THE METHOD OF STEEPEST DESCENT APPLIED TO DETERMINE ZEROS OF POLYNOMIALS AND ADAPTED FOR USE ON THE DIGITAL COMPUTER

by

JOHN DAVID HARTMAN

B. S., Bethany College, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mathematics

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1968

Approved by:

S. Thomas Parker
Major Professor

TABLE OF CONTENTS

INTRODUCTION

The method of steepest descent can be applied in a convergent procedure to determine the zeros of polynomials having either real or complex coefficients. In many practical problems involving the solution of algebraic equations, there is sufficient a priori knowledge of the roots to permit application of the easily applied, rapidly converging synthetic division methods. The limitation of such synthetic division methods as Newton-Raphson's method, Lin's method, Bairstow's method and other synthetic division methods is that their convergence is dependent upon the initial approximations.

For problems in which the a priori information about the location of the roots is inadequate, the Lehmer-Schur method, Graeffe's root-squaring method, Bernoulli's method, and others, may be used and convergence is guaranteed. These methods are by no means simple and straightforward and are not rapidly convergent, and thus for efficiency are usually used to calculate only the approximate root locations so that one of the more rapidly converging synthetic division methods may then be applied. The method of steepest descent is a straightforward method which always converges and which has a rapid convergence in the region of the root.

The method of steepest descent can be readily applied to a digital computer with some slight modifications. The stated purpose of this paper is to show the development of the particular form of the method of steepest descent given by J. B. Moore in the "Journal of the Association for Computing Machinery", Vol. 14, No. 2, April, 1967. Many other forms of the method can be found, but Dr. Moore's form of the method is most suitable for adaptation to the digital computer through the use of Siljak functions.

Dr. Moore states the problem in the following way. Consider the equation $f(z) = u(x,y) + iv(x,y)$, where $f(z)$ is an entire function of the complex variable z, with $z = x + iy$; x and y will be considered as being functions of a parameter t. The differential equations

$$\frac{dx}{dt} = - h \frac{\partial F}{\partial x} \quad \text{and} \quad \frac{dy}{dt} = - h \frac{\partial F}{\partial y} ,$$

where h is a positive constant and $F = F(x,y)$ is some function derived from $f(z)$ satisfying the following conditions, are a basis for the algorithm:

1. F is non-negative.
2. The derivatives $\partial F/\partial x$ and $\partial F/\partial y$ exist.
3. The zeros of F are located at the roots of $f(z) = 0$.
4. These zeros are the only minima of F.

It is suggested that $F(x,y) = u^2 + v^2$ and

$$h = \frac{0.5}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

are suitable choices for these two quantities F and h. Other authors have suggested that $F(x,y) = |u| + |v|$, while the choice of h generally proves to be quite arbitrary and somewhat elusive.

The following algorithm, which arises out of theoretical computations, is the basis for a convergent method for finding the zeros of analytic functions.

(1) Scale to have some roots within the unit circle.

(2) Choose initial approximations other than on the real axis, i.e., x_0, y_0.

(3) Compute $u(x,y)$, $v(x,y)$, $\partial u/\partial x$, $\partial v/\partial x$ and $F(x,y)$ for $x = x_0$, $y = y_0$.

(4) Compute new approximations for x and y by using

$$x_{i+1} = x_i + \Delta x, \quad y_{i+1} = y_i + \Delta y,$$

where

$$\Delta x = \frac{-u\frac{\partial u}{\partial x} - v\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2} \quad \text{and} \quad \Delta y = \frac{u\frac{\partial v}{\partial x} - v\frac{\partial u}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

for $x = x_i$, $y = y_i$.

(5) Take x as x_0 and y as y_0 and recompute the parts of (3).

(6) If the value of $F(x,y)$ as now computed is greater than that of the previous iteration, reduce the increments Δx and Δy used previously until the value of F is smaller than for the preceding iteration (Dr. Moore suggests reduction by one quarter each trial).

(7) If F is sufficiently close to zero, stop. Otherwise repeat the process beginning at step (3).

With the algorithm now at hand, the theoretical development will follow next and then the application to digital computer use will be discussed.

THEORETICAL ASPECTS OF THE ALGORITHM

The first aspect of the algorithm to be considered is the pair of differential equations

$$\frac{dy}{dt} = - h \frac{\partial F}{\partial y} \qquad \text{and} \qquad \frac{dx}{dt} = - h \frac{\partial F}{\partial x} .$$

For the use in the algorithm the corresponding incremental equations

$$\Delta x = - h \frac{\partial F}{\partial x} \qquad \text{and} \qquad \Delta y = - h\frac{\partial F}{\partial y}$$

replace the differential equations. First recall the four conditions imposed upon F(x,y).

(1) F is non-negative.

(2) The derivatives $\partial F/\partial x$ and $\partial F/\partial y$ exist.

(3) The zeros of F are located at the roots of $f(z) = 0$.

(4) These zeros are the only minima of F.

Accepting Dr. Moore's choice of F, i.e., $F(x,y) = u^2 + v^2$ where $u = u(x,y)$ and $v = v(x,y)$, it is immediately apparent that the first three conditions are satisfied. To show that the fourth condition is satisfied, consider that $u^2 + v^2$ is the square of the modulus of f(z). Since the modulus of a complex number is zero if and only if the complex number is zero, the fourth condition follows immediately.

Now it will be shown that the direction of steepest descent is indeed given by the differential equations

$$\frac{dx}{dt} = - h \frac{\partial F}{\partial x} \quad \text{and} \quad \frac{dy}{dt} = - h \frac{\partial F}{\partial y}, \qquad h > 0.$$

Consider now the function F(x,y), defined above, as assigning a functional value to each point of the two dimensional Euclidean space whose coordinates are denoted by the arguments x and y, in other words, the x, y plane. The question is, at a point p(x,y), what direction of motion in the Euclidean space decreases the value of the function F(x,y) most rapidly with respect to the distance moved. This would be the direction of steepest descent.

The idea of a curve in the Euclidean space enters into the development. In particular, a curve may be thought of as generated by a moving point whose coordinates are given as functions of time. Let x = x(t), y = y(t). At any time t this curve may be approximated by a line through p(x,y) with direction numbers equal to the derivatives dx/dt and dy/dt evaluated at p(x,y). This is the tangent line which approximates this curve more closely than any other curve for points corresponding to times close to the chosen time. If the parameter s represents the length of the arc from p(x,y) to the point corresponding to a time t on the curve, then the relation between s, t, x(t) and y(t) can be expressed as

$$\frac{ds}{dt} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} .$$

The question of finding the path of steepest descent can now be posed in the following analytic formulation. Consider a curve originating at point p(x,y). Other points may be denoted either by the time parameter t or by the parameter s denoting the length of the arc of the curve joining them with p(x,y). A choice of a value of t or of s gives a point, and at each point, F(x,y) has a numeric value. Therefore the rate of change of F with respect to s along the curve at any point on the curve has meaning.

This rate of change may be evaluated by elementary calculus. It follows that

$$\frac{dx}{ds} = \frac{dx}{dt} \cdot \frac{dt}{ds} = \frac{dx}{dt} \Big/ \frac{ds}{dt} \quad \text{and} \quad \frac{dy}{ds} = \frac{dy}{dt} \Big/ \frac{ds}{dt}$$

and the rate of change of F(x,y) with respect to s may be written

$$\frac{dF}{ds} = \frac{\partial F}{\partial x}\frac{dx}{ds} + \frac{\partial F}{\partial y}\frac{dy}{ds} \quad .$$

It must be assumed that, at $t = 0$, the moving point is not stationary momentarily, so that $ds/dt \neq 0$. Note that the value of dF/ds is completely determined at any point by the values of dx/dt and dy/dt. This indicates that the rate of descent depends momentarily upon the direction of descent and not on the curvature of the curve followed. Also note that dF/ds is unchanged if dx/dt and dy/dt are multiplied by a constant.

If the quantities dx/dt and dy/dt are temporarily denoted by g and h respectively, the simple rules of calculus will give the desired direction of steepest descent. The problem can be written as asking for values of g and h which maximize dF/ds. In other words, it is necessary to solve

$$\frac{\partial\ dF/ds}{\partial g} = \frac{\partial\ dF/ds}{\partial h} = 0.$$

The solution of these equations yield

$$\frac{\partial F}{\partial x}\left[\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2\right] = \frac{dx}{dt}\left[\frac{\partial F}{\partial x}\frac{dx}{dt} + \frac{\partial F}{\partial y}\frac{dy}{dt}\right]$$

and

$$\frac{\partial F}{\partial y}\left[\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2\right] = \frac{dy}{dt}\left[\frac{\partial F}{\partial y}\frac{dy}{dt} + \frac{\partial F}{\partial x}\frac{dx}{dt}\right] .$$

Though the above two equations may not seem simple at first glance, the equations may be interpreted as meaning that a necessary condition for maximal ascent or maximal descent at $p(x,y)$ is that the derivatives dx/dt and dy/dt be proportional to the partial derivatives $\partial F/\partial x$ and $\partial F/\partial y$ respectively at the point $p(x,y)$. In other words, steepest descent occurs when $dx/dt = - k(\partial F/\partial x)$ and $dy/dt = - k(\partial F/\partial y)$ with $k > 0$.

Hence it has been shown that these differential equations can be used to determine the path of steepest descent. This direction with components $\partial F/\partial x$ and $\partial F/\partial y$ is usually called the gradient of F. Hence, the direction with components $- k(\partial F/\partial x)$ and $- k(\partial F/\partial y)$ can be considered as being the negative gradient of F. This negative gradient vector is orthogonal to a level line through $p(x,y)$ and points in the direction of steepest descent.

For the numerical evaluation of the zeros of $f(z)$, the difference equations corresponding to the above differential equations are used. These difference equations are

$$\Delta x = - h \frac{\partial F}{\partial x} \quad \text{and} \quad \Delta y = - h \frac{\partial F}{\partial y} .$$

It can be seen that the direction of steepest descent $\Delta x/\Delta y$ is independent of h, i.e.,

$$\frac{\Delta x}{\Delta y} = \frac{\partial F/\partial x}{\partial F/\partial y} .$$

The role that h plays is important, however; h should be chosen to minimize the value of F along the path of steepest descent.

This choice of h will be the next aspect of the problem considered before looking at each step of the algorithm. The choice made is dependent upon the choice for F(x,y) that Dr. Moore works with, namely, $F = u^2 + v^2$.

First consider the Taylor expansions for u(x,y) and v(x,y), in the neighborhood of a zero of f(z).

$$u(x,y) \cong u(x_0,y_0) + \frac{\partial u}{\partial x}(x - x_0) + \frac{\partial u}{\partial y}(y - y_0)$$

$$v(x,y) \cong v(x_0,y_0) + \frac{\partial v}{\partial x}(x - x_0) + \frac{\partial v}{\partial y}(y - y_0)$$

where z = x + iy is a zero of f(z). Then it is evident that

$$- u(x_0,y_0) \cong \frac{\partial u}{\partial x}\Delta x + \frac{\partial u}{\partial y}\Delta y$$

$$- v(x_0,y_0) \cong \frac{\partial v}{\partial x}\Delta x + \frac{\partial v}{\partial y}\Delta y$$

where all partial derivatives are evaluated at (x_0,y_0), and Δx and Δy are the distances in the x and y directions respectively from the point (x_0,y_0) in the neighborhood of the zero of f(z) to the point (x,y) which is a zero of f(z).

Now by using the two above equations, the forms of Δx and Δy that facilitate the algorithm and a choice of h that is appropriate can be obtained. The Cauchy-Riemann equations and the above equations derived from the Taylor expansion may be combined as follows:

$$- u = \frac{\partial u}{\partial x}\Delta x + \frac{\partial u}{\partial y}\Delta y, \qquad - v = \frac{\partial v}{\partial x}\Delta x + \frac{\partial v}{\partial y}\Delta y$$

$$- u \frac{\partial v}{\partial x} = \frac{\partial u}{\partial x}\,\frac{\partial v}{\partial x}\Delta x + \frac{\partial u}{\partial y}\,\frac{\partial v}{\partial x}\Delta y$$

$$- v \frac{\partial u}{\partial x} = \frac{\partial u}{\partial x}\,\frac{\partial v}{\partial x}\Delta x + \frac{\partial u}{\partial x}\,\frac{\partial v}{\partial y}\Delta y$$

Subtracting the respective sides of the above equation from those of the second equation above, there results:

$$- u \frac{\partial v}{\partial x} + v \frac{\partial u}{\partial x} = \left(\frac{\partial u}{\partial y}\,\frac{\partial v}{\partial x} - \frac{\partial u}{\partial x}\,\frac{\partial v}{\partial y}\right)\Delta y.$$

Using the Cauchy-Riemann equations,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \qquad \text{and} \qquad \frac{\partial u}{\partial y} = - \frac{\partial v}{\partial x},$$

results in

$$- u \frac{\partial v}{\partial x} + v \frac{\partial u}{\partial x} = -\left(\frac{\partial v}{\partial x}\,\frac{\partial v}{\partial x} - \frac{\partial u}{\partial x}\,\frac{\partial u}{\partial x}\right)\Delta y$$

or

$$\Delta y = \frac{u \frac{\partial v}{\partial x} - v \frac{\partial u}{\partial x}}{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial x}\right)^2} .$$

The similar procedure will now be carried out to obtain an expression for Δx. The two equations obtained from the Taylor expansion are multiplied by $\partial v/\partial y$ and $\partial u/\partial y$ respectively to obtain the following:

$$- u \frac{\partial v}{\partial y} = \frac{\partial u}{\partial x} \frac{\partial v}{\partial y}\Delta x + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y}\Delta y$$

$$- v \frac{\partial u}{\partial y} = \frac{\partial u}{\partial x} \frac{\partial u}{\partial y}\Delta x + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y}\Delta y$$

Subtracting yields

$$- u \frac{\partial v}{\partial y} + v \frac{\partial u}{\partial y} = \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial v}{\partial x} \frac{\partial u}{\partial y}\right) \Delta x.$$

By use of the Cauchy-Riemann equations, the following expression for Δx is obtained.

$$\Delta x = \frac{- u\frac{\partial u}{\partial x} - v\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

Now an appropriate choice for h may be obtained by using these expressions for Δx and Δy. The choice of F is, as before, $F = u^2 + v^2$. Therefore,

$$\frac{\partial F}{\partial x} = 2\ u \frac{\partial u}{\partial x} + 2\ v \frac{\partial v}{\partial x}$$

and

$$\frac{\partial F}{\partial y} = 2\ u \frac{\partial u}{\partial y} + 2\ v \frac{\partial v}{\partial y} \quad .$$

Recalling the difference equations

$$\Delta x = - h \frac{\partial F}{\partial x} \qquad \text{and} \qquad \Delta y = - h \frac{\partial F}{\partial y} ,$$

it follows that

$$\Delta x = - h \ (2u \frac{\partial u}{\partial x} + 2v \frac{\partial v}{\partial x})$$

$$\Delta y = - h \ (2u \frac{\partial u}{\partial y} + 2v \frac{\partial v}{\partial y})$$

or that

$$h = - \frac{1}{2} \ \frac{\Delta x}{u \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial x}}$$

$$h = \frac{0.5}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

Hence, a method of calculating a suitable value for h has been found. This value is in agreement with the choice that $F = u^2 + v^2$. A simple example will now be given to illustrate the use of the algorithm. Consider

$$f(z) = z^2 - z + 1/2$$

which has zeros $z = 1/2 \pm i/2$. For this polynomial,

$$u(x,y) = x^2 - y^2 - x + 1/2,$$

$$v(x,y) = 2\,xy - y,$$

recalling that $f(z) = u(x,y) + iv(x,y)$. Also,

$$\frac{\partial u}{\partial x} = 2x - 1, \qquad \frac{\partial v}{\partial x} = 2y.$$

$$\begin{aligned} F(x,y) &= u^2 + v^2 \\ &= x^4 - 2x^3 + 2x^2 - x - 3xy^2 + 3x^2y^2 + y^4 + 1/4 \end{aligned}$$

Step one of the algorithm can be neglected as there are roots within the unit circle. Step two is to choose initial approximations other than on the real axis. For this example, choose $x_0 = 0.3$ and $y_0 = 0.4$.

Step three is to compute u, v, $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial x}$ and F at the initial point.

$$\begin{aligned} u(.3, .4) &= 0.13000 \\ v(.3, .4) &= -0.16000 \\ u_x(.3, .4) &= -0.40000 \\ v_x(.3, .4) &= 0.80000 \\ F(.3, .4) &= 0.04250 \end{aligned}$$

Step four is to compute the new approximations for the x and y values, using $x_{i+1} = x_i + \Delta x$ and $y_{i+1} = y_i + \Delta y$. First, computing

$$\Delta x = \frac{-u\ u_x - v\ v_x}{u_x^2 + v_x^2} \quad \text{and} \quad \Delta y = \frac{u\ v_x - v\ u_x}{u_x^2 + v_x^2},$$

it follows that

$$\Delta x = 0.22500 \quad \text{and} \quad \Delta y = 0.05000.$$

Hence

$$x_2 = 0.52500 \quad \text{and} \quad y_2 = 0.45000.$$

Now these new approximations of $x_2 = 0.52500$ and $y_2 = 0.45000$ are used to compute new values of u, v, u_x, v_x and F, which in turn lead to new approximations x_3 and y_3. For this example,

$$u(.525, .45) = 0.04813$$

$$v(.525, .45) = 0.02250$$

$$u_x(.525, .45) = 0.65000$$

$$v_x(.525, .45) = 0.90000$$

$$F(.525, .45) = 0.02822$$

The newly computed value of $F(x_2,y_2)$ should now be compared with the value of $F(x_1,y_1)$. For this example $F(x_2,y_2) < F(x_1,y_1)$ so the algorithm may be continued in the normal manner. However, should any computed value be larger than the one for the previous iteration, the values used for Δx and Δy in the last iteration should be reduced and the iteration done again with the new x and y values. Dr. Moore suggests reducing Δx and Δy by a factor of one-fourth for such occurrences.

From the last computed values for u, v, u_x, and v_x, the new values for Δx and Δy may be computed.

$$\Delta x = 0.00287$$

$$\Delta y = -0.00193$$

From this it follows that

$$x_3 = 0.49712$$

$$y_3 = 0.50192 \quad .$$

Then it follows that

$$u(.49712, .50192) = -0.00192$$

$$v(.49712, .50192) = -0.00300$$

$$u_x(.49712,\ .50192) = -0.00577$$

$$v_x(.49712,\ .50192) = 1.00385$$

$$F(.49712,\ .50192) = 0.00001$$

Note that the new value for F is less than that for the previous iteration. This indicates that the new approximations for x and y are getting closer to the actual zeros of f(z) where $z = x + iy$. The new values for Δx and Δy are

$$\Delta x = 0.00287$$

$$\Delta y = -0.00193 \ .$$

Hence, $x_4 = 0.5000$ and $y_4 = 0.5000$. The next iteration will yield that $\Delta x = \Delta y = 0$. Hence, the zeros of $z^2 - z + 1/2$ are

$$z = 1/2 \pm i/2 \ .$$

SCALING A POLYNOMIAL

A technique will now be given to scale a polynomial so that all zeros lie within the unit circle. First of all, an upper bound for the zeros of the polynomial must be determined. There are many possible techniques for finding an upper bound for the roots of a polynomial equation; however, the following technique is very simple.

Consider that a general polynomial may be denoted as follows:

$$p(z) = a_n z^n + a_{n-1} z^{n-1} + \text{---} + a_1 z + a_0 = 0, \; a_n \neq 0.$$

Any polynomial can be transformed into an equivalent monic polynomial by dividing all terms by the leading coefficient. Hence, the polynomial denoted above is equivalent to the polynomial

$$z^n + b_{n-1} z^{n-1} + \text{---} + b_1 z + b_0 = 0 \; ,$$

where $b_k = a_k/a_n$. Once a polynomial is in monic form, an upper bound on the roots of the polynomial equation is one plus the maximum of the absolute values of the coefficients. This follows by considering the companion matrix of the polynomial

$$p(z) = z^n + b_{n-1} z^{n-1} + \text{---} + b_1 z + b_0 \; .$$

The zeros of this polynomial will be the characteristic roots of the companion matrix of the polynomial.

The companion matrix of the polynomial above is the matrix

$$\begin{bmatrix} 0 & 1 & 0 & - & - & - & 0 & 0 \\ 0 & 0 & 1 & 0 & - & - & 0 & 0 \\ - & - & - & - & - & - & - & - \\ 0 & 0 & 0 & - & - & - & 0 & 1 \\ -b_0 & -b_1 & -b_2 & - & - & - & -b_{n-2} & -b_{n-1} \end{bmatrix}$$

It will be true that any characteristic root of a matrix is less than or equal to the minimum of the maximum row sum and maximum column sum. Hence, any characteristic root must be less than the maximum column sum regardless of what the row sums are. From the form of the companion matrix of a polynomial, it then follows that a bound on the zeros of a polynomial is one plus the maximum absolute value of the coefficients of the monic polynomial.

The method for scaling the zeros of a polynomial was chosen for its simplicity and because it can be easily adapted for use on a digital computer.

An example of the technique follows. Consider the polynomial

$$p(z) = 2z^5 - 7.4z^4 + 14.8z^3 - 21.6z^2 + 20.6z - 13.6 \ .$$

First change p(z) into the equivalent monic polynomial,

$$z^5 - 3.7z^4 + 7.4z^3 - 10.8z^2 + 10.3z - 6.8 \ .$$

The maximum of the absolute values of the coefficients is 10.8. Hence, an upper bound on the zeros is 11.8. For use on the computer, 11.8 will suffice; however, to work the example by hand, 12 will be chosen as the upper bound. Let $z = 12w$. Then substitution into the polynomial yields a new polynomial of the complex variable w,

$$(12w)^5 - 3.7(12w)^4 + 7.4(12w)^3 - 10.8(12w)^2 + 10.3(12w) - 6.8 \ .$$

This new polynomial in w is equal to

$$248{,}832w^5 - 76723.2w^4 + 12787.2w^3 - 1555.2w^2 + 129.6w - 6.8 \ .$$

While this new polynomial might look quite formidable, the computer can handle it readily.

ADAPTATION OF THE ALGORITHM FOR USE ON THE DIGITAL COMPUTER

Now consider the problem of adapting the method of steepest descent for efficient use on the digital computer. It would not be entirely practical to use the algorithm as given previously on page 3 of this report, unless one has a very special case to consider. It can be done, however. As an example of the straightforward use of the given algorithm on a digital computer, the program written in FORTRAN IV language for the example

$$f(z) = z^2 - z + 1/2$$

is given on the following page.

This program will readily solve the particular equation

$$f(z) = z^2 - z + 1/2 = 0 \ ,$$

but it is useless for solving any other polynomial with real or complex coefficients. What is needed is a way to write a program which will readily find the zeros of any polynomial with real or complex coefficients by simply supplying the coefficients and the degree of the polynomial. This type program would have a great deal of usefulness to anyone with the problem of finding the zeros of many polynomials, or to a person with a single polynomial to solve even though he has a very limited knowledge of programming and computer usage.

Let the general polynomial with real or complex coefficients be denoted by

$$f(z) = \sum_{k=0}^{n} (a_k + ib_k)z^k$$

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(25),Y(25),U(25),V(25),DX(25),DY(25)
      DIMENSION UX(25),VX(25),F(25)
  100 FORMAT(' ',4F14.8)
  101 FORMAT('0',4F14.8)
  102 FORMAT('1',7X,'X',12X,'Y',12X,'DX',12X,'DY')
      X(1)=.3
      Y(1)=.4
      DO 10 I=1,5
      U(I)=X(I)*X(I)-Y(I)*Y(I)-X(I)+.5
      V(I)=2*X(I)*Y(I)-Y(I)
      UX(I)=2*X(I)-1.0
      VX(I)=2*Y(I)
      F(I)=U(I)**2+V(I)**2
      DX(I)=(-U(I)*UX(I)-V(I)*VX(I))/(UX(I)**2+VX(I)**2)
      X(I+1)=DX(I)+X(I)
      DY(I)=(U(I)*VX(I)-V(I)*UX(I))/(UX(I)**2+VX(I)**2)
   10 Y(I+1)=DY(I)+Y(I)
      WRITE(3,102)
      WRITE(3,101) X(1),Y(1),DX(1),DY(1)
      DO 11 I=2,5
   11 WRITE(3,100) X(I),Y(I),DX(I),DY(I)
      WRITE(3,103)
      WRITE(3,101) U(1),V(1),F(1)
      DO 12 I=2,5
   12 WRITE(3,100) U(I),V(I),F(I)
  103 FORMAT(' ',/////7X,'U',12X,'V',12X,'F')
      STOP
      END
```

```
       X             Y             DX            DY

  0.30000001    0.39999998    0.22500000    0.05000003
  0.52500001    0.45000000   -0.02788463    0.05192307
  0.49711538    0.50192308    0.00287354   -0.00192762
  0.49998892    0.49999546    0.00001108    0.00000454
  0.50000000    0.50000000   -0.00000000    0.00000000




       U             V             F

  0.13000001   -0.15999998    0.04250000
  0.04812500    0.02250001    0.00282227
 -0.00191845   -0.00289571    0.00001207
  0.00000454   -0.00001108    0.00000000
  0.00000000    0.00000000    0.00000000
```

where a_k and b_k are real numbers. It is easily seen that f(z) as defined above is an entire function of the complex variable $z = x + iy$.

By the use of Siljak functions, X_k and Y_k, the algorithm can now be adapted for a general use on the digital computer. Define the Siljak functions X_k and Y_k by the following equation:

$$z^k = X_k + iY_k \ .$$

Hence X_k is the real part of the complex number z^k and Y_k is the imaginary part. Note that $X_0 = 1$, $Y_0 = 0$, $X_1 = x$ and $Y_1 = y$ where $z = x + iy$.

Consider the identity $z^2 - 2xz + (x^2 + y^2) = 0$, noting that

$$z^2 - 2xz + x^2 + y^2 = (x + iy)^2 - 2x(x + iy) + x^2 + y^2$$

$$= x^2 + 2ixy - y^2 - 2x^2 - 2ixy + x^2 + y^2 = 0 \ .$$

Hence, it is also true that $z^k(z^2 - 2xz + x^2 + y^2) = 0$. It follows that $z^{k+2} - 2xz^{k+1} + (x^2 + y^2)z = 0$, or writing this in terms of Siljak functions and equating real and imaginary coefficients,

$$X_{k+2} = 2xX_{k+1} - (x^2 + y^2)X_k$$

$$Y_{k+2} = 2xY_{k+1} - (x^2 + y^2)Y_k \ .$$

Thus a recurrence relationship has been found for the Siljak functions. This means that given any particular value for z, the Siljak functions X_k and Y_k may be easily calculated for all values of k = 0, 1, 2, ..., n, for any desired integer n. This recurrence relationship can be very efficiently used and computed on any digital computer.

The partial derivatives of the Siljak functions with respect to the real variable x are easily obtained from the defining equation $z^k = X_k + iY_k$. Consider

$$\frac{\partial}{\partial x} z^k = kz^{k-1} \frac{\partial z}{\partial x} .$$

Since $\partial z/\partial x = 1$, it follows that

$$\frac{\partial}{\partial x} z^k = kz^{k-1} = k(X_{k-1} + iY_{k-1}) .$$

This says that

$$\frac{\partial}{\partial x} (X_k + iY_k) = k(X_{k-1} + iY_{k-1})$$

which yields, upon equating real and imaginary coefficients, that

$$\frac{\partial X_k}{\partial x} = kX_{k-1}$$

$$\frac{\partial Y_k}{\partial x} = kY_{k-1} .$$

It can be shown that in general,

$$\frac{\partial^n X_k}{\partial x^n} = k(k-1) \ldots (k-n+1)X_{k-n}$$

and

$$\frac{\partial^n Y_k}{\partial x^n} = k(k-1) \ldots (k-n+1)Y_{k-n} .$$

It now follows that

$$f(z) = \sum_{k=0}^{n} (a_k + ib_k)z^k$$

$$= \sum_{k=0}^{n} (a_k + ib_k)(X_k + iY_k)$$

$$= \sum_{k=0}^{n} [(a_kX_k - b_kY_k) + i(b_kX_k + a_kY_k)] .$$

Comparing f(z), written as above, with $f(z) = u + iv$, where $u = u(x,y)$ and $v = v(x,y)$, yields

$$u(x,y) = \sum_{k=0}^{n} (a_kX_k - b_kY_k) \quad \text{and} \quad v(x,y) = \sum_{k=0}^{n} (a_kY_k + b_kX_k) .$$

Using the rules of calculus and the ideas developed above about the partial derivatives of the Siljak functions with respect to x, the following expressions for $\partial u/\partial x$ and $\partial v/\partial x$ can be derived.

$$\frac{\partial u}{\partial x} = \sum_{k=0}^{n} k(a_kX_{k-1} - b_kY_{k-1})$$

$$\frac{\partial v}{\partial x} = \sum_{k=0}^{n} k(a_kY_{k-1} + b_kX_{k-1})$$

Using the above two expressions for $\partial u/\partial x$ and $\partial v/\partial x$, the need of directly finding derivatives of any functions has been eliminated. The fact that no derivatives need be calculated makes the algorithm simpler than when Siljak functions are not utilized.

The algorithm can now be restated in terms of the Siljak functions:

(1). Scale to have some roots within the unit circle.

(2). Choose initial approximations other than on the real axis.

(3a). Compute the Siljak functions X_k and Y_k for $k = 0, 1, \ldots, n$, where n is the degree of the polynomial f(z).

(3). Compute u(x,y), v(x,y), $\partial u/\partial x$, $\partial v/\partial x$, and F(x,y) using the formulas

$$u(x,y) = \sum_{k=0}^{n} (a_k X_k - b_k Y_k)$$

$$v(x,y) = \sum_{k=0}^{n} (a_k Y_k + b_k X_k)$$

$$\frac{\partial u}{\partial x} = \sum_{k=0}^{n} k(a_k X_{k-1} - b_k Y_{k-1})$$

$$\frac{\partial v}{\partial x} = \sum_{k=0}^{n} k(a_k Y_{k-1} + b_k X_{k-1})$$

$$F = u^2 + v^2$$

(4). Compute new approximations from the relationships

$$x_{i+1} = x_i + \Delta x, \qquad y_{i+1} = y_i + \Delta y$$

where

$$\Delta x = \frac{-\,u\frac{\partial u}{\partial x} - v\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}\,, \qquad \Delta y = \frac{u\frac{\partial v}{\partial x} - v\frac{\partial u}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

(5). Take x_1 as x_0 and y_1 as y_0 and repeat steps 3a, 3, 4, 5 and 6.

(6). If the value of F is greater than the value calculated for F on the previous iteration, reduce the values of Δx and Δy used previously until the value for F is smaller than on the preceding iteration.

This algorithm differs from the one given on page two in only two respects. First, an additional step, (3a), has been added. Secondly, in (3), the manner in which the functions u, v, $\partial u/\partial x$ and $\partial v/\partial x$ are calculated has been changed. Step (3) in the new algorithm achieves the same results as the corresponding step in the first algorithm. It is only the method of getting the results that has been changed. It is this second method of computing u, v, $\partial u/\partial x$ and $\partial v/\partial x$ that is well suited for use on a digital computer.

A program written in FORTRAN IV for use on a digital computer follows. This particular program was run on an IBM 360/50 digital computer. The program is quite simple to use and requires only a very limited knowledge of computers. The program is designed to find the zeros of any polynomial and is limited only by the accuracy of the machine in use. Better and quicker results could undoubtedly be attained if double precision were used; however, double precision routines for operating with complex numbers are not available for the machine used. The results, for the particular polynomial solved as an example, are given in tabular form following the program itself.

To use the program a few data cards must be prepared by the user. The actual number of data cards needed is three more than the degree of the polynomial to be solved. Three types of information must be read into the machine by these data cards. The first data card contains a parameter to specify the degree of the polynomial. This parameter, called k, is an interger one greater than the degree of the polynomial. The parameter k is read in by an I3 format statement. This means that the number k appears in the first three columns of the first data card with the last digit in the third column.

The next k data cards will contain the coefficients of the polynomial beginning with constant term, next the coefficient of the first power of the variable, continuing up to the coefficient of the variable to its highest power. Each coefficient is punched onto one data card in two F10.6 fields. The first F10.6 field is the real part of the coefficient and the second F10.6 field is the imaginary part.

The last data card will contain the guessed initial approximation for the zero of the polynomial. This number is read into the machine in the same manner as are the coefficients. The real part of this initial approximation is read in on the first F10.6 field and the imaginary part on the second F10.6 field.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 SA(100),SB(100)
      COMPLEX*16      ZZN(100),CZZN(100),ZN(100),ZZ(100)
      COMPLEX*8 SCOEF(100)
      COMPLEX*16 COMP(100),COEF(100),BOMP(100),Z
      DIMENSION G(100),H(100),XDX(100),YDY(100),FF(100)
      DIMENSION X(100),Y(100),XSJ(100),YSJ(100),A(100),B(100)
      DIMENSION PRD(100),AA(100),BB(100),AN(100),BN(100)
  100 FORMAT(I3)
  101 FORMAT(2F10.6)
  200 FORMAT(' ',7X,'I',8X,'X(I)',17X,'Y(I)')
  201 FORMAT(' ',5X,I3,2X,D15.8,10X,D15.8)
  404 FORMAT(' ',13X,'METHOD OF STEEPEST DESCENT')
  444 FORMAT(' ',//,11X,'FOR FINDING ZEROS OF POLYNOMIALS')
  203 FORMAT(' ',7X,'I',8X,'A(I)',17X,'B(I)')
  305 FORMAT (' ',10X,////)
  306 FORMAT(' ',10X,'COEFFICIENTS OF REDUCED POLYNOMIAL')
  505 FORMAT('1',///)
  506 FORMAT(' ',//)
  777 FORMAT(' ',D15.8)
      WRITE(3,505)
      WRITE(3,404)
      WRITE(3,444)
      READ(1,100)K
      DO 10 I=1,K
   10 READ(1,101) A(I),B(I)
      READ(1,101) X(1),Y(1)
  998 J=0.0
      L=0
      WRITE(3,506)
      WRITE(3,200)
      WRITE(3,506)
      XSJ(1)=1.0
      XSJ(2)=X(1)
      YSJ(1)=0.0
      YSJ(2)=Y(1)
      P=X(1)
      Q=Y(1)
  999 KK=K-2
      DO 11 I=1,KK
      XSJ(I+2)=2*P*XSJ(I+1)-(P*P+Q*Q)*XSJ(I)
   11 YSJ(I+2)=2*P*YSJ(I+1)-(P*P+Q*Q)*YSJ(I)
      U=0.0
      V=0.0
      UX=0.0
      VX=0.0
      F=10**60
      DO 12 I=1,K
      U=U+A(I)*XSJ(I)-B(I)*YSJ(I)
   12 V=V+A(I)*YSJ(I)+B(I)*XSJ(I)
```

```
    DO 13 I=2,K
    UX=UX+(I-1)*(A(I)*XSJ(I-1)-B(I)*YSJ(I-1))
 13 VX=VX+(I-1)*(A(I)*YSJ(I-1)+B(I)*XSJ(I-1))
    S=DABS(U)
    T=DABS(V)
    IF(S.LT.2.5*10.**(-40.0)) GO TO 14
    IF(T.LT.2.5*10.**(-40.0)) GO TO 14
    J=J+1
    F=U*U+V*V
 14 DX=-(U*UX+V*VX)/(UX*UX+VX*VX)
    DY=(U*VX-V*UX)/(UX*UX+VX*VX)
    X(J+1)=X(J)+DX
    Y(J+1)=Y(J)+DY
    P=X(J+1)
    Q=Y(J+1)
    FF(J)=F
    IF(J.LE.2)GO TO 23
    IF(FF(J-1).LT.FF(J)) GO TO 22
    GO TO 23
 22 J=J-1
    IF(J.EQ.15) GO TO 888
    L=L+1
    IF (L.EQ.10) GO TO 1000
    XDX(L)=DX
    YDY(L)=DY
    WRITE(3,201) L,DX,DY
    DX=XDX(1)
    DY=YDY(1)
    WRITE(3,201) L,DX,DY
    DX=DX-L*.25*DX
    DY=DY-L*.25*DY
    WRITE(3,201) L,DX,DY
    IF(DX.EQ.0) GO TO 1000
    G(L)=X(J)
    H(L)=Y(J)
    P=G(1)+DX
    Q=H(1)+DY
    GO TO 999
 23 WRITE(3,201) J,X(J),Y(J)
    IF(F.NE.0) GO TO 999
888 Z=DCMPLX(P,Q)
    DO 24 I=1,K
 24 COMP(I)=DCMPLX(A(I),B(I))
    BCMP(K)=COMP(K)
    KL=K-1
    DO 301 I=1,KL
    L=K-I
    LL=L+1
301 BCMP(L)=COMP(L)+BOMP(LL)*Z
    DO 302 I=1,KL
```

```
      COEF(I)=BOMP(I+1)
      SCOEF(I)=COEF(I)
      SA(I)=REAL(SCOEF(I))
      SB(I)=AIMAG(SCOEF(I))
      A(I)=SA(I)
  302 B(I)=SB(I)
      K=K-1
      IF(K.EQ.1) GO TO 1000
      IF(Q.EQ.0) GO TO 998
      W=-Q
      Z=DCMPLX(P,W)
      BOMP(K)=COEF(K)
      KJ=K-1
      DO 303 I=1,KJ
      L=K-I
      LL=L+1
  303 BOMP(L)=COEF(L)+BOMP(LL)*Z
      KKKK=K-1
      IF(KKKK.EQ.1) GO TO 1000
      WRITE(3,305)
      WRITE (3,306)
      WRITE(3,203)
      DO 304 I=1,KJ
      COEF(I)=BOMP(I+1)
      SCOEF(I)=COEF(I)
      SA(I)=REAL(SCOEF(I))
      SB(I)=AIMAG(SCOEF(I))
      A(I)=SA(I)
      B(I)=SB(I)
  304 WRITE(3,201) I,A(I),B(I)
      K=K-1
      IF(K.EQ.1)GO TO 1000
      GO TO 998
 1000 CONTINUE
      STOP
      END
```

```
        METHOD OF STEEPEST DESCENT


      FOR FINDING ZEROS OF POLYNOMIALS



 I          X(I)                        Y(I)



 1    0.30000000D 00               0.40000000D 00
 2    0.33956155D 00               0.35228385D 00
 3    0.33135642D 00               0.33095995D 00
 4    0.33381207D 00               0.33323567D 00
 5    0.33335252D 00               0.33333296D 00
 6    0.33333328D 00               0.33333408D 00
 7    0.33333316D 00               0.33333410D 00
 8    0.33333315D 00               0.33333410D 00
 9    0.33333315D 00               0.33333410D 00
10    0.33333315D 00               0.33333410D 00
11    0.33333315D 00               0.33333410D 00
11    0.33333315D 00               0.33333410D 00



      COEFFICIENTS OF REDUCED POLYNOMIAL
 I          A(I)                       B(I)
 1     0.49999857D 00             -0.13877788D-15
 2    -0.10000000D 01              0.0
 3     0.10000000D 01              0.0


 I          X(I)                        Y(I)



 1    0.30000000D 00               0.40000000D 00
 2    0.52499928D 00               0.44999857D 00
 3    0.52343745D 00               0.49687346D 00
 4    0.50128301D 00               0.49943948D 00
 5    0.50027856D 00               0.49999661D 00
 6    0.50000094D 00               0.49999849D 00
 7    0.50000004D 00               0.49999857D 00
 8    0.50000000D 00               0.49999857D 00
 9    0.50000000D 00               0.49999857D 00
 9    0.50000000D 00               0.49999857D 00
```

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Dr. S. Thomas Parker for his patient guidance and assistance in the preparation of this report.

BIBLIOGRAPHY

George Arfken;
Mathematical Methods for Physicists,
Academic Press, New York, 1966.

Edwin F. Beckenbach;
Modern Mathematics for the Engineer,
McGraw-Hill Book Company, Inc., New York, 1966.

Leonard E. Fuller;
Matrix Theory, Unpublished.

J. B. Moore;
"A Convergent Algorithm for Solving Polynomial Equations", "Journal of the Association for Computing Machinery", Vol. 14, No. 2, April, 1967, pp. 311-315.

Eduard L. Stiefel;
Numerical Mathematics, Academic Press,
New York, 1963.

THE METHOD OF STEEPEST DESCENT APPLIED TO DETERMINE ZEROS OF POLYNOMIALS AND ADAPTED FOR USE ON THE DIGITAL COMPUTER

by

JOHN DAVID HARTMAN

B.S., Bethany College, 1966

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mathematics

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1968

The method of steepest descent is a straightforward method for finding zeros of polynomials which is very useful in some situations, namely when information about the location of the zeros of a polynomial is inadequate to use the synthetic division methods. For the method of steepest descent, convergence to a zero is guaranteed and convergence to the zero is rapid in a region of the zero.

The method of steepest descent can be readily applied for use on a digital computer with only slight modifications. The purpose of this paper is to show the development of the particular form of the method of steepest descent given by J. B. Moore in the "Journal of the Association for Computing Machinery", Vol. 14, No. 2, April, 1967, pp. 311-315. Other methods of this type are available, but the form which Moore gives is easily adaptable for use on a digital computer.

Dr. Moore states the problem in the following way. Consider the equation $f(z) = u(x,y) + iv(x,y)$, where $f(z)$ is an entire function of the complex variable z, with $z = x + iy$; x and y will be considered as being functions of a parameter t. The differential equations

$$\frac{dx}{dt} = - h\frac{\partial F}{\partial x} \quad \text{and} \quad \frac{dy}{dt} = - h\frac{\partial F}{\partial y}$$

are a basis for the algorithm, where h is a positive constant and $F = F(x,y)$ is a function satisfying the following conditions:

(1) The derivatives $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$ exist.

(2) F is nonnegative.

(3) The zeros of F are located at the roots of $f(z) = 0$.

(4) These zeros are the only minima of F.

The following algorithm, which arises out of theoretical computations, is the basis for a convergent method for finding the zeros of a polynomial.

(1) Scale to have some roots within the unit circle.

(2) Choose initial approximations other than on the real axis, i.e., x_0, y_0.

(3) Compute $u(x,y)$, $v(x,y)$, $\frac{\partial u}{\partial x}$, $\frac{\partial v}{\partial x}$ and $F(x,y)$, for $x = x_0$, $y = y_0$.

(4) Compute new approximations for x and y by using

$$x_{i+1} = x_i + \Delta x \quad \text{and} \quad y_{i+1} = y_i + \Delta y,$$

where

$$\Delta x = \frac{-u\frac{\partial u}{\partial x} - v\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}, \quad \Delta y = \frac{u\frac{\partial v}{\partial x} - v\frac{\partial u}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

for $x = x_i$, $y = y_i$.

(5) Take x as x_0 and y as y_0 and recompute the parts of (3).

(6) If the value of $F(x,y)$ as now computed is greater than that of the previous iteration, reduce the increments Δx and Δy used previously until the value of F is smaller than for the preceding iteration (Dr. Moore suggests reduction by one quarter each trial).

(7) If F is sufficiently close to zero, stop. Otherwise repeat the process beginning at step (3).