NASA CR-65994

Eighth Quarterly Report

ONE-DIMENSIONAL SEARCH COMBINING GOLDEN SECTION AND CUBIC FIT TECHNIQUES

Geraldine Guilfoyle
Analytical Mechanics Associates, Inc.

Ivan Johnson
Patrick Wheatley
NASA Manned Spacecraft Center

Report No. 67-1
Contract NAS 9-4036
January 1967

Analytical Mechanics Associates, Inc.
Westbury, Long Island, N. Y.

NASA, CR-165994

Introduction and Summary

This memorandum presents a one-dimensional search routine combining two different techniques.

The one-dimensional search determines the minimum of a continuous function of one variable. The two techniques are termed the "Golden Section" and the "Cubic Fit" methods. They may be used either independently or in combination depending on the input option chosen. For functions which are very smooth and which thus can be well approximated by a polynomial function, the cubic fit method would probably find the minima fastest. The golden section method alone is appropriate for functions having only piecewise continuous first or second derivatives and which cannot be well approximated by a low degree polynomial.

Assuming $F(k)$ to be the function, it is necessary for the use of the routine only that the following be available:

a. A point k_o, designated as the origin, such that the slope of F at k_o is negative

b. Some method for evaluating F at any $k \geq k_o$

c. Existence of at least one minimum of F for some $k > k_o$

The more assumptions that can be made about the function, the more the program can be tailored by the user to minimize execution time.

Golden Section Method

Briefly, the golden section method is an iteration scheme to nest a sequence of intervals around the function's minimum. Each iteration reduces the size of the interval until it is small enough to justify taking its midpoint as the approximation to the abscissa of the function's minimum.

The technique is derived from the "Optimal One-Dimensional Maximization Theorem" (Ref. 1). Each iteration assumes the function to be <u>unimodal</u>* over the initial interval [A, D]. This interval becomes a "golden section" with the insertion of two intermediate points, B and C, defined as follows:

$$B = \frac{3-\sqrt{5}}{2}(D-A) + A \approx .382\,|D-A| + A$$

$$C = \frac{\sqrt{5}-1}{2}(D-A) + A \approx .618\,|D-A| + A$$

The new interval [A', D'] for the next iteration is determined by comparing the functional values at k = B and k = C.

a. If $F(C) > F(B)$, $[A', D'] = [A, C]$.

b. If $F(C) = F(B)$, $[A', D'] = [B, C]$.

c. If $F(C) < F(B)$, $[A', D'] = [B, D]$.

To initiate the golden section process, A is chosen as the origin and D is the first point at which the function increases. It is terminated when (C - B) is smaller than some arbitrarily specified epsilon (usually 10^{-3} times B).

---

* A continuous function $f(x)$ is defined as unimodal over an interval $[A, D]$ if there exists a point $x_0 \in [A, D]$ such that $f(x)$ is strictly decreasing on $[A, x_0)$ and strictly increasing on $(x_0, D]$.

Cubic Fit Method

The cubic fit method is a scheme to approximate the function with a cubic equation. The minimum of the cubic may then provide a good approximation to the function minimum.

To initiate the cubic fit, a minimum of four and a maximum of ten points and corresponding function values must be known.

If (x_i, y_i) represent a pair of these points, then one of the following two techniques are used to determine the coefficients of the cubic equation

$$y(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

a. The method of least squares is used when $5 \leq i \leq 10$. This means solving the system:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_i (x_i)^2 & \sum_i (x_i)^3 \\ \sum_{i=1}^{n} x_i & \sum_i (x_i)^2 & \sum_i (x_i)^3 & \sum_i (x_i)^4 \\ \sum_{i=1}^{n} (x_i)^2 & \sum_i (x_i)^3 & \sum_i (x_i)^4 & \sum_i (x_i)^5 \\ \sum_{i=1}^{n} (x_i)^3 & \sum_i (x_i)^4 & \sum_i (x_i)^5 & \sum_i (x_i)^6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \\ \sum_i x_i^2 y_i \\ \sum_i x_i^3 y_i \end{bmatrix}$$

b. When $i = 4$, the method of least squares reduces to a perfect fit.

$$\begin{bmatrix} 1 & x_1 & (x_1)^2 & (x_1)^3 \\ 1 & x_2 & (x_2)^2 & (x_2)^3 \\ 1 & x_3 & (x_3)^2 & (x_3)^3 \\ 1 & x_4 & (x_4)^2 & (x_4)^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Once $y(x)$ is completely determined, its minimum is computed as follows:

a. If $a_3 \not\approx 0$, $y_{min} = \dfrac{-a_2 + \sqrt{(a_2)^2 - 3a_1a_3}}{3a_3}$

b. If $a_3 \approx 0$ but $a_2 > 0$, $y_{min} = \dfrac{-a_1}{a_2 + \sqrt{(a_2)^2 - 3a_1a_3}}$

Otherwise, no cubic minimum can be determined.

Development of the Logic

The program can be logically divided into three parts. Each part is outlined by a brief summary and a detailed flow chart. The FORTRAN IV listing found in Appendix B is subdivided into the same three sections.

The purpose of Part 1 is to establish some golden section [A, D] containing the first minimum of the function. Initially this is done by choosing A equal to the origin and D as the first point at which the function shows an increase. B and C are computed in the golden mean ratio. In order that the function be unimodal, F(B) must be less than F(A) and F(C) less than F(D). Part 1 is executed only once during the program unless within some smaller interval nested in [A, D] the unimodality assumption is later proven false.

The primary purpose of Part 2 is to reduce the size of the interval containing the minimum. This is done by comparing the function values of the intermediate points B and C. If the function has increased from B to C, the minimum must be in [A, C]. If it has decreased, the minimum must be in [B, D]. If it has not changed, the minimum is in [B, C].* The new interval is divided into the golden mean ratio yielding a new golden section interval. If this interval passes the unimodality test, control is passed to Part 3; otherwise the program returns to Part 1.

As an option, points and their functional values may be accumulated one-by-one up to a maximum of MPTS.** These points are used in Part 3 for the cubic fit.

---

* The assumption of unimodality is essential for the validity of this logic.

** The value of MPTS is specified by the user.

In Part 3 the minimum of the function is sought by one of three methods determined by the input parameter KUBIC. All the methods use, as a starter, the golden section found in Part 1, which is known to contain the minimum of the function.

The first method (KUBIC = 0) is the golden section method. When the interval [B, C] is sufficiently small, that is, less than EPSGS,* the final minimum, CAYMIN, becomes the midpoint between B and C. Until this criterion is met, control is returned to Part 2.

The second method (KUBIC = 1) abandons the golden section technique and relies entirely on a perfect cubic fit (MPTS = 4) with the previous minimum used as one of the points.

The third method (KUBIC = 2) combines the golden section and cubic fit techniques. The cubic equation is determined using points from the golden section and may be either a perfect fit (MPTS = 4) or a least squares fit ($5 \le \text{MPTS} \le 10$).

The second method fits a cubic equation and finds its minimum. Whichever point has the greatest function value is replaced by the minimum of the cubic and another cubic is computed and minimized. This process continues until a minimum is found which is within a pre-specified distance from the point with the smallest function value. This minimum then becomes CAYMIN. Any of the following conditions will cause this method to fail:

a. The nature of the cubic makes its minimum impossible to determine.

---

* This parameter is prescribed as a constant value, but in a comparison it is used relative to the magnitude of the variables with which it is compared.

b. The minimum of the cubic is not in the interval containing the function minimum.

c. The function value of the cubic minimum is greater than that of any one of the four points used to determine the cubic equation.

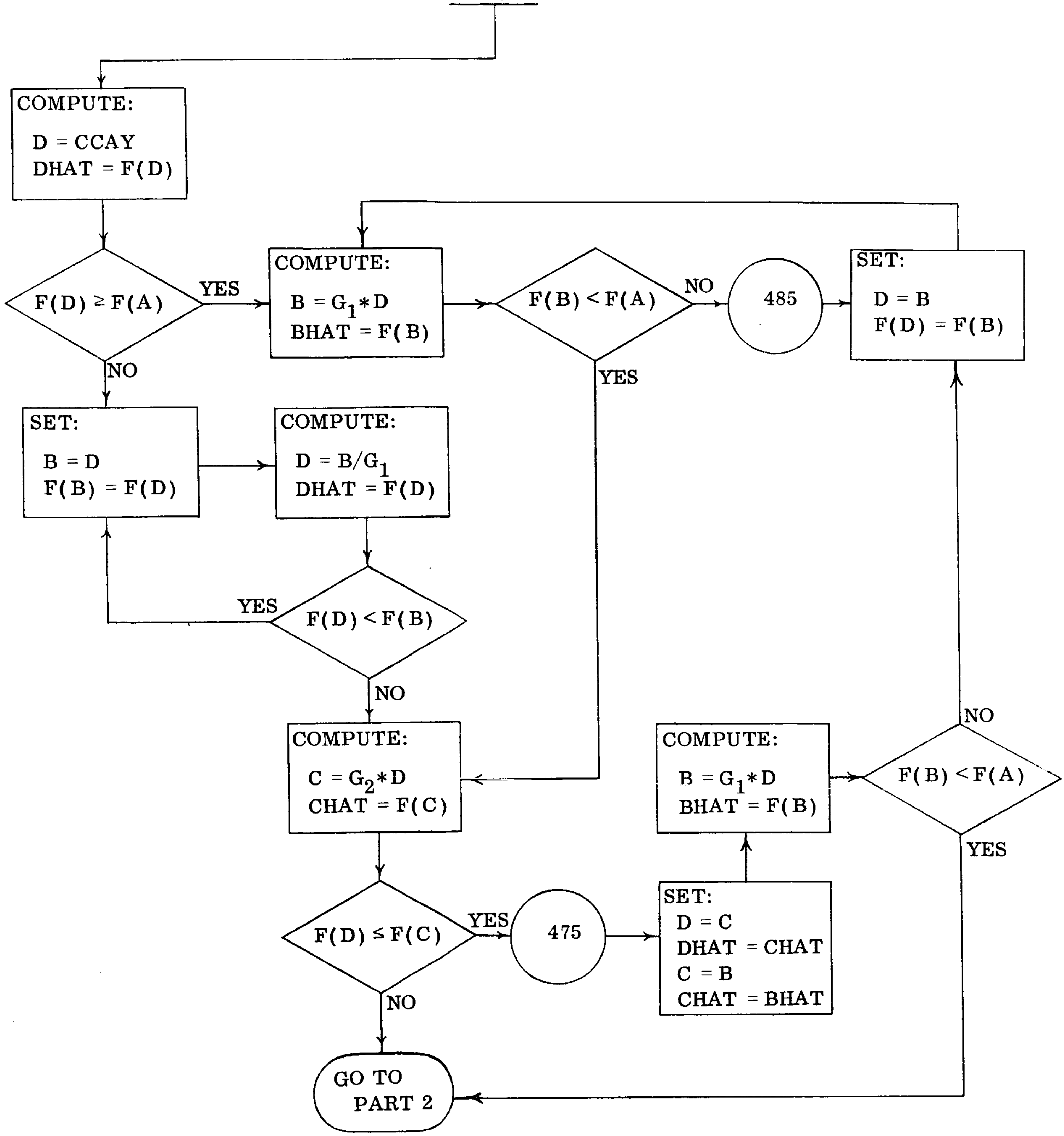
Should any of these occur, the program reverts to Method 3 to insure that the final minimum can still be determined.

The golden section criterion for Method 1 is utilized also in Method 3. However, if it is not met, a cubic equation is fit and its minimum found and stored. The final minimum, CAYMIN, is the midpoint between two successive minima found by the cubic fit technique, provided their distance apart is not greater than the parameter EPSCF.* Until this criterion is met, control is returned to Part 2 (the golden section technique). This guarantees that if the cubic fit cannot find the minimum, eventually it will be located by the golden section technique.

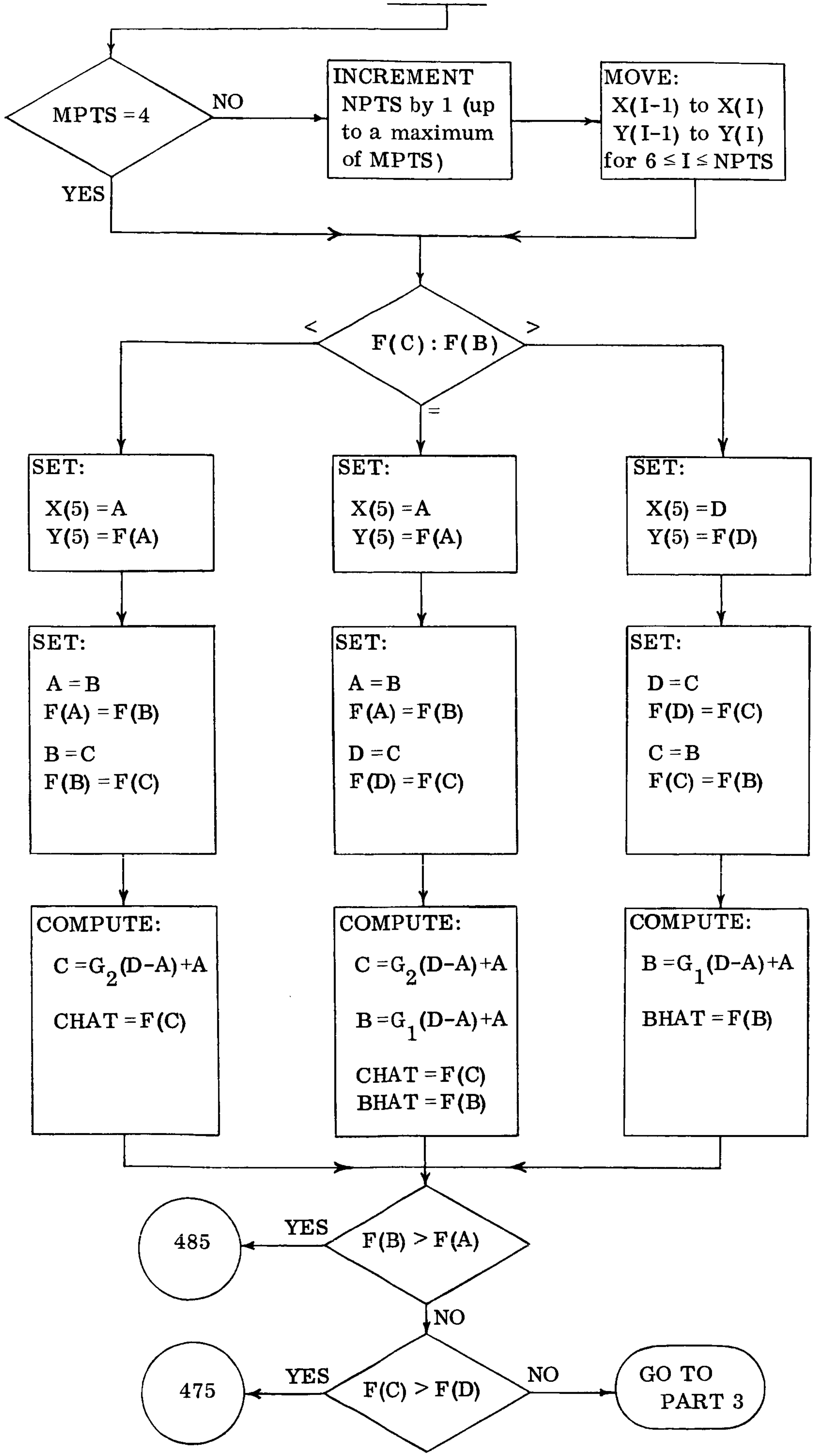
---

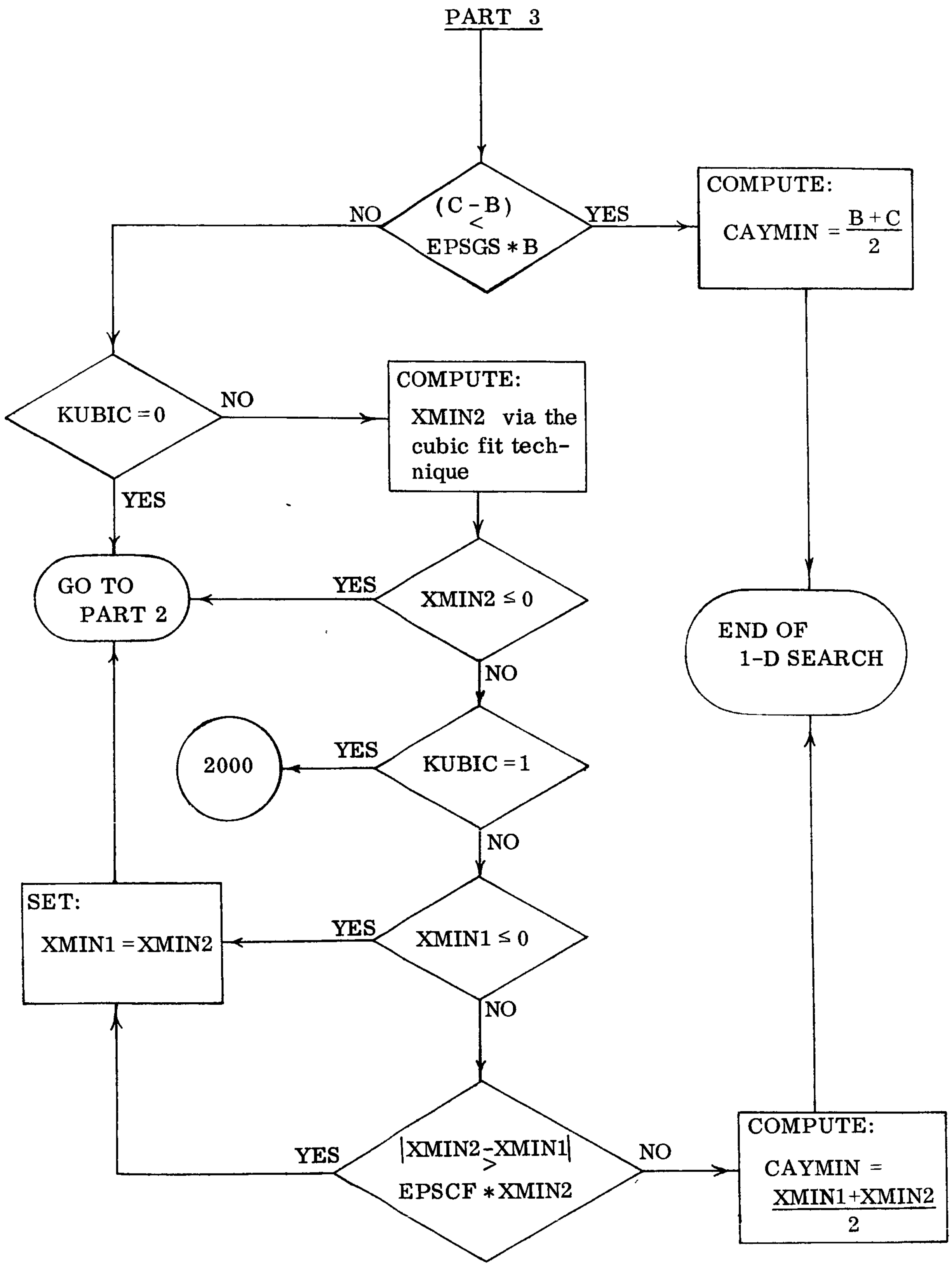
* This parameter is prescribed as a constant value, but in a comparison it is used relative to the magnitude of the variables with which it is compared.

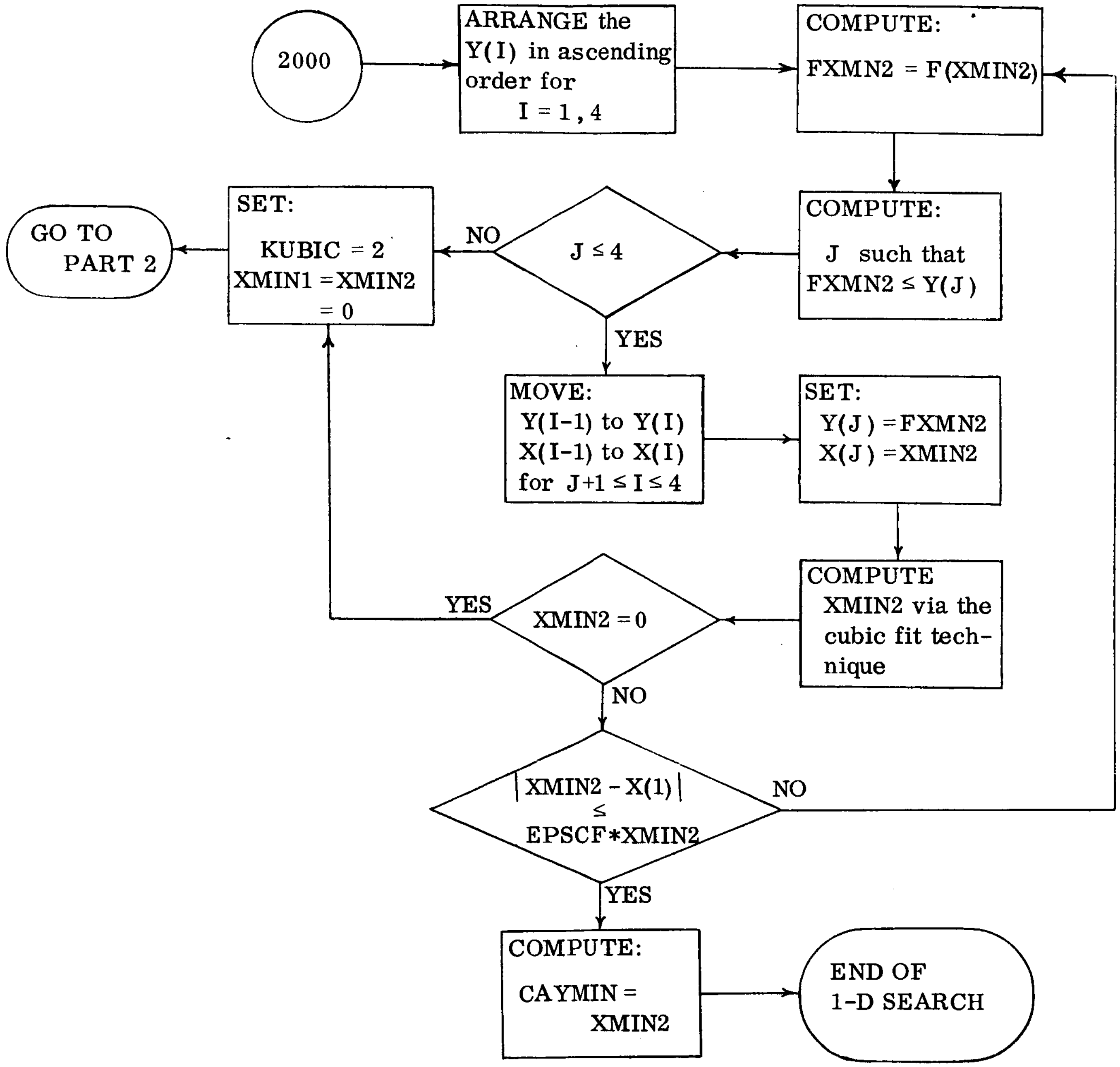
PART I

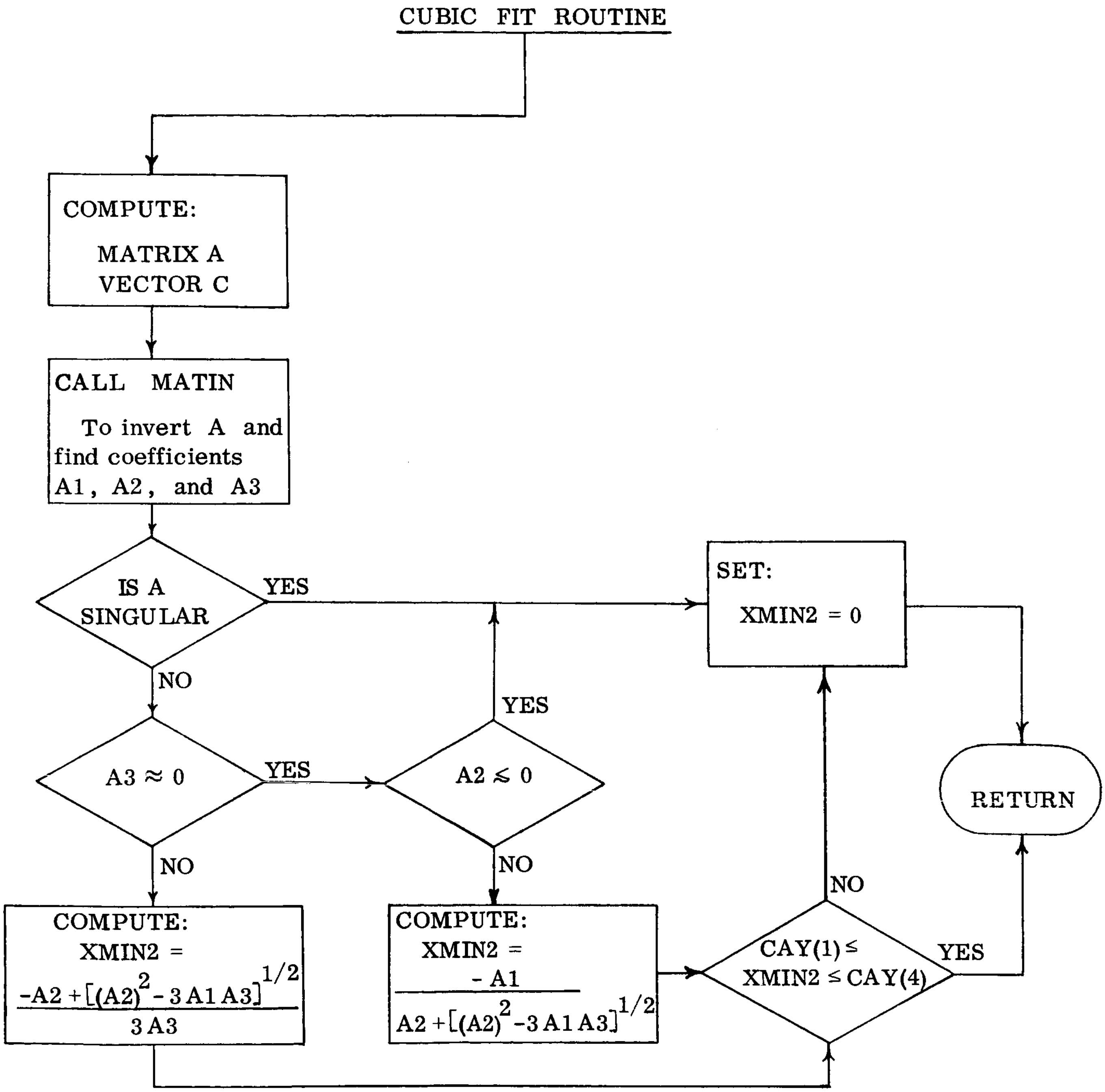

COMPUTE:
D = CCAY
DHAT = F(D)
F(D) ≥ F(A)
YES
NO
COMPUTE:
B = G1*D
BHAT = F(B)
F(B) < F(A)
NO
YES
485
SET:
D = B
F(D) = F(B)
SET:
B = D
F(B) = F(D)
COMPUTE:
D = B/G1
DHAT = F(D)
F(D) < F(B)
YES
NO
COMPUTE:
C = G2*D
CHAT = F(C)
COMPUTE:
B = G1*D
BHAT = F(B)
F(B) < F(A)
NO
YES
F(D) ≤ F(C)
YES
NO
475
SET:
D = C
DHAT = CHAT
C = B
CHAT = BHAT
GO TO
PART 2

PART 2


MPTS = 4
NO
YES
INCREMENT NPTS by 1 (up to a maximum of MPTS)
MOVE: X(I-1) to X(I) Y(I-1) to Y(I) for 6 ≤ I ≤ NPTS
F(C) : F(B)
<
>
=
SET: X(5) = A Y(5) = F(A)
SET: X(5) = A Y(5) = F(A)
SET: X(5) = D Y(5) = F(D)
SET: A = B F(A) = F(B) B = C F(B) = F(C)
SET: A = B F(A) = F(B) D = C F(D) = F(C)
SET: D = C F(D) = F(C) C = B F(C) = F(B)
COMPUTE: C = G2(D-A) + A CHAT = F(C)
COMPUTE: C = G2(D-A) + A B = G1(D-A) + A CHAT = F(C) BHAT = F(B)
COMPUTE: B = G1(D-A) + A BHAT = F(B)
F(B) > F(A)
YES
485
NO
F(C) > F(D)
YES
475
NO
GO TO PART 3


PART 3
(C - B) < EPSGS * B
NO
YES
COMPUTE:
CAYMIN = (B + C)/2
KUBIC = 0
NO
YES
COMPUTE:
XMIN2 via the cubic fit technique
GO TO PART 2
YES
XMIN2 ≤ 0
NO
END OF 1-D SEARCH
2000
YES
KUBIC = 1
NO
SET:
XMIN1 = XMIN2
YES
XMIN1 ≤ 0
NO
YES
|XMIN2 - XMIN1| > EPSCF * XMIN2
NO
COMPUTE:
CAYMIN = (XMIN1+XMIN2)/2


2000
ARRANGE the Y(I) in ascending order for I = 1, 4
COMPUTE: FXMN2 = F(XMIN2)
COMPUTE: J such that FXMN2 ≤ Y(J)
J ≤ 4
NO
SET: KUBIC = 2 XMIN1 = XMIN2 = 0
GO TO PART 2
YES
MOVE: Y(I-1) to Y(I) X(I-1) to X(I) for J+1 ≤ I ≤ 4
SET: Y(J) = FXMN2 X(J) = XMIN2
COMPUTE XMIN2 via the cubic fit technique
XMIN2 = 0
YES
NO
\|XMIN2 - X(1)\| ≤ EPSCF*XMIN2
NO
YES
COMPUTE: CAYMIN = XMIN2
END OF 1-D SEARCH


CUBIC FIT ROUTINE
COMPUTE:
MATRIX A
VECTOR C
CALL MATIN
To invert A and find coefficients A1, A2, and A3
IS A SINGULAR
YES
NO
A3 ≈ 0
YES
NO
A2 ≲ 0
YES
NO
SET:
XMIN2 = 0
RETURN
COMPUTE:
XMIN2 =
-A2 + [(A2)^2 - 3 A1 A3]^1/2 / 3 A3
COMPUTE:
XMIN2 =
- A1 / A2 + [(A2)^2 -3 A1 A3]^1/2
CAY(1) ≤ XMIN2 ≤ CAY(4)
YES
NO

References

1. Bellman, R. and Dreyfus, S.; Applied Dynamic Programming, Princeton University Press, Princeton, N.J., 1962, p. 153. See Appendix A for a summary.

APPENDIX A

Theorems and Formulas Related to The Golden Section Method

The basis for choosing the Golden Section Method is the "Optimal One-Dimensional Maximization Theorem" (Ref. 1). In essence this theorem states that if F_n represents the interval of maximum length on which it is possible to locate the minimum of a unimodal function by calculating the function at most n times, then F_n is a Fibonacci number. That is

$$F_n = F_{n-1} + F_{n-2} \quad , \quad n \geq 2 \tag{1}$$

where $F_o = F_1 =$ an arbitrary unit interval.

It is easily shown that, except for the boundary conditions, (1) is satisfied by choosing

$$F_{i-1} = \frac{\sqrt{5} - 1}{2} F_i \tag{2}$$

For simplicity define:

$$G_2 = \frac{\sqrt{5} - 1}{2} \approx .6180339887498949$$

and

$$G_1 = 1 - G_2 = \frac{3 - \sqrt{5}}{2} \approx .3819660112501051$$

Applied to the golden section iteration notation, F_n is the interval $[A, D]$. Then the points B and C should be chosen in such a way that

$$F_{n-1} = [A, C] = [B, D]$$

since these would normally represent the "best" interval on the next iteration. But (2) implies that

$$G_2(D - A) = (C - A) = (D - B)$$

or finally that B and C are determined by

$$B = G_1(D - A) + A$$

$$C = G_2(D - A) + A$$

The terminology "golden section" comes from the Greek "golden mean" which divided an interval into two segments, one of which was $\frac{\sqrt{5} + 1}{2}$ times the other. This is equivalent to the segments being in the ratio $G_2{:}G_1$.

With the golden section points A, B, C, D and the numbers G_1 and G_2 defined as above, the following equivalency relationships are used in the program:

Let A', B', C', D' represent the points in the next golden section.

Theorem 1. If $[A', D'] = [A, C]$ then $B' = A + C - B$ and $C' = B$

Theorem 2. If $[A', D'] = [B, D]$ then $B' = C$ and $C' = B + D - C$

Theorem 3. If $[A', D'] = [B, C]$ then $C' = 3B - C - A$ and $B' = C + B - C'$

Before the theorems can be proven, the following lemmas are necessary:

Lemma 1. $(G_2)^2 = G_1$

Proof follows immediately from the definitions of G_1 and G_2.

Lemma 2. $G_1 * G_2 = G_2 - G_1$

Proof: $G_1 + G_2 = 1$ (by definition)

$G_1 * G_2 + (G_2)^2 = G_2$

$G_1 * G_2 + G_1 = G_2$ (by Lemma 1)

$G_1 * G_2 = G_2 - G_1$

Lemma 3. $(G_1)^2 = 2G_1 - G_2$

Proof: $G_1 + G_2 = 1$

$(G_1)^2 + G_1 * G_2 = G_1$

$(G_1)^2 = G_1 - G_2 + G_1$ (by Lemma 2)

Proof of Theorems 1-3

Theorem 1.

Given: $B' = G_1(D' - A') + A'$ (by definition of B')

$= G_1(C - A) + A$

$= G_1[G_2(D - A)] + A$ (by definition of C)

$= (G_1 * G_2)(D - A) + A$

$= (G_2 - G_1)(D - A) + A$ (by Lemma 2)

$= G_2(D - A) - G_1(D - A) + 2A - A$

$= [G_2(D - A) + A] - [G_1(D - A) + A] + A$

Conclusion: $B' = C - B + A$ (by definition of C and B)

Given: $D' - C' = B' - A'$

$C - C' = B' - A$

$C' = C - B' + A$

$C' = C - (C - B + A) + A$

Conclusion: $C' = B$

Theorem 2.

Given: $C' = G_2(D' - A') + A'$

$= G_2(D - B) + B$

$= G_2(C - A) + B$

$= G_2[G_2(D - A)] + B$ (by definition of C)

$= (G_2)^2(D - A) + B$

$= G_1(D - A) + B$ (by Lemma 1)

$= G_1(D - A) + G_1(D - A) + A$ (by definition of B)

$= 2[G_1(D - A) + A] - A$

$= 2B - A$

$= B + (B - A)$

Conclusion: $C' = B + (D - C)$

Given: $D' - C' = B' - A'$

$D - C' = B' - B$

$D - (B + D - C) + B = B'$

Conclusion: $C = B'$

Theorem 3.

Given: $C' = G_2(D' - A') + A'$

$= G_2(C - B) + B$

$= G_2[G_2(D - A) - G_1(D - A)] + B$ (by definition of B and C)

$= G_2[G_2 * G_1(D - A)] + (B - A) + A$

$= (G_2)^2 * G_1(D - A) + G_1(D - A) + A$ (by definition of B)

$= (G_1)^2(D - A) + G_1(D - A) + A$ (by Lemma 1)

$= 3G_1(D - A) - G_2(D - A) + 3A - 2A$ (by Lemma 3)

$= [3G_1(D - A) + 3A] - [G_2(D - A) + A] - A$

Conclusion: $C' = 3B - C - A$

Given: $(D' - C') = (B' - A')$

$C - C' = B' - B$

Conclusion: $B' = B + C - C'$

APPENDIX B

Dictionary of FORTRAN Terms

Term	Meaning
MPTS	The maximum number of points desired for the cubic fit. $4 \leq MPTS \leq 10$
NTAPE	Output tape drive
EX1	Initial value of the abscissa. It is designated as the origin.
CAYMIN	Value of the abscissa which yields the final minimum of the function.
KUBIC	Input parameter indicating which method will be used to find the function minimum.
EPSGS	Parameter defining how small the interval [B,C] must be before accepting its midpoint as CAYMIN.
EPSCF	Parameter used to determine when a cubic minimum is acceptable for computing CAYMIN.
ITC	Number of times the function has been evaluated during the search.
ITT	Cumulative sum of the ITC's if the search is executed more than once during a program.
CCAY	Estimate for the abscissa value to initiate the search for the first golden section containing the minimum.
XMIN1, XMIN2	Successive minima obtained by the cubic fit technique.
FXMN1, FXMN2	Functional values of XMIN1 and XMIN2.

Term	Meaning
CAY(I) I=1,4	I^{th} point in the current golden section, i.e. CAY(1) = A, CAY(3) = C, CAY(2) = B, CAY(4) = D
FHAT(I)	Functional value of CAY(I).
CALC1, CALC2	Double precision storage locations for intermediate calculations.
EX(1)	Value of the abscissa at which the function is to be evaluated.
JSW	Control switch to return to the proper program location after evaluating the function.
F	Value of the function at EX(1)
X(I) I=1,NPTS	Points used to fit a cubic equation.
Y(I) I=1,NPTS	Functional value of X(I).
A	The matrix which must be inverted in order to find the coefficients of the cubic equation.
C	1) Before inverting matrix A, C is the vector of constant terms of the system of equations yielding a cubic fit. 2) After the inversion of A, C contains the solution to this system, i.e., the coefficients of the cubic fit.
KEY	Rank of matrix A
A_1, A_2, A_3	Coefficients of the cubic fit $A_3 x^3 + A_2 x^2 + A_1 x + A_0 = y$
MATIN	Subroutine used to invert matrix A and solve for vector C.
KSW	Control switch to return to the proper program location after finding the cubic minimum.