32** (1990), 3–15. {**139**}
19. B. Beckermann and E. Bourreau, *How to choose modified moments?*, Publication ANO 368, Université de Lille, 1997, To appear in *J. Comput. Appl. Math.* {**18**}
20. B. Beckermann and C. Carstensen, *Global identities in the non-normal Newton-Padé approximation table*, J. Approx. Theory **74** (1993), 199–220. {**139**}
21. ———, *QD-type algorithms for the nonnormal Newton-Padé approximation table*, Constr. Approx. **12** (1996), no. 3, 307–329. {**139**}
22. M. Beckers and A. Haegemans, *Transformations of integrands for lattice rules*, Numerical Integration: Recent Developments, Software and Applications (T. O. Espelid and A. Genz, eds.), Kluwer Academic Publishers, 1992, pp. 329–340. {**93**}
23. V. Belevitch, *Interpolation matrices*, Philips Res. Repts **25** (1970), 337–369. {**139**}
24. C. Belingeri and P. E. Ricci, *On asymptotic formulas for the first zero of the Bessel function J_ν*, Journal of Information & Optimization Sciences **17** (1996), no. 2, 267–274. {**98**}
25. W. Bergweiler, *Iteration of meromorphic functions*, Bulletin (New Series) of the Amer. Math. Soc. **29** (1993), no. 2, 151–188. {**135**}
26. J. Berntsen, T. O. Espelid, and A. Genz, *Algorithm 698: DCUHRE—An adaptive multidimensional integration routine for a vector of integrals*, ACM Trans. Math. Softw. **17** (1991), no. 4, 452–456. {**93**}
27. J.-P. Berrut, *Rational functions for guaranteed and experimentally well-conditioned global interpolation*, Comput. Math. Applic. **15** (1988), no. 1, 1–16. {**139**}
28. ———, *Linear rational interpolation of continuous functions over an interval*, Mathematics of Computation 1943–1993: A half-century of computational mathematics, Proceedings of Symposia in Applied Mathematics, vol. 48, American Mathematical Society, 1994, pp. 261–264. {**139**}
29. ———, *The barycentric weights of rational interpolation with prescribed poles*, J. Comput. Appl. Math. (1997), 45–52. {**139**}
30. J.-P. Berrut and H. D. Mittelmann, *Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval*, Computers Math. Applic. **33** (1997), no. 6, 77–86. {**139**}
31. ———, *Matrices for the direct determination of the barycentric weights of rational interpolation*, J. Comput. Appl. Math. **78** (1997), no. 2, 355–370. {**68, 139**}
32. D. A. Bini, *Numerical computation of polynomial zeros by means of Aberth's method*, Numerical Algorithms **13** (1996), 179–200. {**13**}
33. D. A. Bini, L. Gemignani, and B. Meini, *Factorization of analytic functions by means of Koenig's Theorem and Toeplitz computations*, Unpublished manuscript. {**13**}
34. D. A. Bini and B. Meini, *Inverting block Toeplitz matrices in block Hessenberg form by means of displacement operators: Application to queueing problems*, Linear Algebr. Appl. **272** (1998), 1–16. {**xxviii, 153**}
35. D. A. Bini and V. Y. Pan, *Polynomial and matrix computations*, vol. 1: Fundamental Algorithms, Birkhäuser, 1994. {**153, 162, 173**}
36. ———, *Graeffe's, Chebyshev-like, and Cardinal's processes for splitting a polynomial into factors*, J. Complexity **12** (1996), 492–511. {**13**}
37. R. R. Bitmead and B. D. O. Anderson, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebr. Appl. **34** (1980), 103–116. {**154**}
38. P. Blanchard, *Complex analytic dynamics on the Riemann sphere*, Bulletin (New Series) of the Amer. Math. Soc. **11** (1984), no. 1, 85–141. {**135**}

39. A. W. Bojanczyk and G. Heinig, *A multi-step algorithm for Hankel matrices*, J. Complexity **10** (1994), no. 1, 142–164. {**24, 153**}
40. T. Boros, T. Kailath, and V. Olshevsky, *Predictive pivoting and backward stability of fast Cauchy solvers*, Unpublished manuscript. {**154**}
41. L. C. Botten, M. S. Craig, and R. C. McPhedran, *Complex zeros of analytic functions*, Comput. Phys. Commun. **29** (1983), no. 3, 245–259. {**xiv, 6**}
42. T. Boult and K. Sikorski, *An optimal complexity algorithm for computing the topological degree in two dimensions*, SIAM J. Sci. Stat. Comput. **10** (1989), no. 4, 686–698. {**11**}
43. R. Brent, F. Gustavson, and D. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms **1** (1980), no. 3, 259–295. {**154**}
44. A. G. Buckley, *Conversion of Fortran 90: a case study*, ACM Trans. Math. Software **20** (1994), no. 3, 308–353. {**38**}
45. A. Bultheel and M. Van Barel, *Euclid, Padé and Lanczos: another golden braid*, Report TW 188, Department of Computer Science, K.U.Leuven, April 1993. {**16**}
46. ———, *Formal orthogonal polynomials for arbitrary moment matrix and Lanczos type methods*, Proceedings of the Cornelius Lanczos International Centenary Conference (Philadelphia, PA) (J. D. Brown, M. T. Chu, D. C. Ellison, and R. J. Plemmons, eds.), Society for Industrial and Applied Mathematics, 1994, pp. 273–275. {**16**}
47. ———, *Linear algebra, rational approximation and orthogonal polynomials*, Studies in Computational Mathematics, vol. 6, North-Holland, 1997. {**xvii, 16, 20**}
48. E. E. Burniston and C. E. Siewert, *The use of Riemann problems in solving a class of transcendental equations*, Proc. Camb. Philos. Soc. **73** (1973), 111–118. {**13**}
49. S. Cabay and D. Choi, *Algebraic computations of scaled Padé fractions*, SIAM J. Comput. **15** (1986), 243–270. {**155**}
50. S. Cabay, M. H. Gutknecht, and R. Meleshko, *Stable rational interpolation?*, Systems and Networks: Mathematical Theory and Applications. Volume II: Invited and Contributed Papers (U. Helmke, R. Mennicken, and J. Saurer, eds.), Mathematical Research, vol. 79, Akademie Verlag, 1994, pp. 631–634. {**139**}
51. S. Cabay and R. Meleshko, *A weakly stable algorithm for Padé approximants and the inversion of Hankel matrices*, SIAM J. Matrix Anal. Appl. **14** (1993), no. 3, 735–765. {**24, 153**}
52. J.-P. Cardinal, *On two iterative methods for approximating the roots of a polynomial*, The Mathematics of Numerical Analysis (Providence, Rhode Island) (J. Renegar, M. Shub, and S. Smale, eds.), Lectures in Applied Mathematics, vol. 32, 1995 AMS-SIAM Summer Seminar in Applied Mathematics, July 17–August 11, 1995, Park City, Utah, American Mathematical Society, 1996, pp. 165–188. {**13**}
53. J.-P. Cardinal and B. Mourrain, *Algebraic approach of residues and applications*, The Mathematics of Numerical Analysis (Providence, Rhode Island) (J. Renegar, M. Shub, and S. Smale, eds.), Lectures in Applied Mathematics, vol. 32, 1995 AMS-SIAM Summer Seminar in Applied Mathematics, July 17–August 11, 1995, Park City, Utah, American Mathematical Society, 1996, pp. 189–210. {**85**}
54. M. P. Carpentier and A. F. Dos Santos, *Solution of equations involving analytic functions*, J. Comput. Phys. **45** (1982), 210–220. {**6, 7**}
55. C. Carstensen and T. Sakurai, *Simultaneous factorization of a polynomial by rational approximation*, J. Comput. Appl. Math. **61** (1995), no. 2, 165–178. {**13**}
56. Ll. G. Chambers, *An upper bound for the first zero of Bessel functions*, Math. Comput. **38** (1982), no. 158, 589–591. {**98**}
57. T. F. Chan and P. C. Hansen, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Process. **40** (1992), no. 5, 1079–1090. {**153**}
58. ———, *A look-ahead Levinson algorithm for indefinite Toeplitz systems*, SIAM J. Matrix Anal. Appl. **13** (1992), no. 2, 490–506. {**153**}

59. G. Claessens, *A new algorithm for osculatory rational interpolation*, Numer. Math. **27** (1976), 77–83. {**139**}
60. ———, *A generalization of the QD algorithm*, J. Comput. Appl. Math. **7** (1981), no. 4, 237–247. {**139**}
61. A. Córdova, W. Gautschi, and S. Ruscheweyh, *Vandermonde matrices on the circle: spectral properties and conditioning*, Numer. Math. **57** (1990), no. 6/7, 577–591. {**17**}
62. M. Crampin and F. A. E. Pirani, *Applicable differential geometry*, London Mathematical Society Lecture Note Series, vol. 59, Cambridge University Press, Cambridge, 1986. {**84**}
63. B. Davies, *Locating the zeros of an analytic function*, J. Comput. Phys. **66** (1986), 36–49. {**7**}
64. F. de Hoog, *A new algorithm for solving Toeplitz systems of equations*, Linear Algebr. Appl. **88/89** (1987), 123–138. {**155**}
65. D. W. Decker, H. B. Keller, and C. T. Kelley, *Convergence rates for Newton's method at singular points*, SIAM J. Numer. Anal. **20** (1983), no. 2, 296–314. {**124**}
66. D. W. Decker and C. T. Kelley, *Newton's method at singular points: I*, SIAM J. Numer. Anal. **17** (1980), no. 1, 66–70. {**124**}
67. ———, *Newton's method at singular points: II*, SIAM J. Numer. Anal. **17** (1980), no. 3, 465–471. {**124**}
68. ———, *Convergence acceleration for Newton's method at singular points*, SIAM J. Numer. Anal. **19** (1981), no. 1, 219–229. {**124**}
69. ———, *Expanded convergence domains for Newton's method at nearly singular roots*, SIAM J. Sci. Stat. Comput. **6** (1985), no. 4, 951–966. {**124**}
70. L. M. Delves and J. N. Lyness, *A numerical method for locating the zeros of an analytic function*, Math. Comput. **21** (1967), 543–560. {**xiii, 5**}
71. A. Draux, *Polynômes orthogonaux formels—applications*, Lecture Notes in Mathematics, vol. 974, Springer, 1983. {**xv, 14, 16**}
72. ———, *Formal orthogonal polynomials revisited. Applications*, Numer. Algorithms **11** (1996), 143–158. {**xv, 14**}
73. Ö. Eğecioğlu and Ç. K. Koç, *A fast algorithm for rational interpolation via orthogonal polynomials*, Math. Comput. **53** (1989), no. 187, 249–264. {**68, 139**}
74. I. Z. Emiris and J. F. Canny, *Efficient incremental algorithms for the sparse resultant and the mixed volume*, J. Symbolic Computation **20** (1995), no. 2, 117–149. {**85**}
75. K. Engelborghs, T. Luzyanina, and D. Roose, *Bifurcation analysis of periodic solutions of neutral functional differential equations: A case study*, Int. J. Bifurcation Chaos **8** (1998), no. 10, 1889–1905. {**34**}
76. P. J. Erdelsky, *Computing the Brouwer degree in* $\mathbb{R}^2$, Math. Comput. **27** (1973), no. 121, 133–137. {**84**}
77. A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi, *Bateman manuscript project. Higher transcendental functions*, vol. 2, McGraw-Hill, 1953. {**115–118**}
78. J. C. Faugère, P. Gianni, D. Lazard, and T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symbolic Computation **16** (1993), no. 4, 329–344. {**85**}
79. M. Fiedler, *Hankel and Loewner matrices*, Linear Algebr. Appl. **58** (1984), 75–95. {**xxix, 155, 156, 167**}
80. T. Finck, G. Heinig, and K. Rost, *An inversion formula and fast algorithms for Cauchy-Vandermonde matrices*, Linear Algebr. Appl. **183** (1993), 179–191. {**160, 189**}
81. G. S. Fishman, *Monte Carlo: Concepts, algorithms, and applications*, Springer, 1996. {**93**}
82. P. Fitzpatrick, *On the scalar rational interpolation problem*, Math. Control Signals Systems **9** (1996), 352–369. {**139**}
83. H. Flanders, *Differential forms with applications to the physical sciences*, Dover, New York, 1989. {**84**}

84. R. W. Freund, *A look-ahead Bareiss algorithm for general Toeplitz matrices*, Numer. Math. **68** (1994), 35–69. {**153**}
85. R. W. Freund and H. Zha, *Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems*, Linear Algebr. Appl. **188/189** (1993), 255–303. {**153**}
86. _____, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math. **64** (1993), 295–321. {**24, 153**}
87. F. D. Gakhov, *Boundary value problems*, International Series of Monographs in Pure and Applied Mathematics, vol. 85, Pergamon Press, Oxford, 1966. {**13**}
88. K. Gallivan, S. Thirumalai, and P. Van Dooren, *A block Toeplitz look-ahead Schur algorithm*, SVD and Signal Processing III (Amsterdam) (M. Moonen and B. De Moor, eds.), Elsevier, 1995, pp. 199–206. {**154**}
89. K. A. Gallivan, S. Thirumalai, and P. Van Dooren, *A look-ahead Schur algorithm*, Proceedings of the Fifth SIAM Conference on Applied Linear Algebra (Snowbird, Utah), June 1994, pp. 450–454. {**153**}
90. K. A. Gallivan, S. Thirumalai, P. Van Dooren, and V. Vermaut, *High performance algorithms for Toeplitz and block Toeplitz matrices*, Linear Algebr. Appl. **241–243** (1996), 343–388. {**154**}
91. W. Gautschi, *On inverses of Vandermonde and confluent Vandermonde matrices*, Numer. Math. **4** (1962), 117–123. {**17**}
92. _____, *On inverses of Vandermonde and confluent Vandermonde matrices. II*, Numer. Math. **5** (1963), 425–430. {**17**}
93. _____, *On the construction of Gaussian quadrature rules from modified moments*, Math. Comput. **24** (1970), no. 110, 245–260. {**xvi, 18**}
94. _____, *Optimally conditioned Vandermonde matrices*, Numer. Math. **24** (1975), no. 1, 1–12. {**17**}
95. _____, *On inverses of Vandermonde and confluent Vandermonde matrices. III*, Numer. Math. **29** (1978), 445–450. {**17**}
96. _____, *On generating orthogonal polynomials*, SIAM J. Sci. Stat. Comput. **3** (1982), no. 3, 289–317. {**xvi, 18**}
97. _____, *On the sensitivity of orthogonal polynomials to perturbations in the moments*, Numer. Math. **48** (1986), 369–382. {**xvi, 18**}
98. _____, *How (un)stable are Vandermonde systems?*, Asymptotic and Computational Analysis: Conference in Honor of Frank W. J. Olver's 65th Birthday (R. Wong, ed.), Lecture Notes in Pure and Applied Mathematics, vol. 124, Marcel Dekker, 1990, pp. 193–210. {**17, 91**}
99. W. Gautschi and G. Inglese, *Lower bounds for the condition number of Vandermonde matrices*, Numer. Math. **52** (1988), 241–250. {**17**}
100. L. Gemignani, *Rational interpolation via orthogonal polynomials*, Computers Math. Applic. **26** (1993), no. 5, 27–34. {**68, 139**}
101. _____, *Chebyshev rational interpolation*, Numerical Algorithms **15** (1997), 91–110. {**139**}
102. _____, *Schur complements of Bezoutians and the inversion of block Hankel and block Toeplitz matrices*, Linear Algebr. Appl. **253** (1997), 39–59. {**155**}
103. B. Gleyse and V. Kaliaguine, *On algebraic computation of number of poles of meromorphic functions in the unit disk*, Nonlinear Numerical Methods and Rational Approximation II (A. Cuyt, ed.), Kluwer Academic Publishers, 1994, pp. 241–246. {**78**}
104. I. Gohberg, T. Kailath, and I. Koltracht, *Efficient solution of linear systems of equations with recursive structure*, Linear Algebr. Appl. **80** (1986), 81–113. {**154**}
105. I. Gohberg, T. Kailath, I. Koltracht, and P. Lancaster, *Linear complexity parallel algorithms for linear systems of equations with recursive structure*, Linear Algebr. Appl. **88/89** (1987), 271–315. {**154**}

106. I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comput. **64** (1995), no. 212, 1557–1576. {**154**}
107. ———, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comput. **64** (1995), no. 212, 1557–1576. {**154**}
108. I. Gohberg and I. Koltracht, *Mixed, componentwise, and structured condition numbers*, SIAM J. Matrix Anal. Appl. **14** (1993), no. 3, 688–704. {**17**}
109. I. Gohberg and V. Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations Operator Theory **20** (1994), 44–83. {**154**}
110. G. Golub and V. Olshevsky, *Pivoting on structured matrices with applications*, Unpublished manuscript. {**154**}
111. G. H. Golub, P. Milanfar, and J. Varah, *A stable numerical method for inverting shape from moments*, Unpublished manuscript. {**17**}
112. W. B. Gragg, F. D. Gustavson, D. D. Warner, and D. Y. Y. Yun, *On fast computation of superdiagonal Padé fractions*, Math. Program. Stud. **18** (1982), 39–42. {**155**}
113. W. B. Gragg and M. H. Gutknecht, *Stable look-ahead versions of the Euclidean and Chebyshev algorithms*, Approximation and Computation: A Festschrift in Honor of Walter Gautschi (R. V. M. Zahar, ed.), Birkhäuser, 1994, pp. 231–260. {**xv, 14, 16, 153**}
114. P. R. Graves-Morris, *Practical, reliable, rational interpolation*, J. Inst. Maths Applics **25** (1980), 267–286. {**139**}
115. ———, *Efficient reliable rational interpolation*, Padé Approximation and its Applications (M. G. de Bruin and H. van Rossum, eds.), Lecture Notes in Mathematics, vol. 888, Springer, 1981, pp. 28–63. {**139**}
116. ———, *Symmetrical formulas for rational interpolants*, J. Comput. Appl. Math. **10** (1984), 107–111. {**139**}
117. P. R. Graves-Morris and T. R. Hopkins, *Reliable rational interpolation*, Numer. Math. **36** (1981), 111–128. {**139**}
118. A. Gray and G. B. Mathews, *A treatise on Bessel functions and their applications to physics*, Dover, 1966. {**98**}
119. A. Griewank, *Starlike domains of convergence for Newton's method at singularities*, Numer. Math. **35** (1980), 95–111. {**124**}
120. ———, *On solving nonlinear equations with simple singularities or nearly singular solutions*, SIAM Review **27** (1985), no. 4, 537–563. {**124**}
121. A. Griewank and M. R. Osborne, *Newton's method for singular problems when the dimension of the null space is* > 1, SIAM J. Numer. Anal. **18** (1981), no. 1, 145–149. {**124**}
122. ———, *Analysis of Newton's method at irregular singularities*, SIAM J. Numer. Anal. **20** (1983), no. 4, 747–773. {**124**}
123. M. Gutknecht and M. Hochbruck, *Optimized look-ahead recurrences for adjacent rows in the Padé table*, BIT **36** (1996), 264–286. {**153**}
124. M. H. Gutknecht, *Continued fractions associated with the Newton-Padé table*, Numer. Math. **56** (1989), 547–589. {**67, 68, 139**}
125. ———, *In what sense is the rational interpolation problem well posed?*, Constr. Approx. **6** (1990), no. 4, 437–450. {**68, 139**}
126. ———, *The rational interpolation problem revisited*, Rocky Mt. J. Math. **21** (1991), no. 1, 263–280. {**68, 139**}
127. ———, *Block structure and recursiveness in rational interpolation*, Approximation Theory VII (E. W. Cheney, C. K. Chui, and L. L. Schumaker, eds.), Academic Press, 1992, pp. 93–130. {**68, 139**}
128. ———, *A completed theory of the unsymmetric Lanczos process and related algorithms, Part I*, SIAM J. Matrix Anal. Appl. **13** (1992), no. 2, 594–639. {**xv, 14, 20**}

129. ———, *Stable row recurrences for the Padé table and a generically superfast look-ahead solver for non-Hermitian Toeplitz systems*, Linear Algebr. Appl. **188/189** (1993), 351–421. {**153, 155**}
130. ———, *A completed theory of the unsymmetric Lanczos process and related algorithms, Part II*, SIAM J. Matrix Anal. Appl. **15** (1994), no. 1, 15–58. {**xv, 14, 20**}
131. ———, *The multipoint Padé table and general recurrences for rational interpolation*, Nonlinear Numerical Methods and Rational Approximation II (A. Cuyt, ed.), Kluwer Academic Publishers, 1994, pp. 109–135. {**139**}
132. M. H. Gutknecht and M. Hochbruck, *Look-ahead Levinson- and Schur-type recurrences in the Padé table*, Electronic Transactions on Numerical Analysis **2** (1994), 104–129. {**153, 155**}
133. ———, *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems*, Numer. Math. **70** (1995), 181–228. {**153, 155**}
134. G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, Linear Algebra in Signal Processing, IMA volumes in Mathematics and its Applications, vol. 69, IMA, 1994, pp. 95–114. {**154, 189**}
135. ———, *Solving Toeplitz systems via extension and interpolation*, CALCOLO **33** (1996), 115–129, Proceedings of the workshop *Toeplitz Matrices: Structure, Algorithms and Applications*. Cortona (Italy), September 9–12, 1996. {**154, 189**}
136. ———, *Solving Toeplitz systems via tangential Lagrange interpolation*, Submitted to *SIAM J. Matrix. Anal. Appl.*, 1996. {**149**}
137. ———, *Transformation approaches for fast and stable solution of Toeplitz systems and polynomial equations*, Proceedings of the International Workshop "Recent Advances in Applied Mathematics" (State of Kuwait), May 4–7 1996, pp. 223–238. {**154, 189**}
138. G. Heinig and A. Bojanczyk, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices: I. Transformations*, Linear Algebr. Appl. **254** (1997), 193–226. {**154, 189**}
139. ———, *Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices: II. Algorithms*, Linear Algebr. Appl. **278** (1998), no. 1–3, 11–36. {**154, 189**}
140. G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Operator Theory: Advances and Applications, vol. 13, Birkhäuser, 1984. {**154, 158, 181**}
141. P. Henrici, *Applied and computational complex analysis: I. Power series—integration—conformal mapping—location of zeros*, Wiley, 1974. {**8, 9**}
142. J. Herlocker and J. Ely, *An automatic and guaranteed determination of the number of roots of an analytic function interior to a simple closed curve in the complex plane*, Reliable Computing **1** (1995), no. 3, 239–250. {**11**}
143. N. J. Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl. **11** (1990), no. 1, 23–41. {**154**}
144. B. J. Hoenders and C. H. Slump, *On the calculation of the exact number of zeros of a set of equations*, Computing **30** (1983), no. 2, 137–147. {**99**}
145. ———, *On the determination of the number and multiplicity of zeros of a function*, Computing **47** (1992), 323–336. {**99**}
146. A. Hoy, *A relation between Newton and Gauss-Newton steps for singular nonlinear equations*, Computing **40** (1988), 19–27. {**124**}
147. ———, *An efficiently implementable Gauss-Newton-like method for solving singular nonlinear equations*, Computing **41** (1989), 107–122. {**124**}
148. A. Hoy and H. Schwetlick, *Some superlinearly convergent methods for solving singular nonlinear equations*, Computational Solution of Nonlinear Systems of Equations (Providence, Rhode Island) (E. L. Allgower and K. Georg, eds.), Lectures in Applied Mathematics, vol. 26, 1988 SIAM-AMS Summer Seminar on Computational Solution of Nonlinear Systems of Equations, July 18-29, 1988, Fort Collins, Colorado, American Mathematical Society, 1990, pp. 285–300. {**124**}

149. V. Hribernig and H. J. Stetter, *Detection and validation of clusters of polynomial zeros*, J. Symbolic Computation **24** (1997), no. 6, 667–681. {**57**}
150. T. Huckle, *A look-ahead algorithm for solving nonsymmetric linear Toeplitz equations*, Proceedings of the Fifth SIAM Conference on Applied Linear Algebra (Snowbird, Utah), June 1994, pp. 455–459. {**153**}
151. E. K. Ifantis and C. G. Kokologiannaki, *Location of the complex zeros of Bessel functions and Lommel polynomials*, Zeitschrift für Analysis und ihre Anwendungen **12** (1993), 605–612. {**98**}
152. E. K. Ifantis and P. D. Siafarikas, *Ordering relations between the zeros of miscellaneous Bessel functions*, Appl. Anal. **23** (1986), 85–110. {**98**}
153. ———, *A differential inequality for the positive zeros of Bessel functions*, J. Comput. Appl. Math. **44** (1992), 115–120. {**98**}
154. Y. Ikebe, Y. Kikuchi, and I. Fujishiro, *Computing zeros and orders of Bessel functions*, J. Comput. Appl. Math. **38** (1991), 169–184. {**98**}
155. Y. Ikebe, Y. Kikuchi, I. Fujishiro, N. Asai, K. Takanashi, and M. Harada, *The eigenvalue problem for infinite compact complex symmetric matrices with application to the numerical computation of complex zeros of $J_0(z) - iJ_1(z)$ and of Bessel functions $J_m(z)$ of any real order m*, Linear Algebr. Appl. **194** (1993), 35–70. {**98**}
156. N. I. Ioakimidis, *Application of the generalized Siewert-Burniston method to locating zeros and poles of meromorphic functions*, J. Appl. Math. Phys. **36** (1985), 733–742. {**78**}
157. ———, *Determination of poles of sectionally meromorphic functions*, J. Comput. Appl. Math. **15** (1986), 323–327. {**78**}
158. ———, *Quadrature methods for the determination of zeros of transcendental functions—a review*, Numerical Integration: Recent Developments, Software and Applications (P. Keast and G. Fairweather, eds.), Reidel, Dordrecht, The Netherlands, 1987, pp. 61–82. {**xiii, 5**}
159. ———, *A unified Riemann-Hilbert approach to the analytical determination of zeros of sectionally analytic functions*, J. Math. Anal. Appl. **129** (1988), no. 1, 134–141. {**13**}
160. ———, *A note on the closed-form determination of zeros and poles of generalized analytic functions*, Stud. Appl. Math. **81** (1989), 265–269. {**13**}
161. N. I. Ioakimidis and E. G. Anastasselou, *A modification of the Delves-Lyness method for locating the zeros of analytic functions*, J. Comput. Phys. **59** (1985), 490–492. {**7**}
162. ———, *A new, simple approach to the derivation of exact analytical formulae for the zeros of analytic functions*, Appl. Math. Comput. **17** (1985), 123–127. {**13**}
163. ———, *On the simultaneous determination of zeros of analytic or sectionally analytic functions*, Computing **36** (1986), 239–247. {**13**}
164. S. Joe and I. H. Sloan, *Implementation of a lattice method for numerical multiple integration*, ACM Trans. Math. Softw. **19** (1993), 523–545. {**93**}
165. ———, *Corrigendum*, ACM Trans. Math. Softw. **20** (1994), no. 2, 245. {**93**}
166. E. Jonckheere and C. Ma, *A simple Hankel interpretation of the Berlekamp-Massey algorithm*, Linear Algebr. Appl. **125** (1989), 65–76. {**16**}
167. M. Kac, W. L. Murdock, and G. Szegő, *On the eigen-values of certain Hermitian forms*, J. Rational Mech. Anal. **2** (1953), 767–800. {**164**}
168. S. W. Kahng, *Osculatory interpolation*, Math. Comput. **23** (1969), 621–629. {**139**}
169. T. Kailath and V. Olshevsky, *Diagonal pivoting for partially reconstructible Cauchy-like matrices, with applications to Toeplitz-like linear equations and to boundary rational matrix interpolation problems*, Linear Algebr. Appl. **254** (1997), 251–302. {**154**}
170. T. Kailath and A. H. Sayed, *Displacement structure: theory and applications*, SIAM Review **37** (1995), 297–386. {**153, 154**}
171. M. H. Kalos and P. A. Whitlock, *Monte Carlo methods. Volume I: Basics*, Wiley, 1986. {**93**}

172. D. J. Kavvadias and M. N. Vrahatis, *Locating and computing all the simple roots and extrema of a function*, SIAM J. Sci. Comput. **17** (1996), no. 5, 1232–1248. {**84**}
173. H. B. Keller, *Geometrically isolated nonisolated solutions and their approximation*, SIAM J. Numer. Anal. **18** (1981), no. 5, 822–838. {**124**}
174. C. T. Kelley and R. Suresh, *A new acceleration method for Newton's method at singular points*, SIAM J. Numer. Anal. **20** (1983), no. 5, 1001–1009. {**124**}
175. M. K. Kerimov and S. L. Skorokhodov, *Calculation of the multiple zeros of the derivatives of the cylindrical Bessel functions $J_\nu(z)$ and $Y_\nu(z)$*, U.S.S.R. Comput. Maths. Math. Phys. **25** (1985), no. 6, 101–107. {**xxiii, 98**}
176. ———, *Multiple complex zeros of derivatives of the cylindrical Bessel functions*, Sov. Phys. Dokl. **33** (1988), no. 3, 196–198. {**xxiii, 98**}
177. R. F. King, *Improving the Van de Vel root-finding method*, Computing **30** (1983), 373–378. {**xxv, 123, 131, 134**}
178. P. Kirrinnis, *Newton iteration towards a cluster of polynomial zeros*, Foundations of Computational Mathematics (F. Cucker and M. Shub, eds.), Springer, 1997, pp. 193–215. {**57**}
179. C. G. Kokologiannaki and P. D. Siafarikas, *Non-existence of complex and purely imaginary zeros of transcendental equation involving Bessel functions*, Zeitschrift für Analysis und ihre Anwendungen **10** (1991), no. 4, 563–567. {**98**}
180. C. G. Kokologiannaki, P. D. Siafarikas, and C. B. Kouris, *On the complex zeros of $H_\mu(z)$, $J'_\mu(z)$, $J''_\mu(z)$ for real or complex order*, J. Comput. Appl. Math. **40** (1992), 337–344. {**98**}
181. G. Kowalewski, *Interpolation und genäherte Quadratur*, Teubner, 1932. {**156**}
182. S. G. Krantz, *Function theory of several complex variables*, Wiley, 1982. {**86**}
183. P. Kravanja, R. Cools, and A. Haegemans, *Computing zeros of analytic mappings: A logarithmic residue approach*, BIT **38** (1998), no. 3, 583–596. {**xxi, 83**}
184. P. Kravanja and A. Haegemans, *A modification of Newton's method for analytic mappings having multiple zeros*, To appear in *Computing*. {**xxv, 123**}
185. P. Kravanja, O. Ragos, M. N. Vrahatis, and F. A. Zafiropoulos, *ZEBEC: A mathematical software package for computing simple zeros of Bessel functions of real order and complex argument*, Comput. Phys. Commun. **113** (1998), no. 2–3, 220–238. {**xxii, 35, 97, 111**}
186. P. Kravanja, T. Sakurai, and M. Van Barel, *On locating clusters of zeros of analytic functions*, Report TW 280, K.U.Leuven, Dept. Computer Science, July 1998. {**xiii, xix, 5, 57**}
187. P. Kravanja and M. Van Barel, *A fast block Hankel solver based on an inversion formula for block Loewner matrices*, CALCOLO **33** (1996), no. 1–2, 147–164, Proceedings of the workshop *Toeplitz Matrices: Structure, Algorithms and Applications*, Cortona (Italy), September 9–12, 1996. {**xxvii, xxix, 139, 155**}
188. ———, *A derivative-free algorithm for computing zeros of analytic functions*, Report TW 285, K.U.Leuven, Dept. Computer Science, November 1998. {**xiii, 5**}
189. ———, *A fast Hankel solver based on an inversion formula for Loewner matrices*, Linear Algebr. Appl. **282** (1998), no. 1–3, 275–295. {**xxvii, xxix, 139, 155**}
190. P. Kravanja, M. Van Barel, and A. Haegemans, *Logarithmic residue based methods for computing zeros of analytic functions and related problems*, Submitted to *Hellenic European Research on Computer Mathematics and its Applications* (E. A. Lipitakis, ed.), Proceedings of the HERCMA '98 Conference, Athens (Greece), September 24–26, 1998. {**xiii, 5**}
191. ———, *On computing zeros and poles of meromorphic functions*, To appear in the proceedings of the International Conference *Computational Methods and Function Theory* (CMFT'97) (N. Papamichael, St. Ruscheweyh and E. B. Saff, eds.), Nicosia (Cyprus), October 13–17, 1997. {**xx, 77**}
192. P. Kravanja, M. Van Barel, O. Ragos, M. N. Vrahatis, and F. A. Zafiropoulos, *ZEAL: A mathematical software package for computing zeros of analytic functions*, Submitted to *Comput. Phys. Commun.* {**xiii, 5**}

193. P. Kravanja and P. Verlinden, *On the zeros of* $J_n(z) \pm iJ_{n+1}(z)$ *and* $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$, Submitted to *IMA J. Appl. Math.* {**xxiv, 113**}
194. R. Kumar, *A fast algorithm for solving a Toeplitz system of equations*, IEEE Trans. Acoust. Speech Signal Process. **33** (1985), 254–267. {**155**}
195. P. Kunkel, *Efficient computation of singular points*, IMA J. Numer. Anal. **9** (1989), 421–433. {**124**}
196. G. Labahn and S. Cabay, *Matrix Padé fractions and their computation*, SIAM J. Comput. **18** (1989), 639–657. {**155**}
197. D. P. Laurie, *Periodizing transformations for numerical integration*, J. Comput. Appl. Math. **66** (1996), no. 1–2, 337–344. {**93**}
198. N. N. Lebedev, *Special functions and their applications*, Dover, 1972. {**98**}
199. T.-Y. Li, *On locating all zeros of an analytic function within a bounded domain by a revised Delves/Lyness method*, SIAM J. Numer. Anal. **20** (1983), no. 4, 865–871. {**7**}
200. ———, *Numerical solutions of multivariate polynomial systems by homotopy continuation methods*, Acta Numerica, vol. 6, Cambridge University Press, 1997, pp. 399–436. {**85**}
201. N. G. Lloyd, *Degree theory*, Cambridge Tracts in Mathematics, vol. 73, Cambridge University Press, 1978. {**11, 99, 101**}
202. K. Loewner, *Über monotone Matrixfunktionen*, Math. Z. **38** (1934), 177–216. {**155**}
203. J. N. Lyness and L. M. Delves, *On numerical contour integration round a closed contour*, Math. Comput. **21** (1967), 561–577. {**28**}
204. D. A. MacDonald, *The roots of* $J_0(z) - iJ_1(z) = 0$, Quart. Appl. Math. **XLVII** (1989), no. 2, 375–378. {**xxiv, 113, 119**}
205. ———, *The zeros of* $J_1^2(\zeta) - J_0(\zeta)J_2(\zeta) = 0$ *with an application to swirling flow in a tube*, SIAM J. Appl. Math. **51** (1991), no. 1, 40–48. {**xxiv, 113**}
206. ———, *On the computation of zeroes of* $J_n(z) - iJ_{n+1}(z) = 0$, Quart. Appl. Math. **LV** (1997), no. 4, 623–633. {**xxiv, 113, 114, 118, 119**}
207. H. Maehly and Ch. Witzgall, *Tschebycheff-Approximationen in kleinen Intervallen II*, Numer. Math. **2** (1960), 293–307. {**139**}
208. J. M. McNamee, *A bibliography on roots of polynomials*, J. Comput. Appl. Math. **47** (1993), 391–394. {**13**}
209. ———, *A supplementary bibliography on roots of polynomials*, J. Comput. Appl. Math. **78** (1997), no. 3, 1. {**13**}
210. Z. Mei, *A special extended system and a Newton-like method for simple singular nonlinear equations*, Computing **45** (1990), 157–167. {**124**}
211. H.-G. Meier, *Diskrete und kontinuierliche Newton-Systeme im Komplexen*, Ph.D. thesis, Rheinisch-Westfälische Technische Hochschule Aachen, 1991. {**135**}
212. J. Meinguet, *On the solubility of the Cauchy interpolation problem*, Approximation Theory (A. Talbot, ed.), Academic Press, 1970, pp. 137–163. {**68, 139**}
213. B. Meini, *Fast algorithms for the numerical solution of structured Markov chains*, Ph.D. thesis, Università degli Studi di Pisa, Dipartimento di Matematica, 1997. {**xxviii, 153**}
214. M. Morf, *Doubling algorithms for Toeplitz and related equations*, Proc. IEEE Internat. Conf. on Acoustics, Speech and Signal Processing (Denver, CO), 1980, pp. 954–959. {**154**}
215. A. P. Morgan, A. J. Sommese, and C. W. Wampler, *Computing singular solutions to nonlinear analytic systems*, Numer. Math. **58** (1991), 669–684. {**124**}
216. ———, *Computing singular solutions to polynomial systems*, Adv. Appl. Math. **13** (1992), no. 3, 305–327. {**124**}
217. ———, *A power series method for computing singular solutions to nonlinear analytic systems*, Numer. Math. **63** (1992), 391–409. {**124**}

218. B. Mourrain and V. Y. Pan, *Solving special polynomial systems by using structured matrices and algebraic residues*, Foundations of Computational Mathematics (F. Cucker and M. Shub, eds.), Springer, 1997, pp. 287–304. {**85**}
219. M. E. Muldoon, *Approximate distribution density of zeros of Bessel functions*, Europhys. Lett. **20** (1992), no. 1, 1–5. {**98**}
220. ———, *Electrostatics and zeros of Bessel functions*, J. Comput. Appl. Math. **65** (1995), 299–308. {**xxii, 98**}
221. B. R. Musicus, *Levinson and fast Cholesky algorithms for Toeplitz and almost Toeplitz matrices*, Report, Res. Lab. of Electronics, M.I.T., 1984. {**155**}
222. R. Narasimhan, *Analysis on real and complex manifolds*, Advanced Studies in Pure Mathematics, vol. 1, North-Holland, Amsterdam, 1968. {**84**}
223. M. Z. Nashed and X. Chen, *Convergence of Newton-like methods for singular operator equations using outer inverses*, Numer. Math. **66** (1993), 235–257. {**124**}
224. B. Neta and H. D. Victory, *A higher order method for determining nonisolated solutions of a system of nonlinear equations*, Computing **32** (1984), 163–166. {**124**}
225. T. Ojika, *Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations*, J. Math. Anal. Appl. **123** (1987), 199–221. {**124**}
226. ———, *Modified deflation algorithm for the solution of singular problems. II. Nonlinear multipoint boundary value problems*, J. Math. Anal. Appl. **123** (1987), 222–237. {**124**}
227. T. Ojika, S. Watanabe, and T. Mitsui, *Deflation algorithm for the multiple roots of a system of nonlinear equations*, J. Math. Anal. Appl. **96** (1983), 463–479. {**124**}
228. T. O'Neil and J. W. Thomas, *The calculation of the topological degree by quadrature*, SIAM J. Numer. Anal. **12** (1975), no. 5, 673–680. {**84**}
229. V. Y. Pan, *Solving a polynomial equation: some history and recent progress*, SIAM Review **39** (1997), no. 2, 187–220. {**13**}
230. J. R. Partington, *An introduction to Hankel operators*, London Mathematical Society Student Texts, vol. 13, Cambridge University Press, 1988. {**16**}
231. H.-O. Peitgen, M. Prüfer, and K. Schmitt, *Global aspects of the continuous and discrete Newton method: A case study*, Acta Applicandae Mathematicae **13** (1988), 123–202. {**135**}
232. M. S. Petković, *Inclusion methods for the zeros of analytic functions*, Computer Arithmetic and Enclosure Methods (L. Atanassova and J. Herzberger, eds.), North-Holland, 1992, pp. 319–328. {**13**}
233. M. S. Petković, C. Carstensen, and M. Trajkovíc, *Weierstrass formula and zero-finding methods*, Numer. Math. **69** (1995), 353–372. {**12**}
234. M. S. Petković and D. Herceg, *Higher-order iterative methods for approximating zeros of analytic functions*, J. Comput. Appl. Math. **39** (1992), 243–258. {**12**}
235. M. S. Petković and Z. M. Marjanović, *A class of simultaneous methods for the zeros of analytic functions*, Comput. Math. Appl. **22** (1991), no. 10, 79–87. {**13**}
236. R. Piessens, *Rational approximations for zeros of Bessel functions*, J. Comput. Phys. **42** (1981), no. 2. {**98**}
237. ———, *Chebyshev series approximations for the zeros of the Bessel functions*, J. Comput. Phys. **53** (1984), no. 1. {**98**}
238. ———, *A series expansion for the first positive zero of the Bessel functions*, Math. Comput. **42** (1984), no. 165, 195–197. {**98**}
239. ———, *Approximation for the turning points of Bessel functions*, J. Comput. Phys. **64** (1986), no. 1. {**98**}
240. ———, *On the computation of zeros and turning points of Bessel functions*, Bulletin of the Greek Mathematical Society **31** (1990), 117–122. {**xxii, 98**}

241. R. Piessens, E. de Doncker-Kapenga, C. W. Überhuber, and D. K. Kahaner, *QUADPACK: A subroutine package for automatic integration*, Springer Series in Computational Mathematics, vol. 1, Springer, 1983. {**xviii, xxiii, 35, 37, 100, 102, 103**}
242. R. Piessens and G. Engelen, *The computation of zeros and turning points of the Bessel functions of the first kind*, Report TW 72, K.U.Leuven, Dept. Computer Science, March 1985. {**xxii, 98**}
243. O. Ragos, M. N. Vrahatis, and F. A. Zafiropoulos, *The topological degree for the computation of the exact number of equilibrium points of dynamical systems*, Hellenic European Research on Mathematics and Informatics '94 (HERMIS '94). Proceedings of the second Hellenic European Conference on Mathematics and Informatics, Athens (Greece), September 22–24, 1994 (E. A. Lipitakis, ed.), 1994, pp. 533–542. {**84**}
244. A. D. Rawlins, *Note on the roots of* $f(z) = J_0(z) - iJ_1(z)$, Quart. Appl. Math. **47** (1989), no. 2, 323–324. {**113**}
245. G. W. Reddien, *On Newton's method for singular problems*, SIAM J. Numer. Anal. **15** (1978), no. 5, 993–996. {**124**}
246. ———, *Newton's method and high order singularities*, Computers Math. Applic. **5** (1979), 79–86. {**124**}
247. L. Reichel, *Newton interpolation at Leja points*, BIT **30** (1990), 332–346. {**154**}
248. T. Sakurai, T. Torii, N. Ohsako, and H. Sugiura, *A method for finding clusters of zeros of analytic function*, Special Issues of Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM). Issue 1: Numerical Analysis, Scientific Computing, Computer Science, 1996, Proceedings of the International Congress on Industrial and Applied Mathematics (ICIAM/GAMM 95), Hamburg, July 3–7, 1995, pp. 515–516. {**74**}
249. H. E. Salzer, *Note on osculatory rational interpolation*, Math. Comput. **16** (1962), 486–491. {**139**}
250. A. H. Sayed and T. Kailath, *A look-ahead block Schur algorithm for Toeplitz-like matrices*, SIAM J. Matrix Anal. Appl. **16** (1995), no. 2, 388–414. {**154**}
251. A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu, *Recursive solutions of rational interpolation problems via fast matrix factorization*, Integr. Equat. Oper. Th. **20** (1994), 84–118. {**139**}
252. C. Schneider and W. Werner, *Some new aspects of rational interpolation*, Math. Comput. **47** (1986), no. 175, 285–299. {**139**}
253. J. Segura and A. Gil, *ELF and GNOME: Two tiny codes to evaluate the real zeros of the Bessel functions of the first kind for real orders*, Unpublished manuscript. {**xxiii, 98**}
254. A. Sidi, *A new variable transformation for numerical integration*, Numerical Integration IV (H. Brass and G. Hämmerlin, eds.), International Series of Numerical Mathematics, vol. 112, Birkhäuser Verlag, 1993, pp. 359–373. {**93**}
255. I. H. Sloan and S. Joe, *Lattice methods for multiple integration*, Clarendon, Oxford, 1994. {**93**}
256. V. I. Smirnov and N. A. Lebedev, *Functions of a complex variable: Constructive theory*, Iliffe Books, 1968. {**12**}
257. H. Stahl, *Existence and uniqueness of rational interpolants with free and prescribed poles*, Approximation Theory, Tampa (E. B. Saff, ed.), Lecture Notes in Mathematics, vol. 1287, Springer, 1987, pp. 180–208. {**139**}
258. ———, *Convergence of rational interpolants*, Numerical Analysis: A numerical analysis conference in honour of Jean Meinguet (A. Bultheel, A. Magnus, and P. Van Dooren, eds.), Belgian Mathematical Society, December 1996, pp. 11–32. {**139**}
259. G. W. Stewart, *Perturbation theory for the generalized eigenvalue problem*, Recent Advances in Numerical Analysis (C. De Boor and G. H. Golub, eds.), Academic Press, 1978, pp. 193–206. {**60**}

260. J. Stoer, *Über zwei Algorithmen zur Interpolation mit rationalen Funktionen*, Numer. Math. **3** (1961), 285–304. {**139**}
261. Y. Sugiyama, *An algorithm for solving discrete-time Wiener-Hopf equations based on Euclid's algorithm*, IEEE Trans. Inf. Theory **32** (1986), 394–409. {**155**}
262. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, *A new method for solving key equations for decoding Goppa codes*, Internat. J. Control **27** (1975), 87–99. {**154**}
263. C. E. Synolakis, *The runup of solitary waves*, J. Fluid Mech. **185** (1987), 523–545. {**xxiv, 113**}
264. ———, *On the roots of* $f(z) = J_0(z) - iJ_1(z)$, Quart. Appl. Math. **46** (1988), no. 1, 105–107. {**xxiv, 113**}
265. O. Szász, *Inequalities concerning ultraspherical polynomials and Bessel functions*, Proc. Amer. Math. Soc. **1** (1950), no. 2, 256–267. {**116**}
266. S. Tadepalli and C. E. Synolakis, *Roots of* $J_\gamma(z) \pm iJ_{\gamma+1}(z) = 0$ *and the evaluation of integrals with cylindrical function kernels*, Quart. Appl. Math. **LII** (1994), no. 1, 103–112. {**xxiv, 113, 115, 119**}
267. N. M. Temme, *Special functions : An introduction to the classical functions of Mathematical Physics*, Wiley, 1996. {**xxiii, 98**}
268. I. J. Thompson and A. R. Barnett, *Modified Bessel functions* $I_\nu(z)$ *and* $K_\nu(z)$ *of real order and complex argument, to selected accuracy*, Comput. Phys. Commun. **47** (1987), 245–257. {**xxiii, 102, 103**}
269. J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-based complexity*, Academic Press, 1988. {**11**}
270. T. Tsuchiya, *Enlargement procedure for resolution of singularities at simple singular solutions of nonlinear equations*, Numer. Math. **52** (1988), 401–411. {**124**}
271. E. E. Tyrtyshnikov, *How bad are Hankel matrices?*, Numer. Math. **67** (1994), 261–269. {**18, 91**}
272. F. v. Haeseler and H.-O. Peitgen, *Newton's method and complex dynamical systems*, Acta Applicandae Mathematicae **13** (1988), 3–58. {**135**}
273. W. Van Assche, *Orthogonal polynomials in the complex plane and on the real line*, Special Functions, q-Series and Related Topics (Providence, Rhode Island) (M. E. H. Ismail, D. R. Masson, and M. Rahman, eds.), Fields Institute Communications, vol. 14, American Mathematical Society, Providence, Rhode Island, 1997, pp. 211–245. {**27**}
274. M. Van Barel and A. Bultheel, *A new approach to the rational interpolation problem*, J. Comput. Appl. Math. **32** (1990), 281–289. {**xxvii, 70, 72, 139, 140**}
275. ———, *A general module theoretic framework for vector M-Padé and matrix rational interpolation*, Numer. Algorithms **3** (1992), 451–462. {**139, 142, 145, 146**}
276. ———, *The "look-ahead" philosophy applied to matrix rational interpolation problems*, Systems and networks: Mathematical theory and applications, Volume II: Invited and contributed papers (U. Helmke, R. Mennicken, and J. Saurer, eds.), Mathematical Research, vol. 79, Akademie Verlag, 1994, pp. 891–894. {**146**}
277. ———, *A lookahead algorithm for the solution of block Toeplitz systems*, Linear Algebr. Appl. **266** (1997), 291–335. {**154**}
278. M. Van Barel, G. Heinig, and P. Kravanja, *A stabilized superfast solver for indefinite Toeplitz systems*, In preparation. {**xxvii, xxix, 139, 155**}
279. M. Van Barel and P. Kravanja, *A stabilized superfast solver for indefinite Hankel systems*, Linear Algebr. Appl. **284** (1998), no. 1–3, 335–355, Special issue on the International Linear Algebra Society Symposium "Linear Algebra in Control Theory, Signals and Image Processing," held at the University of Manitoba, Canada, 6–8 June 1997. {**xxvii, xxix, 139, 155**}

280. M. Van Barel and Z. Vavřín, *Inversion of a block Löwner matrix*, J. Comput. Appl. Math. **69** (1996), 261–284. {**xxx, 158, 159, 169–171**}
281. H. Van de Vel, *A method for computing a root of a single nonlinear equation, including its multiplicity*, Computing **14** (1975), 167–171. {**xxv, 123, 131, 134**}
282. P. Van Hentenryck, D. McAllester, and D. Kapur, *Solving polynomial systems using a branch and prune approach*, SIAM J. Numer. Anal. **34** (1997), no. 2, 797–827. {**85**}
283. M. Vander Straeten and H. Van de Vel, *Multiple root-finding methods*, J. Comput. Appl. Math. **40** (1992), no. 1, 105–114. {**xxv, 123, 131**}
284. Z. Vavřín, *Inverses of Löwner matrices*, Linear Algebr. Appl. **63** (1984), 227–236. {**xxx, 159**}
285. P. Verlinden, *An asymptotic estimate of Hilb's type for generalized Jacobi polynomials on the unit circle*, Technical report TW 260, K.U.Leuven, Dept. Computer Science, 1997. {**115**}
286. J. Verschelde and R. Cools, *Polynomial homotopy continuation, A portable Ada software package*, The Ada-Belgium Newsletter **4** (1996), 59–83. {**85**}
287. C. Von Westenholz, *Differential forms in mathematical physics*, Studies in Mathematics and its Applications, vol. 3, North-Holland, Amsterdam, 1978. {**84**}
288. M. N. Vrahatis, *Algorithm 666. CHABIS: A mathematical software package for locating and evaluating roots of systems of nonlinear equations*, ACM Trans. Math. Softw. **14** (1988), no. 4, 330–336. {**xxiii, 101–103**}
289. ———, *Solving systems of nonlinear equations using the nonzero value of the topological degree*, ACM Trans. Math. Softw. **14** (1988), no. 4, 312–329. {**xxiii, 101**}
290. M. N. Vrahatis, T. N. Grapsa, O. Ragos, and F. A. Zafiropoulos, *On the localization and computation of zeros of Bessel functions*, Z. angew. Math. Mech. **77** (1997), no. 6, 467–475. {**101**}
291. M. N. Vrahatis, O. Ragos, T. Skiniotis, F. A. Zafiropoulos, and T. N. Grapsa, *RFSFNS: A portable package for the numerical determination of the number and the calculation of roots of Bessel functions*, Comput. Phys. Commun. **92** (1995), 252–266. {**xxii, 98**}
292. ———, *The topological degree theory for the localization and computation of complex zeros of Bessel functions*, Numer. Funct. Anal. and Optimiz. **18** (1997), no. 1 & 2, 227–234. {**101**}
293. M. N. Vrahatis, O. Ragos, F. A. Zafiropoulos, and T. N. Grapsa, *Locating and computing zeros of Airy functions*, Z. angew. Math. Mech. **76** (1996), no. 7, 419–422. {**101**}
294. H. Wallin, *Potential theory and approximation of analytic functions by rational interpolation*, Complex Analysis, Joensuu 1978 (I. Laine, O. Lehto, and T. Sorvali, eds.), Lecture Notes in Mathematics, vol. 747, Springer, 1979, pp. 434–450. {**139**}
295. G. N. Watson, *A treatise on the theory of Bessel functions*, second ed., Cambridge University Press, 1966. {**xxiii, 53, 98, 114, 116, 118**}
296. H. Weber and W. Werner, *On the accurate determination of nonisolated solutions of nonlinear equations*, Computing **26** (1981), 315–326. {**124**}
297. H. Werner, *A reliable method for rational interpolation*, Padé Approximation and its Applications (L. Wuytack, ed.), Lecture Notes in Mathematics, vol. 765, Springer, 1979, pp. 257–277. {**139**}
298. ———, *Ein Algorithmus zur rationalen Interpolation*, Numerical Methods of Approximation Theory: Volume 5 (L. Collatz, H. Meinardus, and H. Werner, eds.), International Series of Numerical Mathematics, vol. 52, Birkhäuser, 1980, pp. 319–337. {**139**}
299. ———, *Algorithm 51: A reliable and numerically stable program for rational interpolation in Lagrange data*, Computing **31** (1983), 269–286. {**139**}
300. J. H. Wilkinson, *The evaluation of the zeros of ill-conditioned polynomials. Part I*, Numer. Math. **1** (1959), 150–166. {**6**}
301. L. Wuytack, *An algorithm for rational interpolation similar to the qd-algorithm*, Numer. Math. **20** (1973), 418–424. {**139**}

302. _____, *On some aspects of the rational interpolation problem*, SIAM J. Numer. Anal. **11** (1974), no. 1, 52–60. {**139**}
303. _____, *On the osculatory rational interpolation problem*, Math. Comput. **29** (1975), no. 131, 837–843. {**139**}
304. P. Wynn, *Über einen Interpolations-algorithmus und gewisse andere Formeln die in der Theorie der Interpolation durch rationale Funktionen bestehen*, Numer. Math. **2** (1960), 151–182. {**139**}
305. J.-Cl. Yakoubsohn, *Approximating the zeros of analytic functions by the exclusion algorithm*, Numerical Algorithms **6** (1994), 63–88. {**13**}
306. X. Ying, *A reliable root solver for automatic computation with application to stress analysis of a composite plane wedge*, D. Sc. dissertation, Washington University in St. Louis, 1986. {**9, 56**}
307. X. Ying and I. N. Katz, *A reliable argument principle algorithm to find the number of zeros of an analytic function in a bounded domain*, Numer. Math. **53** (1988), 143–163. {**9, 11**}
308. _____, *A simple reliable solver for all the roots of a nonlinear function in a given domain*, Computing **41** (1989), no. 4, 317–333. {**13**}
309. T. J. Ypma, *Finding a multiple zero by transformations and Newton-like methods*, SIAM Review **25** (1983), no. 3, 365–378. {**124**}

Index

Curriculum Vitae

Education:

- PhD student in Computer Science, K.U.Leuven, entered fall 1994.
  Advisors: Prof. Dr. ir. Ann Haegemans and Prof. Dr. ir. Marc Van Barel.
- "Burgerlijk Ingenieur" in Computer Science, K.U.Leuven, 1994 (80.0%).
  This is a 5-year programme. The degree is considered equivalent to the degree of Master of Science in Engineering (US) or Diplom Ingenieur (Germany).
  Thesis: "A Szegő Theory for Rational Matrix Functions" at the Department of Computer Science, Division of Numerical Analysis and Applied Mathematics. *Advisor*: Prof. Dr. Adhemar Bultheel.

Grants:

- Research assistant at the K.U.Leuven, Department of Computer Science, Division of Numerical Analysis and Applied Mathematics, October 1, 1998 – September 30, 1999.
- A PhD research grant from the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry, October 1, 1994 – September 30, 1998.

Research Interests:

- *Computational complex analysis*: logarithmic residue based methods for computing all solutions to systems of analytic equations; computing zeros of analytic functions; zeros of Bessel functions.
- *Structured numerical linear algebra*: fast and superfast methods for solving systems of linear equations that have low displacement rank (for example, Toeplitz or Hankel systems); rational interpolation.
- *Continuous Newton methods.*

Conferences that I attended:

- *HERCMA 98: Hellenic European Research on Computer Mathematics and its Applications*, September 24–26, 1998, Athens University of Economics and Business, Athens, Greece.
- *ICCAM 98: International Congress on Computational and Applied Mathematics*, July 27 – August 1, 1998, Katholieke Universiteit Leuven, Leuven, Belgium.

- *IWOP '98: International Workshop on Orthogonal Polynomials*, June 29–July 2, 1998, Universidad Carlos III de Madrid, Leganés, Madrid, Spain.
- *FAA98: Fourier Analysis and Applications*, May 3–6, 1998, Kuwait University, Kuwait.
- *Numerical Methods in Systems and Control*, January 19, 1998, a 1-day meeting organized by the Interuniversity Pole of Attraction "Modelling, Identification, Simulation and Control of Complex Systems."
- *Computer Arithmetic and Numerical Techniques*, October 23, 1997, Antwerp, Belgium. Colloquium organized by the Scientific Research Network "Advanced Numerical Methods for Mathematical Modelling."
- *Computational Methods and Function Theory '97*, October 13–17, 1997, Nicosia, Cyprus.
- *Sommerschule über Nichtlineare Gleichungssysteme*, March 17–21, 1997, TU Hamburg-Harburg, Germany.
- *Constructive Complex Analysis*, November 28, 1996, Leuven, Belgium.
- *Toeplitz Matrices: Structure, Algorithms and Applications*, September 9–12, 1996, Cortona, Italy.
- *ICCAM 96: International Congress on Computational and Applied Mathematics*, July 21–26, 1996, Katholieke Universiteit Leuven, Leuven, Belgium.
- *AMS-BeNeLux Congress*, May 22–25, 1996, Antwerp, Belgium.
- *Special Functions and their Applications*, a 1-day meeting on Special Functions, Orthogonal Polynomials and Rational Approximation in honour of Prof. André Ronveaux, February 1, 1996, Leuven, Belgium.
- *Special Topics in Numerical Analysis and Applied Mathematics*, a 1-day conference in honour of Prof. Jean Meinguet, December 1, 1995, Louvain-la-Neuve, Belgium.
- *SEA-95: Résolution des Systèmes d'Equations Algébriques*, November 13–17, 1995, Luminy, Marseille, France.
- *Workshop Rational Approximation*, October 17, 1995, Leuven, Belgium.

Conference talks:

- P. Kravanja, T. Sakurai and M. Van Barel, *On locating clusters of zeros of analytic functions*, Hellenic European Research on Computer Mathematics and its Applications (HERCMA 98), September 24–26, 1998, Athens, Greece.
- P. Kravanja, M. Van Barel and T. Sakurai, *A method for locating clusters of zeros of analytic functions*, Annual meeting of Japan SIAM, September 12–14, 1998, Tokyo, Japan. (In Japanese)
- P. Kravanja, T. Sakurai and M. Van Barel, *On locating clusters of zeros of analytic functions*, International Congress on Computational and Applied Mathematics (ICCAM '98), July 27 – August 1, 1998, Katholieke Universiteit Leuven, Leuven, Belgium.

- M. Van Barel, G. Heinig and P. Kravanja, *Stabilized superfast Toeplitz solvers via rational interpolation*, Minisymposium "Numerical Methods for Structured Matrices and Applications," SIAM Annual Meeting 1998, July 13–17, 1998, University of Toronto, Toronto, Canada.
- P. Kravanja, M. Van Barel and T. Sakurai, *A method for locating clusters of zeros of analytic functions*, GAMM-Workshop on Iterative Processes for Solving Equations, July 3–5, 1998, Mathematisches Seminar, Christian-Albrechts-Universität zu Kiel, Kiel, Germany.
- P. Kravanja, T. Sakurai and M. Van Barel, *On locating clusters of zeros of analytic functions*, International Workshop on Orthogonal Polynomials (IWOP '98), June 29–July 2, 1998, Universidad Carlos III de Madrid, Leganés, Madrid, Spain.
- M. Van Barel, G. Heinig and P. Kravanja, *Using Fast Fourier Transformation techniques to solve Toeplitz systems in a superfast way*, FAA98: Fourier Analysis and Applications, May 3–6, 1998, Kuwait University, Kuwait.
- P. Kravanja, T. Sakurai and M. Van Barel, *On locating clusters of zeros of analytic functions*, April 9, 1998. Center of Research and Applications of Nonlinear Systems, University of Patras, Patras, Greece.
- P. Kravanja and A. Haegemans, *A modification of Newton's method for analytic mappings having multiple zeros*, April 2, 1998. Center of Research and Applications of Nonlinear Systems, University of Patras, Patras, Greece.
- M. Van Barel, G. Heinig and P. Kravanja, *A superfast algorithm to solve Toeplitz systems*, Sixth SIAM Conference on Applied Linear Algebra, October 30, 1997, Snowbird, Utah, U.S.A.
- P. Kravanja and A. Haegemans, *A modification of Newton's method for analytic mappings having multiple zeros*, Computer Arithmetic and Numerical Techniques, October 23, 1997, Antwerp, Belgium. Colloquium organized by the Scientific Research Network "Advanced Numerical Methods for Mathematical Modelling."
- P. Kravanja, M. Van Barel and A. Haegemans, *On computing zeros of analytic functions*, Computational Methods and Function Theory '97, October 13–17, 1997, Nicosia, Cyprus.
- P. Kravanja, M. Van Barel and A. Haegemans, *On computing zeros of analytic functions*, General Seminar of Mathematics, June 18, 1997. Department of Mathematics, University of Patras, Patras, Greece.
- M. Van Barel and P. Kravanja, *A superfast algorithm to solve Hankel systems*, International Linear Algebra Society Symposium on Fast Algorithms for Control and Image Processing, June 6–8, 1997. Institute of Industrial Mathematical Sciences, University of Manitoba, Winnipeg, Canada.
- P. Kravanja and M. Van Barel, *A fast Hankel solver based on an inversion formula for Loewner matrices*, DEA Course MAPA3011: Special Topics in

Numerical Linear Algebra, November 18, 1996. Institut de Mathématique Pure et Appliquée, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

- P. Kravanja and M. Van Barel, *A fast Hankel solver based on a transformation technique*, The ROLLS final meeting, November 2–3, 1996, Leipzig, Germany.
- P. Kravanja and M. Van Barel, *A fast Hankel solver based on an inversion formula for Loewner matrices*, Toeplitz Matrices: Structure, Algorithms and Applications, September 9–12, 1996, Cortona, Italy.
- P. Kravanja, R. Cools and A. Haegemans, *A cubature method for solving systems of analytic equations*, ICCAM 96: International Congress on Computational and Applied Mathematics, July 21–26, 1996, Leuven, Belgium.
- P. Kravanja and A. Haegemans, *Computing zeros of analytic functions via the qd-algorithm*, SEA-95: Résolution des Systèmes d'Equations Algébriques, November 13–17, 1995, Luminy, Marseille, France.

The talks were given by the person whose name is underlined.

Research stays:

- Research stay at the Department of Mathematics of the University of Patras (Patras, Greece). March 22–April 18, 1998. Invited by Prof. M. Vrahatis.
- Research stay at the Department of Mathematics of the University of Patras (Patras, Greece). June 2–27, 1997. Invited by Prof. M. Vrahatis.

Research Publications:

In international journals:

- P. Kravanja and A. Haegemans, A modification of Newton's method for analytic mappings having multiple zeros. To appear in Computing.
- M. Van Barel and P. Kravanja, A stabilized superfast solver for indefinite Hankel systems, Linear Algebr. Appl. **284** (1998), nr. 1–3, 335–355. Special issue on the International Linear Algebra Society Symposium "Linear Algebra in Control Theory, Signals and Image Processing," held at the University of Manitoba, Canada, 6–8 June 1997.
- P. Kravanja and M. Van Barel, A fast Hankel solver based on an inversion formula for Loewner matrices, Linear Algebr. Appl. **282** (1998), nr. 1–3, 275–295.
- P. Kravanja, O. Ragos, M. N. Vrahatis and F. A. Zafiropoulos, ZEBEC: A mathematical software package for computing simple zeros of Bessel functions of real order and complex argument, Comput. Phys. Commun. **113** (1998), nr. 2–3, 220–238.
- P. Kravanja, R. Cools and A. Haegemans, Computing zeros of analytic mappings: a logarithmic residue approach, BIT **38** (1998), nr. 3, 583–596.
- P. Kravanja and M. Van Barel, A fast block Hankel solver based on an inversion formula for block Loewner matrices, CALCOLO **33** (1996),

147–164. Proceedings of the workshop 'Toeplitz Matrices: Structure, Algorithms and Applications,' Cortona (Italy), September 9–12, 1996

In conference proceedings:

- P. Kravanja, M. Van Barel and A. Haegemans, On computing zeros and poles of meromorphic functions. To appear in the proceedings of *Computational Methods and Function Theory* (CMFT'97) (N. Papamichael, St. Ruscheweyh and E. B. Saff, eds.), Nicosia (Cyprus), October 13–17, 1997.

Submitted papers:

- P. Kravanja, M. Van Barel, O. Ragos, M. N. Vrahatis and F. A. Zafiropoulos, ZEAL: A mathematical software package for computing zeros of analytic functions. Submitted to Comput. Phys. Commun.
- P. Kravanja and M. Van Barel, A derivative-free algorithm for computing zeros of analytic functions. Submitted to Computing. Available as Technical Report TW 285, K.U.Leuven, November 1998.
- P. Kravanja and A. Haegemans, Multidimensional analytic deflation. Submitted to Am. Math. Monthly.
- P. Kravanja and P. Verlinden, On the zeros of $J_n(z) \pm iJ_{n+1}(z)$ and $[J_{n+1}(z)]^2 - J_n(z)J_{n+2}(z)$. Submitted to IMA J. Appl. Math.
- P. Kravanja, T. Sakurai and M. Van Barel, On locating clusters of zeros of analytic functions. Submitted to BIT. Available as Technical Report TW 280, K.U.Leuven, July 1998.
- P. Kravanja, M. Van Barel and A. Haegemans, Logarithmic residue based methods for computing zeros of analytic functions and related problems. Submitted to *Hellenic European Research on Computer Mathematics and its Applications* (E. A. Lipitakis, ed.), Proceedings of the HERCMA '98 Conference, Athens (Greece), September 24–26, 1998.

In preparation:

- M. Van Barel, G. Heinig and P. Kravanja, A stabilized superfast solver for indefinite Toeplitz systems.
- M. Van Barel and P. Kravanja, Generically superfast computation of Hankel determinants.
- G. Van den Eynde, P. Kravanja and R. Cools, CONE: An implementation of Diener's extended continuous Newton method for computing all solutions to systems of nonlinear equations.

I guided the following **Master's Thesis**:

- Gert Van den Eynde, Continue Newton methodes, Dept. Computer Science, K.U.Leuven, academic year 1997–1998 (in Dutch).

Teaching Experience: Numerical Analysis, Calculus.

Editorial Work:

- Editor of "The SCME Digest," an electronically distributed newsletter on Scientific Computing and Mathematical Engineering.
  Cf. `http://www.cs.kuleuven.ac.be/~scme/`

Memberships:

- ILAS: International Linear Algebra Society

E-mail: `Peter.Kravanja@na-net.ornl.gov`