CS 60

TWO ALGORITHMS BASED ON SUCCESSIVE LINEAR INTERPOLATION

BY

J. H. WILKINSON

TECHNICAL REPORT NO. CS 60
APRIL 10, 1967

This work was supported by the
National Science Foundation and the
Office of Naval Research

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

TWO ALGORITHMS BASED ON SUCCESSIVE LINEAR INTERPOLATION

by

J. H. Wilkinson*

ABSTRACT

The method of successive linear interpolation has a very satisfactory asymptotic rate of convergence but the behavior in the early steps may lead to divergence. The regular falsi has the advantage of being safe but its asymptotic behavior is unsatisfactory. Two modified algorithms are described here which overcome these weaknesses. Although neither is new, discussions of their main features do not appear to be readily available in the literature.

*National Physical Laboratory, Teddington, Middlesex, England and Computer Science Department, Stanford University.
This work was supported by N.S.F. and O.N.R.

1. Introduction. One of the simplest methods of locating a zero of a function of one variable $f(z)$ is by successive linear interpolation. (In general we do not distinguish between interpolation and extrapolation except where the content makes it self-evident that the two are being contrasted). In this method a succession of approximations z_1 is determined by the relation

$$z_{r+1} = (z_r f(z_{r-1}) - z_{r-1} f(z_r)) / (f(z_{r-1}) - f(z_r)) \quad . \tag{1.1}$$

If z_r does indeed converge to a simple zero, which without any essential loss of generality we may assume is at $z = 0$, then the ultimate asymptotic behavior is easy to analyse.

A very elementary analysis will suffice for our purposes. (For a detailed study see A. Ostrowski [1]). We assume that $f(z)$ is of the form $Az + Bz^2 + \ldots$ in the neighborhood of $z = 0$ and accordingly

$$z_{r+1} = [z_r(Az_{r-1} + Bz_{r-1}^2 + \ldots) - z_{r-1}(Az_r + Bz_r^2 + \ldots)] / [(Az_{r-1} + Bz_{r-1}^2 + \ldots) - (Az_r + Bz_r^2 + \ldots)] \sim \frac{B}{A} z_r z_{r-1} \tag{1.2}$$

Hence ultimately

$$\frac{B}{A} z_{r+1} = \left(\frac{B}{A} z_r\right)\left(\frac{B}{A} z_{r-1}\right) \tag{1.3}$$

and writing $y_r = \log\left(\frac{B}{A} z_r\right)$ we have

$$y_{r+1} = y_r + y_{r-1} \tag{1.4}$$

giving $y_r = P\lambda_1^r + Q\lambda_2^r$ where $\lambda_1, \lambda_2 = (1 \pm 5^{\frac{1}{2}})/2$ (1.5)

Accordingly the asymptotic behavior of z_r obeys the relation

$$z_{r+1} \sim k z_r^{\lambda_1} \quad , \quad \lambda_1 \doteq 1.62 \quad . \tag{1.6}$$

In this analysis f(z) could, of course, be a complex function of a complex variable but we shall be interested only in real zeros of real functions.

In spite of the asymptotic behavior of the convergents, the above process is not completely satisfactory in practice. Suppose we start from two values z_1 and z_2 such that $f(z_1)f(z_2) < 0$, i.e. $f(z_1)$ and $f(z_2)$ are of opposite signs. Then we would like to have a method which converges to a zero of f(z) lying between z_1 and z_2. That this may not happen is illustrated in Figure 1 in which it is assumed $f(Z) \to 0$ as $z \to +\infty$.

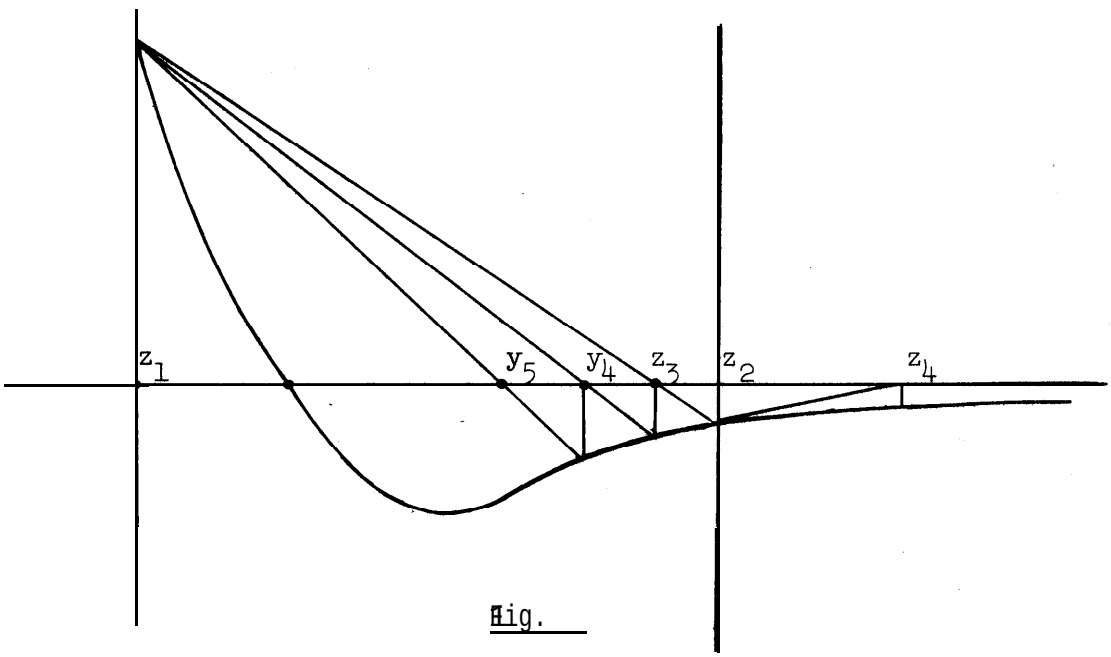

Fig.

It is clear that from z_4 onwards the convergents lie outside the interval z_1 z_2 and diverge to $+\infty$.

Convergence can be restored by ensuring that no extrapolations are employed. At each stage true interpolation is performed between the last interpolation point and the most recent previous point at which f(z) has the opposite sign. This gives the points $y_4, y_5, \ldots$ in Figure 1. We now have convergence to the root between z_1 and z_2 but the convergence rate is merely linear. In fact, we have ultimately

$$Y_{r+1} \sim y_r (f(z_1) - Az_1)/f(z_1) = \mu y_r \qquad (1.7)$$

and since A and z_1 are negative and $f(z_1) - Az_1$ is positive

$$0 < \mu < 1 \qquad (1.8)$$

However, if $f(z_1)$ is large compared with $|Az_1|$, then μ is close to 1 and convergence is slow. It may well be much slower than bisection and the latter is simpler and is just as 'safe'. This modified interpolation process is usually known as the "regula falsi".

We now describe two algorithms in which successive interpolation is used in such a way as to give superlinear convergence without sacrificing safety.

2. Algorithm 1

In Algorithm 1 [2] the 'regula falsi' is modified so as to avoid its most obvious weakness, namely, that ultimately one of the interpolation points remains fixed while the function value at the other one steadily diminishes. On the other hand it retains its main attraction, that of interpolating between function values of opposite signs. The weakness of the regula falsi may be

described in the following simple terms. Ultimately the function value at one of the interpolation points is very large compared with that at the other.

This last comment provides the clue. If $f(z)$ and $f(z_{r\,1})$ have opposite signs then there is certainly a zero in the interval (z_r, z_{r-1}). If we perform linear interpolation between weighted function values $k_r f(z_r)$ and $k_{r\,1} f(z_{r\,1})$ with k_r , $k_{r-1} > 0$ then we still obtain an intermediate point. In the method of bisection the centre point is taken, so that the weighting factors are effectively $k_r = |f(z_{r-1})|$, $k_{r-1} = |f(z_r)|$. It happens that there is a much better strategy for choosing the weighting factors than that employed in bisection or the regula falsi. It may be described as follows.

At each stage interpolation is performed between two points z_r and z_{r-1} such that $f(z_r)f(z_{r-1}) < 0$ and z_r is the last point to be determined. If z_{r-1} is being used for the sth successive time then we choose as our weighting factors

$$k_r = 1 \ (s = 1) \ , \ k_r = 2^p \ , \text{ where } p = (s-1)(s-2) \ (s > 1)$$
$$k_{r-1} = 1 \qquad (2.1)$$

The weighted interpolation point is taken to be z_{r+1} , and z_r (new) is taken to be z_r(old) or $z_{r\,1}$, the choice being made so that $f(z_{r+1})$ and $f(z_r(\text{new}))$ have opposite signs.

For simplicity of notation we shall take the zero between z_1 and z_2 to be at $z = 0$. We assume that $f(z)$ can be

represented in the form $Az+Bz^2+\ldots$ in the neighborhood of $z = 0$ and there is no essential loss of generality in assuming $A > 0$, $B > 0$. We now show that a uniform pattern of behavior ultimately emerges. Let us assume that $z_r < 0$ and $z_{r+1} > 0$ are being used for the first time as interpolation points and that both are small. The next three steps are then as follows. (See Figure 2).

STEP 1. z_{r+2} is determined by a true linear interpolation and hence from (1.2).

$$z_{r+2} = \frac{B}{A} z_{r+1} z_r \tag{2.2}$$

z_{r+2} is negative and $f(z_{r+2})$ is negative. The next interpolation is therefore between z_{r+1} and z_{r+2} .

STEP 2. Again the weighting factors are unity, we have a true linear interpolation and (1.2) gives

$$r+3 = \frac{B}{A} \quad r+2 \; z_{r+1} = \frac{B^2}{A^2} \tag{2.3}$$

Again z_{r+3} and $f(z_{r+3})$ are negative and hence the next interpolation is between z_{r+3} and z_{r+1} .

STEP 3. z_{r+1} is now being used for the third time in succession. Hence $f(z_{r+3})$ is used with a weight factor of 2. The weighted interpolation gives

$$z_{r+4} = [z_{r+3}f(z_{r+1}) - z_{r+1}2f(z_{r+3})]/[f(z_{r+1}) - 2f(z_{r+3})]$$

$$= \frac{\frac{B^2}{A^2} z_r z_{r+1}^2 (Az_{r+1}+Bz_{r+1}^2+\ldots) - 2z_{r+1}(A\frac{B^2}{A^2} z_r z_{r+1}^2+\ldots)}{Az_{r+1} + Bz_{r+1}^2 - 2(A\frac{B^2}{A^2} z_r z_{r+1}^2+\ldots)}$$

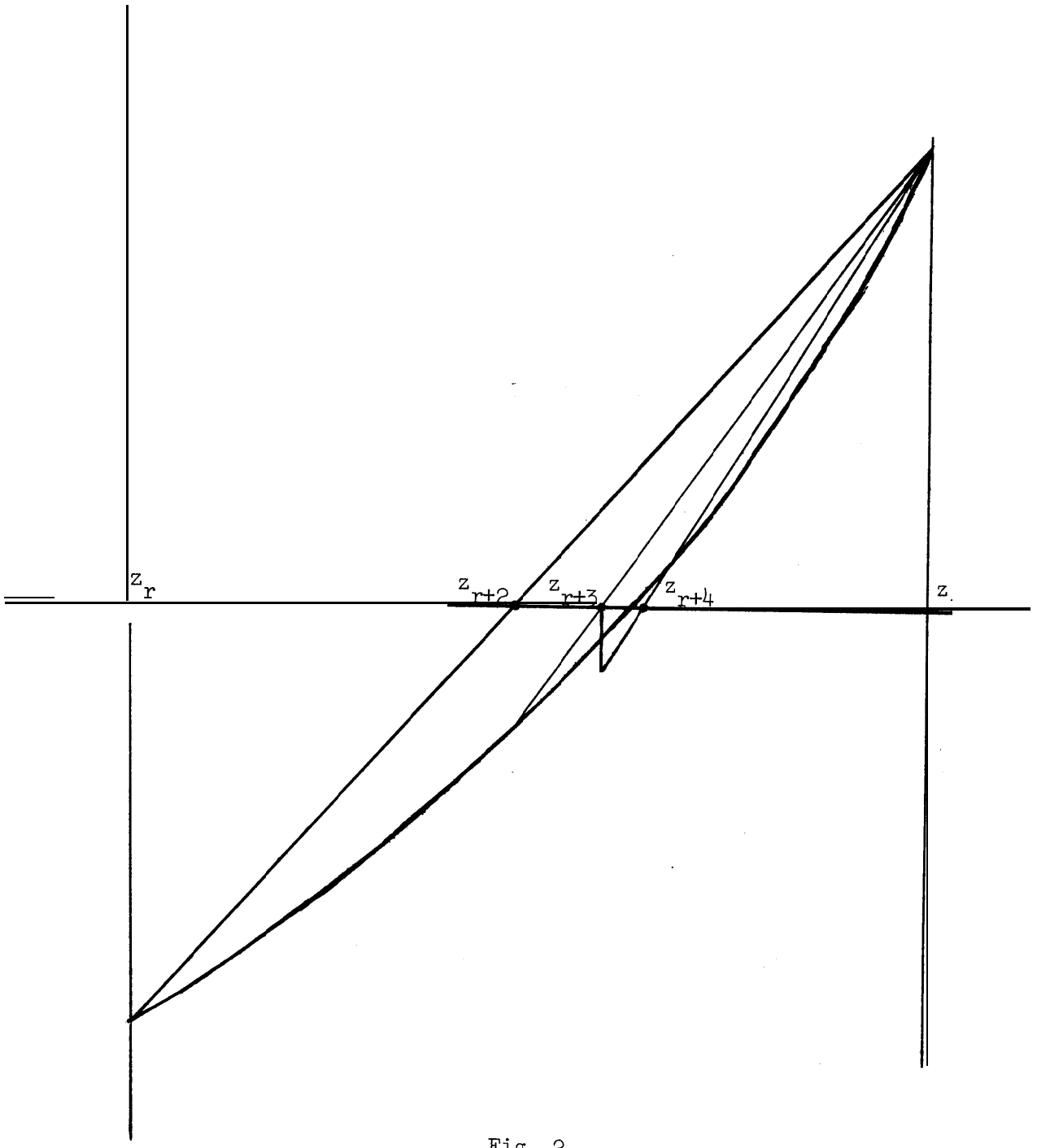

Fig. 2

$$= \frac{-\frac{B^2}{A}\, z_{r+1}^3 z_r}{A\, z_r} = \frac{-B^2}{A^2}\, z_{r+1}^2 z_r = -z_{r+3} \,. \qquad (2.4)$$

We see that z_{r+4} is now positive, (and hence $f(z_{r+4})$), so that the next interpolation is between z_{r+4} and z_{r+3} and the weighting factors are unity. Notice that z_{r+4} and z_{r+3} are equal and opposite in value at this stage, neglecting higher order terms. The next three steps are similar and

$$z_{r+6} = \frac{B^2}{A^2}\, z_{r+4}^2 \; z_{r+3} = \frac{B^2}{A^2}\, z_{r+3}^3 \qquad (2.5)$$

$$z_{r+7} = -\frac{B^2}{A^2}\, z_{r+4}^2 \; z_{r+3} = -\frac{B^2}{A^2}\, z_{r+3}^3 \qquad (2.6)$$

This shows that from now on we always have groups of three steps at a time such that

$$z_{r+3k} = -z_{r+3k+1} = \frac{B^2}{A^2}\, z_{r+3k-2}^2 \; z_{r+3k-3} = \frac{B^2}{A^2}\, z_{r+3k-3}^3 \qquad (2.7)$$

and the order of convergence is $3^{\frac{1}{3}} = 1.44$. At the end of each group of three steps $(z_{r+3k}+z_{r+3k+1})/2$ is a much better approximation than either z_{r+3k} or z_{r+3k+1} .

The modified algorithm therefore ultimately has superlinear convergence though of a lower order than successive linear interpolation. The provision of a stopping criterion is also more satisfactory with the modified procedure than with successive linear interpolation or the regular falsi. With either of the latter one is virtually forced to discriminate on the difference between

successive interpolates. That this is unsatisfactory becomes obvious if we consider the function

$$f(z) = z(z-1)^5 \text{ , with } z_1 = -\frac{1}{2} \text{ , } z_2 = 0.99 \qquad (2.8)$$

We have

$$f(z_1) = \frac{1}{2}(\frac{3}{2})^5 = 3.8 \text{ , } f(z_2) = 0.99 \times (0.01)^5 = 0.99 \times 10^{-10} \qquad (2.9)$$

Here z_3 is so close to z_2 that on a ten-decimal digit computer, for example, the computed $z_3 = z_2$. Even if we use an adequate precision a vast number of steps are needed before the limit is approached.

With Algorithm 1 we work with the distance between the two ordinates used for interpolation (f(z) always has opposite signs at these points) and the distance between these points tends to zero.

The use of progressively increasing weighting factors is of no importance as far as the asymptotic behavior is concerned but may play a vital role initially. In the above example z_1 will be used repeatedly because $f(z_1)$ is so large compared with $f(z_2)$. When it is being used for the sth successive time interpolation is performed between $2^{\frac{1}{2}(s-1)(s-2)} f(z_{s+1})$ and $f(z_1)$ and this enables us to get away from z_2 comparatively rapidly.

3. Algorithm 2.

Algorithm 1 [2] avoids interpolation between points giving function values of the same sign but in doing this it sacrifices some of the speed ultimately attainable. In Algorithm 2 [3] the

asymptotic behavior is always that of successive linear interpolation but it avoids both interpolation and extrapolation when they give unsatisfactory results.

At the beginning of the rth step three points a_r, b_r and c_r are involved. The points b_r and c_r are such that $f(b_r)$ and $f(c_r)$ are of opposite signs and $|f(b_r)| \leq |f(c_r)|$. Interpolation is always performed between a_r and b_r though the function may well have the same signs and hence give an extrapolated result. Initially two points b_1 and c_1 are given such that $f(b_1)f(c_1) < 0$ and these points are named so that $|f(b_1)| \leq |f(c_1)|$; we also take $c_1 = a_1$. The rth step is as follows apart from a minor addition which is described later.

(i) Determine a point i_r by interpolating between a_r and b_r

(ii) Determine a point m_r, the mid-point of b_r and c_r.

We 'accept' i_r if it is between b_r and m_r otherwise we regard the interpolated point as unreliable and 'accept' m_r instead. This is a 'reasonable' decision because $f(b_r)f(c_r) < 0$ and $|f(b_r)| \leq |f(c_r)|$. Hence the zero is between b_r and c_r and if f(x) is reasonably behaved from the point of view of interpolation we would expect the zero to be nearer b_r than to c_r. We then take as provisional new values

$$a_{r+1} = br \text{ , } b_{r+1} = i_r \text{ or } m_r \text{ (whichever is 'accepted'),}$$
$$c_{r+1} = c_r \qquad (3.1)$$

If b_{r+1} and c_{r+1} satisfy the conditions $|f(b_{r+1})| \leq |f(c_{r+1})|$

and $f(b_{r+1})f(c_{r+1}) < 0$ we can proceed immediately to the next step. Otherwise the provisional values are adjusted.

First if sign $f(c_{r+1})$ = sign $f(b_{r+1})$ then we take instead $c_{r+1} = b_r$ and this ensures that sign $f(c_{r+1})$ sign $f(b_{r+1}) < 0$ because of the conditions established before the rth step. We now have to make sure that $|f(b_{r+1})| \leq |f(c_{r+1})|$ (with current values of b_{r+1} and c_{r+1} of course). If this is not so, we can interchange b_{r+1} and c_{r+1} and a_{r+1} is taken to be the same as the new c_{r+1}. The right conditions now hold for the beginning of step (r+1).

The essential device here is that whenever the interpolated or extrapolated value violates a simple 'common sense' criterion the method of bisection is used and the latter is always safe. Ultimately if f(z) is well behaved near the zero interpolation or extrapolation is always used at every step.

The stopping criterion is based on the distance between b_r and c_r and these points straddle a zero at every step. The use of this criterion necessitates an additional feature which is of fundamental importance. -Suppose we have reached the situation illustrated in Figure 3. It is clear that from this point onward the b_i approach the zero monotonically from above, the interpolated points will always be accepted, the provisional points will always satisfy the required criteria and $c_i = c_r$ for all subsequent stages. Hence $|b_i - c_i|$ will not tend to zero!

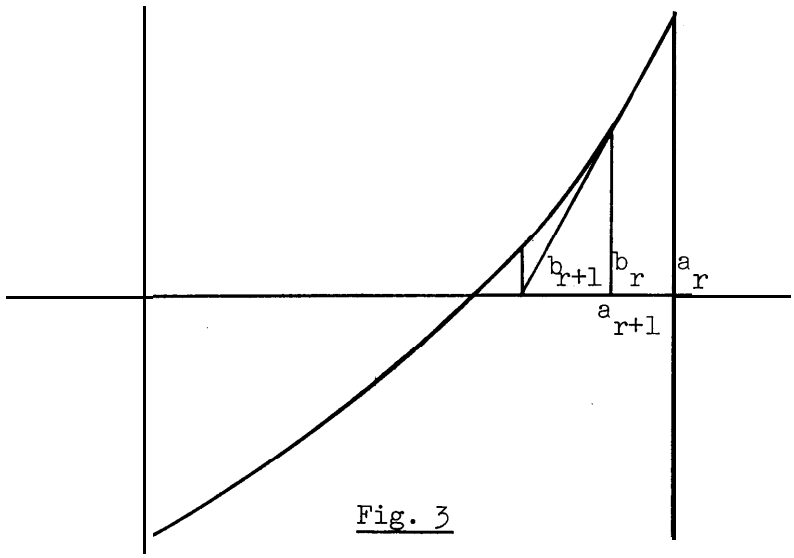

Fig. 3

This is easily overcome as follows. Suppose the stopping criterion is $|b_i-c_i| < tol$. Then if $|i_s-b_s| < tol$ the i_s is replaced by $i_s+(c_s-b_s)tol$. This ensures that when the process has converged a b_{s+1} is obtained which is beyond the zero. As soon as this happens $f(c_{s+1}) = f(c_r)$ is no longer of opposite sign from $f(b_{s+1})$ and c_{s+1} is switched in the normal way, immediately giving a b_{s+1} and c_{s+1} straddling the root and with a separation less than the tolerance.

This simple stratagem also avoids the difficulty associated with the example (2.8). Here b_2 and a_2 may well be equal to working accuracy but the modification by tol deals with this problem. Incidentally that example is dealt with more efficiently by Algorithm 2 than Algorithm 1 in the initial stages as is illustrated in Figure 4. The interpolation between b_2 and a_2 is rejected and b_3 becomes the mid-point of b_2c_2 . The next interpolation is also rejected in favor of bisection and at this stage the neighborhood of the zero has been reached.

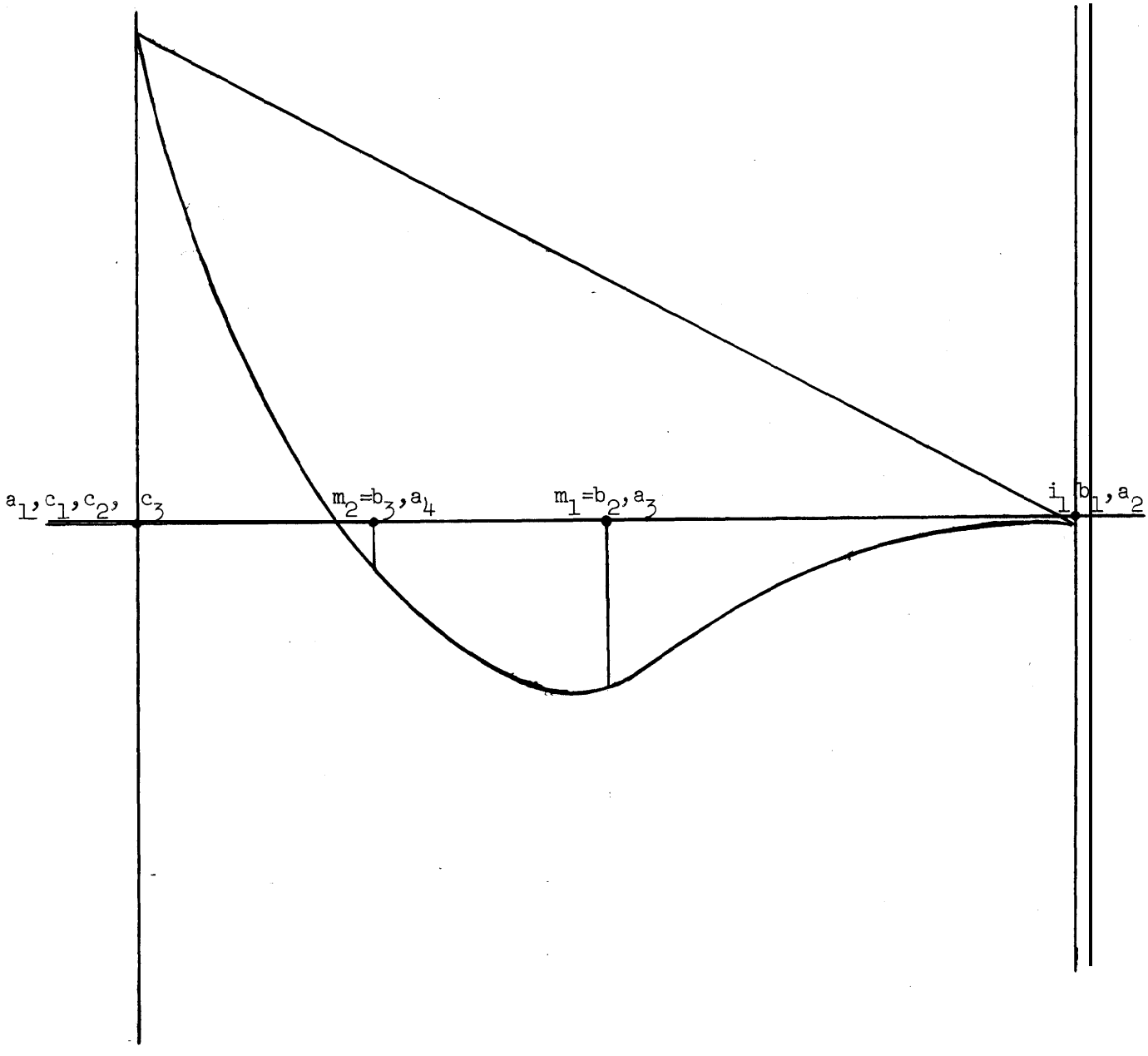

Fig. 4

4. General Comments. Of the two algorithms the second would appear to be the superior in general, though there may be situations in which the first would be superior. It is possible' that modifications of the basic idea used in Algorithm 1 might be even more successful. Both algorithms have been used extensively at N.P.L. particularly in connection with determining the zeros of $A_n-\lambda B_n$ where A_n and B_n are real band-symmetric matrices of order n and B_n is positive definite. Here one has the further advantage that the determinants of $A_r-\lambda B_r$ (n = 1,..., n) form a Sturm sequence and this can be used to locate roots initially. One can work with the zeros of $\det(A_n-\lambda B_n)$ directly or with the zeros of $\det(A_n-\lambda B_n)/\det(A_{n-1}-\lambda B_{n-1})$. The latter function has the advantage of possessing only simple zeros but has the disadvantages that when $A_{n-1}-\lambda B_{n-1}$ possesses some eigenvalues which are very close to those of $A_n-\lambda_n B$ these are concealed by the division and it also has poles. Hence the Sturm sequence location may have to be used very extensively with this function.

Acknowledgement:

This work was done while the author was Visiting Professor at Stanford's Computer Science Division during a three months leave of absence from the National Physical Laboratory, England.

References

1. Ostrowski, A. M., Solution of equations and systems of equations. 2nd edition. Academic Press, New York (1966).

2. Wilkes, M. V., Wheeler, D. J. and Gill, S. The preparation of programs for an electronic digital computer. Addison-Wesley, Reading, Massachusetts (1951).

3. van Wijngaarden, A., Zonneveld, J. A. and Dijkstra, E. W., Programs AP 200 and AP 230 De serie AP 200, edited by T. J. Dekker, The Mathematical Centre, Amsterdam (1963).