

Diss. ETH Nr. 6759

NULLSTELLENBESTIMMUNG BEI POLYNOMEN
UND ALLGEMEINEN ANALYTISCHEN FUNKTIONEN ALS
ANWENDUNG DER SCHNELLEN FOURIERTRANSFORMATION

ABHANDLUNG
zur Erlangung des Titels eines
DOKTORS DER MATHEMATIK
der
EIDGENOESSISCHEN TECHNISCHEN HOCHSCHULE ZUERICH

vorgelegt von
PETER GEIGER
Dipl. Math. ETH
geboren am 4. April 1948
von Ermatingen TG

Angenommen auf Antrag von
Prof. Dr. Peter Henrici, Referent
Prof. Dr. Jürg T. Marti, Korreferent

1981

ZUSAMMENFASSUNG

Im ersten Teil wird die diskrete Fouriertransformation als Approximation der Fourierkoeffizienten diskutiert. Dann wird eine Implementation der schnellen Fouriertransformation (FFT) erarbeitet, bei welcher die Anzahl der Punkte nicht vorgegeben sein muss.

Im zweiten Teil wird dieser Algorithmus angewendet auf die Bestimmung der Nullstellen z_1, z_2, \dots einer in einem Gebiet $G \subset \mathbb{C}$ analytischen Funktion f . Diese Aufgabe wird gelöst, indem das Gebiet G mit Kreisscheiben K überdeckt und zu K dasjenige Polynom p_K konstruiert wird, welches genau die Nullstellen von f besitzt, die in K liegen. Dieser Prozess kann iteriert werden, indem K mit kleineren Kreisscheiben überdeckt wird. p_K kann mit Hilfe der Newtonschen Formeln konstruiert werden aus den Integralen

$$s_k := \frac{1}{2\pi i} \int_{\partial K} z^k \frac{f'(z)}{f(z)} dz = \sum_{z_i \in K} z_i^k, \quad k \geq 0$$

Andererseits sind die s_k gerade die Fourierkoeffizienten der Funktion $f'(e^{2\pi i t})/f(e^{2\pi i t})$, falls K die Einheitskreisscheibe ist (was immer durch eine lineare Transformation erreicht werden kann). Somit können die s_k mit Hilfe unseres FFT-Algorithmus berechnet werden. Die für den Iterationsprozess benötigte Funktion p'_K/p_K kann nun auch mit Hilfe des qd-Algorithmus direkt als Kettenbruch dargestellt werden. Der Fall, wo f ein Polynom ist, wird separat behandelt wegen der daraus resultierenden Vereinfachungen.

Als Anwendung werden die Nullstellen von verschiedenen Polynomen berechnet und die Ergebnisse mit denen anderer Nullstellenprogramme verglichen. Dann werden Nullstellen des komplexen Fehlerintegrals und verschiedener Besselfunktionen sowie die Fixpunkte der Exponentialfunktion berechnet. Im Anhang werden ALGOL-Implementationen all dieser Algorithmen vorgestellt.

ABSTRACT

In part I we review some results concerning the discrete Fourier transform as an approximation to the continuous transform. We then describe a new implementation of the Fast Fourier Transform Algorithm (FFT) which is dynamic in the sense that the number of nodes is not specified in advance.

In part II this algorithm is applied to the problem of locating the zeros z_1, z_2, \dots of a function f analytic in a region $G \subset \mathbb{C}$. The method consists in covering G by disks K and to construct, for each disk, the polynomial p_K whose zeros are precisely the zeros of f in that disk. This, in principle, is always possible by computing the integrals

$$s_k := \frac{1}{2\pi i} \int_{\partial K} z^k \frac{f'(z)}{f(z)} dz = \sum_{z_i \in K} z_i^k, \quad k \geq 0$$

from which the polynomial p_K may be found by Newton's identities. This process may then be iterated, covering the disk K by smaller disks. On the other side, if K is the unit disk (that can always be achieved by a linear transformation), the s_k are exactly the Fourier coefficients of the function $f'(e^{2\pi i t})/f(e^{2\pi i t})$ and thus may be calculated rapidly and with controllable accuracy by means of our version of FFT. From the s_k we obtain by means of the qd algorithm a continued fraction representation for p'_K/p_K which is required to continue the iteration. If f is a polynomial, deflation is used as soon as p_K is available in order to continue the process with f/p_K .

As an application of the method we compute the zeros of several polynomials whose factorisation is known to be difficult. Comparisons with other rootfinding algorithms are given. We also compute the zeros of the complex error function and various Bessel functions as well as the fixed points of the exponential function. ALGOL implementations of all algorithms are given in the appendix.