A PROCEDURE FOR THE SOLUTION OF LAMBERT'S ORBITAL BOUNDARY-VALUE PROBLEM

R. H. GOODING

Royal Aerospace Establishment, Farnborough, Hants, England

(Received: 30 May 1989; accepted: 9 January 1990)

Abstract. A procedure is described that provides a universal solution for Lambert's problem. Based on the approach of Lancaster and his colleagues, the procedure uses Halley's cubic iteration process to evaluate the unknown parameter, x, at the heart of the approach, initial estimates for x being selected so that three iterations of the process always suffice to yield an accurate value. The overall procedure has been implemented via three Fortran-77 subroutines, listings of which are appended to the paper, and the way in which the subroutines have been tested is outlined.

1. Introduction

Lambert's 'orbital boundary-value problem', like the solution of Kepler's equation, continues to attract the attention of mathematicians searching for computing procedures of ever-greater generality, accuracy and efficiency. The problem may be stated as follows: an unperturbed orbit, about a given inverse-square-law centre of force, C say, is to be found connecting two given points, P_1 and P_2, with a flight time Δt $(=t_2 - t_1)$ that has been specified. There will always be at least one solution, and the actual number (N say) depends on the 'kinematic geometry' of the triangle CP_1P_2 and the value of Δt; it will be assumed, for convenience and with no loss of generality, that $\Delta t > 0$.

Interest in the problem goes back more than 200 years, and the contributions to celestial mechanics of Lambert himself have been recorded by Volk (1980). The renewed interest of the space age may be largely attributed to the problem's relevance to orbital rendezvous; thus, papers have recently been published at a rate approaching one a year, and 22 (from 1961 to 1987) were cited in the author's extensive report on the subject (Gooding, 1988a), of which the present paper is a shortened version. (Copies of 'the Report', as it will be referred to from now on, can be made available on request.) A number of notable contributions have been made by Battin, culminating in two chapters of a new text-book (Battin, 1987), which together constitute a recension of his preceding papers. However, the outstanding contribution to the subject has been the pair of papers by Lancaster and his colleagues: the first of these (Lancaster, Blanchard and Devaney, 1966) was extremely short and the material was expanded in the later paper (Lancaster and Blanchard, 1969). The present paper builds on the basic approach of Lancaster *et al.*, extending it to the following important topics: iteration; starting formulae (initial estimates) for the iteration process; universal accuracy; and subroutine testing.

Celestial Mechanics and Dynamical Astronomy **48:** 145–165, 1990.

2. Kinematic Geometry

We use the term 'kinematic geometry' to reflect the geometry of the triangle CP_1P_2 in relation to the kinematics of the Lambert problem's solution(s). Suppose first that the triangle is not degenerate, so that θ, the angle subtended by P_1P_2 at C, lies between 0 and π. Then it would appear that there must be at least two solutions, since an orbital path can be found that subtends an angle $2\pi - \theta$, as well as one that subtends θ. We can avoid this duality by supposing the direction of motion to be specified in advance, so that the two angles can be deemed to define different problems. There is a further complication, however, since if Δt is large enough, other paths will be possible, each of which includes a number of complete revolutions, and such paths, incorporating a specific number of complete revolutions (m say) normally occur in pairs. Hence, as Δt increases for a given triangle and specified direction of motion, N is an increasing odd integer, apart from being even (instantaneously) at each value of Δt at which two new solutions emerge. We simplify the approach to multiple revolutions by extending the distinction between individual Lambert problems so that θ is regarded as an angle of unrestricted positive magnitude defined by the geometry of the path and not just by that of the triangle; we write θ_r for the reduced angle (such that $0 < \theta_r < 2\pi$) when it is necessary to discriminate. We have now effectively redefined N such that $N = 1$ if $\theta < 2\pi$; if $\theta > 2\pi$, on the other hand, $N = 0$, 1 or 2, depending on whether Δt is less than, equal to or greater than a particular critical value.

Turning to degenerate triangles, we consider these on the basis of the unlimited values of θ just introduced, so that θ is now $k\pi$ for some integer k ($\geqslant 0$). Then if orbital paths exist that are themselves still non-degenerate, their number must be infinite, since any plane through the degenerate triangle contains valid paths. If we choose an orbital plane (as well as the direction of motion) arbitrarily, however, we have $N = 0$, 1 or 2, if k is odd, exactly as in the last paragraph; this is actually the simplest of all cases to deal with in practice, though the literature contains a number of solution procedures that fail here quite unnecessarily. But there are real difficulties when k is even (and identifiable with $2m$). First, the orbital path is now necessarily rectilinear, unless P_1 and P_2 coincide (in which case the orbit is highly arbitrary even when $m > 0$). More interesting is the existence of a form of discontinuity, the effect of which is that to maintain the truth of the statement '$N = 0$, 1 or 2' we have to distinguish the angle $(k\pi)_-$, which symbolizes the representation of θ as $2(m - 1)\pi$ plus a θ_r of 2π, from $(k\pi)_+$, which symbolizes its representation as $2m\pi$ plus a θ_r of zero.

We can now summarize the data involved in the Lambert-problem solution procedure to be developed. The input quantities are the constant μ (strength of the given force centre at C), r_1 and r_2 (equal to CP_1 and CP_2), θ (the unrestricted angle P_1CP_2) and Δt. The output quantities, $4N + 1$ in number, consist of N itself and N sets of four quantities, viz. V_R (radial velocity) and V_T (transverse velocity) at both P_1 and P_2. It is assumed that the value of N does not exceed 2, though no distinction between $(2m\pi)_-$ and $(2m\pi)_+$ is made in the procedure, multiple revolutions of a rectilinear orbit being meaningless in practice.

3. Lambert Invariance

Lambert invariance is a concept that derives from Lambert's theorem and the equivalence classification of triangles that this makes possible. Lambert's theorem may be stated as follows (see Wintner (1941) or Sarnecki (1988) for a universal proof): if c is the chord (side P_1P_2) of the triangle CP_1P_2, then Δt (for a connecting orbital path) can be expressed as a (multivalued) function of just three quantities (not counting μ), viz. $r_1 + r_2$, c and a, this last being the semi-major axis of the path (taken as negative for hyperbolic orbits). Defining s as the semi-perimeter of the triangle, so that $r_1 + r_2 = 2s - c$, and noting that a is equivalent to the total energy of the motion per unit mass, we can also express Δt as a function of s, c and energy.

Lambert's theorem shows that triangles with the same values of s and c are equivalent, from the viewpoint of the relation between Δt and energy, and the set of all triangles CP_1P_2 can thus be divided into equivalence classes. Each class contains a unique (apart from orientation) isosceles triangle with $r_1 = r_2$, and the general class (with $0 < c < s$) contains a pair of degenerate triangles such that one of the points P_1 and P_2 lies between the other point and C; a connecting orbit for either of these degenerate triangles is necessarily rectilinear. Classes with $c = 0$ contain only a single member each, which is simultaneously isosceles and doubly degenerate. The other extreme (with $c = s$) is such that the classes have their widest membership, in regard to the $r_2 : r_1$ ratios possible, though *all* members are now degenerate – each class contains a pair of doubly degenerate triangles such that either P_1 or P_2 coincides with C, whilst the remaining (singly degenerate) triangles (infinite in number, as in the general case) all have P_1 and P_2 on *opposite* sides of C. Connecting orbits for the singly degenerate triangles of this extreme case cannot be rectilinear, but a connecting orbit for either of the doubly degenerate triangles is *bound* to be rectilinear; for the extreme classes with $c = s$, therefore, it is convenient to regard the term 'degenerate' as referring only to the doubly degenerate triangles. If, further, we cease to distinguish between the pair of degenerate triangles with $r_1 < r_2$ and $r_1 > r_2$, then we can say, in all cases, that an equivalence class contains exactly one degenerate triangle.

Triangles in the same equivalence class may be described as 'Lambert congruent', or L-congruent for brevity, and this introduction of a congruence relation suggests the allied one of similarity, just as in elementary geometry; thus, two triangles may be described as L-similar whenever they have the same value of c/s, this being a dimensionless quantity. L-similarity is a weaker relation than L-congruence, with wider equivalence classes, and the two relations induce two different equivalence classifications for individual Lambert problems; then a Lambert-invariant parameter (geometric or dynamic), for either relation, may be defined as one that has the same value for all the problems in the same equivalence class. A sufficient condition for two problems to be equivalent, on a congruence basis, is that s, c and Δt are the same for both; to be equivalent on a similarity basis, it suffices that c/s and an appropriate non-dimensionalized Δt are the same. Thus, one fewer parameter is required to characterize the problems in the same similarity class, as opposed to those in the same

congruence class, and it follows that similarity is normally the more useful type of equivalence.

We must recognize that θ, introduced as an ideal triangle parameter in Section 2, is not Lambert-invariant (in either sense), but in principle we can get round this by using φ (instead of θ) as a parameter of the general triangle, where φ is defined as being the θ for the equivalent isosceles triangle: then φ can be regarded (like θ) as an angle of unrestricted magnitude. In computation, however, it is better to follow Lancaster *et al.* in taking the defining parameter for each triangle class, under L-similarity, to be the quantity q, given by

$$q = \cos \tfrac{1}{2}\varphi_r/(1 + \sin \tfrac{1}{2}\varphi_r) = \tan \tfrac{1}{4}(\pi - \varphi_r). \tag{1}$$

Here φ_r is φ reduced to the range $(0, 2\pi)$, so that

$$\sin \tfrac{1}{2}\varphi_r = c/(2s - c). \tag{2}$$

Clearly, Equations (1) and (2) lead to the simple result that

$$q^2 = (1 - \sin \tfrac{1}{2}\varphi_r)/(1 + \sin \tfrac{1}{2}\varphi_r) = 1 - c/s, \tag{3}$$

but Equation (3) does not define the sign of q, whereas (1) does. We also have, from (1) and (2),

$$q = (1 - c/2s) \cos \tfrac{1}{2}\varphi_r; \tag{4}$$

when expressed in terms of θ_r, instead of φ_r, this last result may be written

$$q = \frac{\sqrt{r_1 r_2}}{s} \cos \tfrac{1}{2}\theta_r. \tag{5}$$

The only disadvantage in using q, rather than φ, to characterize the underlying triangle in Lambert's problem, is that m is then required as well, to specify the number of completed circuits.

4. Iteration Parameter

As with Kepler's equation, Lambert's problem has no satisfactory direct solution and we have to proceed by iteration on a suitable unknown parameter. The use of a Lambert-invariant parameter is desirable, and we have seen that it is best tied to L-similarity rather than L-congruence. The energy-equivalent orbital parameter a is certainly Lambert-invariant (though only in the narrower sense), but this is not true of e (eccentricity) or p (semi-latus rectum). The use of p as iteration variable is intuitively appealing, in spite of this, because of its direct relation to true anomaly and hence to θ, and it has been recommended as recently as by Boltz (1984). But p fails, in a somewhat paradoxical fashion, when $\theta = \pi$. The paradox is that p is given, as $2r_1r_2/(r_1 + r_2)$, without the need to iterate at all: because Δt is not involved in this formula, however, no further progress can then be made without iterating on some other variable.

Thus a is an immediate candidate for the iteration variable, but its direct use would

be unsatisfactory because orbital paths with a particular value of a occur in pairs or not at all. There is, in fact, an upper limit (corresponding to minimum possible energy) to the possible values of $1/a$; it is given by $2/s$, for which value the pair of paths coincide. It follows from this that if we write

$$x^2 = 1 - s/2a, \tag{6}$$

then x is a satisfactory substitute for a, such that the choice of sign in the implied square root distinguishes between the two paths of each pair. The parameter x is universal (defined independently of the type of orbital path), unlike the parameter g used by Sun *et al.* (1987), for example; in addition, x is non-dimensional, being associated with L-similarity as desired.

The parameter x, which has just been introduced, was originally adopted by Lancaster *et al.*, and has also been used in the later work of Battin. (But it is ignored in the paper by Taff and Randall (1985), who tacitly assume that the only energy-equivalent parameter available is a itself.) It has been shown by Sarnecki (1988) that x has a dynamical interpretation, being a non-dimensionalized value of the velocity at P_1 in the (rectilinear) solution of the Lambert problem for the degenerate triangle that is L-congruent to the given one and is such that $r_1 \geqslant r_2$; x is positive or negative according to whether the direction of V (now pure radial) is inward or outward, and this effectively resolves the ambiguity in the sign of x, outstanding from the last paragraph. With the ambiguity resolved, it is clear from Equation (6) that $|x| < 1$ for elliptic orbits, $x = 1$ for parabolas and $x > 1$ for hyperbolas. (Values of $x \leqslant -1$ are assumed not to arise, though they could be associated with negative values of Δt.)

5. The Algorithm for Flight-Time

We have seen that the flight-time can be derived from x and a single triangle parameter (φ), so long as Δt is expressed in non-dimensional form. Following Lancaster *et al.*, therefore, we replace Δt by T, where

$$T = (8\mu/s^3)^{1/2}\Delta t. \tag{7}$$

Though T is a function of just x and φ, it has already been noted that in practice it is better to compute from q and m, rather than φ; Figure 1 plots T against x for particular values of q and m (corresponding to selected values of φ), following the figure of Lancaster *et al.* Its most striking feature consists in the gaps (unrealizable regions) that occur in the part of the figure associated with elliptic orbits. Thus, as φ increases through a multiple of 2π, for a fixed value of x such that $|x| < 1$, q jumps from -1 back to $+1$ and there is a jump in the value of T. The value of φ is the same ($2m\pi$) for both pairs of values of q and m, which is why (as noted in Section 2) we should distinguish $(2m\pi)_-$ from $(2m\pi)_+$ if we work with φ. Further analysis of Figure 1, and its unrealizable regions, can be found in the Report.

The basic formulae for computing T from x, q and m were given in the papers by Lancaster *et al.*, and have been repeated with further details (and some improvements)

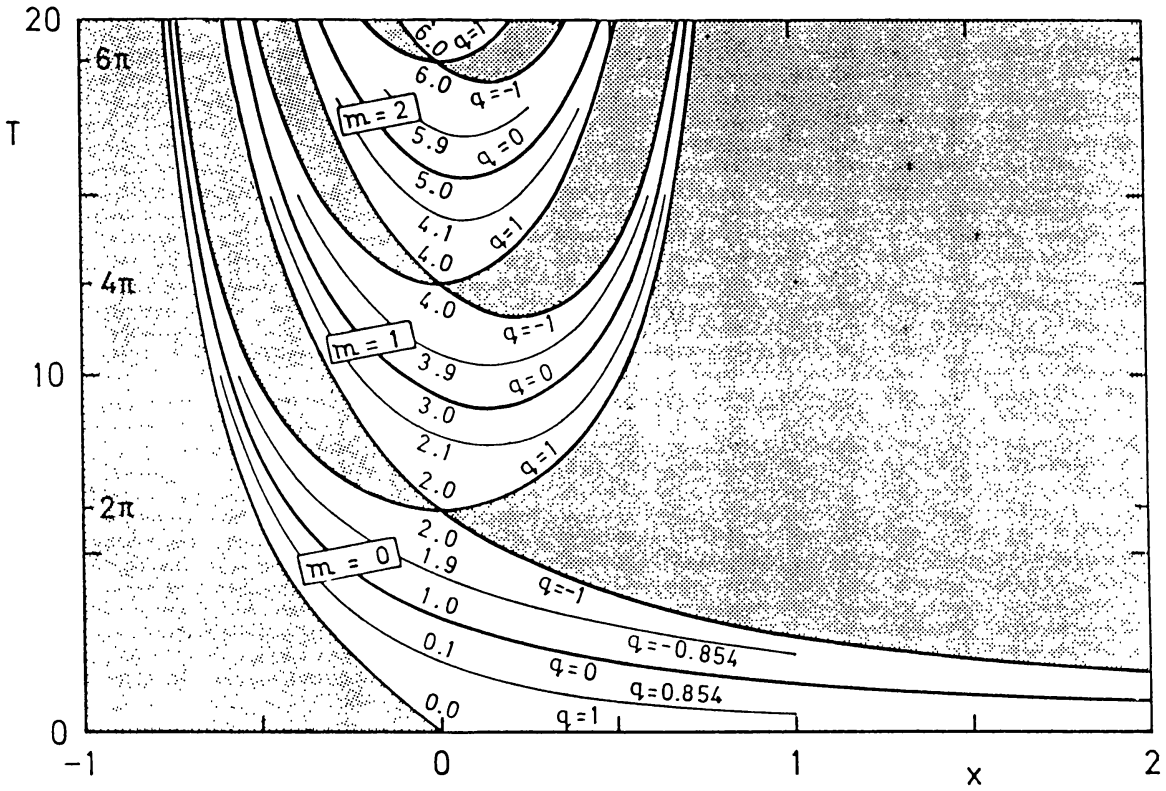

Fig. 1. Plot of T vs x for selected values of φ/π (corresponding to q and m as indicated).

in an appendix to the Report. The algorithm has been implemented by the Fortran-77 subroutine, TLAMB, listed here as Appendix A. TLAMB is at the heart of the paper's universal procedure for solving Lambert's problem, but the description here will be confined to a few remarks. The first is that, in addition to T, the subroutine can generate partial derivatives of T with respect to x; up to three derivatives, denoted by T', T'' and T''', can be computed, the number being as specified by the fifth argument of the subroutine. (Formulae for the derivatives are given in the Report's appendix just cited.) Secondly, the basic formulae of Lancaster *et al.* do not give full accuracy in all circumstances, and the somewhat complicated operation of TLAMB has been designed to recitify this; in particular, rounding error is minimized by series computation when x is (arbitrarily) in the range $\sqrt{0.6} < x < \sqrt{1.4}$, the two series of Lancaster *et al.* being amalgamated to increase accuracy and efficiency. Thirdly, one of the quantities used by the subroutine is conveniently introduced here, as it is needed in Section 9. The quantity is z, defined by

$$z = \sqrt[+]{(1 - q^2 + q^2x^2)}. \tag{8}$$

It has a dynamical interpretation, similar to that given for x in Section 4, following Sarnecki (1988); thus z/q is a non-dimensionalized value of the velocity at the closer point in the degenerate Lambert problem (z is never negative because the direction of this velocity is dealt with by the sign of q). The computation of z by Equation (8) is a good example of the potential loss of accuracy that TLAMB avoids, since in many

situations it is desirable that $1 - q^2$ be regarded as a quantity available independently of q; this is true, in particular, when $m = 0$ and q is close to 1, i.e. when φ is small. Because of this danger, the subroutine has $1 - q^2$ as an extra argument, the assumption being that it is obtained, if necessary (before TLAMB is called), from φ, via the formula

$$1 - q^2 = c/s = 2 \sin \tfrac{1}{2}\varphi_r/(1 + \sin \tfrac{1}{2}\varphi_r), \tag{9}$$

which is immediate from (3).

6. Iteration Process

Though the starting value, x_0, has to be available before iteration can begin, the iteration process is considered first, because the iteration process drives the starting formulae rather than vice versa. Thus, if just the basic Newton–Raphson method of iteration is used, the devising of starting formulae to cover all cases accurately becomes an almost impossible task. When the Halley process, found to work extremely well in the solution of Kepler's equation, for hyperbolic orbits (Gooding and Odell, 1988) as well as elliptic orbits (Odell and Gooding, 1986), is used instead, however, the task is greatly eased. This is well illustrated by the solution for x when its true value is 0.5 and φ is small, say $10^{-5}\pi$. Then (with the starting formula given by Equation (11) in the next Section) x_0 is roughly double the true x, after which the Newton-Raphson process leads to an x_1 very close to zero and the iterative process effectively stagnates; the Halley process, on the other hand, gives an x_1 very close to the true value, after which convergence is rapid. In *defence* of the Newton–Raphson process, on the other hand, it is remarked that it works very well when the value of φ approaches 2π and $|x|$ is small, so long as the appropriate starting formula from Section 7.2 is used; this is notwithstanding the explicit warning (against using the process in these circumstances) issued by Lancaster and Blanchard (1969), to which there will be further reference at the end of Section 7.2.

The Halley method is essentially the Newton–Raphson method extended to give third-order convergence, so it requires T'' as well as T'. Since Halley's method was also adopted for the iteration involved in a subsidiary problem that arises when $m > 0$ (see Section 7.3), to satisfy an equation expressed in terms of T', we sometimes also need T'''; this is why the subroutine TLAMB generates derivatives up to the third.

In the present study, the iteration process is incorporated in the Fortran-77 subroutine, XLAMB, that generates solutions for x and is listed in Appendix B. The input for XLAMB consists of m, q, $1 - q^2$ (supplied separately for the same reason as in TLAMB) and T, and its output is as follows: the integer N (from Section 2) that specifies the actual number of solutions (it should get set to 0, 1 or 2, though a value of -1 is also theoretically possible, constituting the flag to be defined in Section 8); x, a solution when at least one exists; and x^+, the second solution when there are two (x^+ is actually the *first* of the two solutions to be described in Section 7.3, and is therefore always positive).

7. Starting Formulae

7.1. INTRODUCTORY REMARKS

The requirement in regard to starting formulae for the iteration process is for an approximation to the inverse of the function that (for given m and q) generates T from x. When $m = 0$, there is formally no difficulty, since there is always a unique x to which the 'starter' is to approximate (from the assumption $T \neq 0$). When $m > 0$, on the other hand, this uniqueness does not normally apply: the normal cases are of 'no solution' and 'two solutions', but the case of just one solution, arising as two solutions merge, is covered as well.

All the starters have been incorporated in the subroutine XLAMB.

7.2. SINGLE-REVOLUTION STARTERS

For $m = 0$, the curves in Figure 1 are strictly monotonic, except when $q = 1$ and $x > 0$ (excluded cases with $T = 0$). As an obvious first move, we can determine the sign of x by evaluating (by TLAMB) the value of T, T_o say,* corresponding to $x = 0$. Then the sign of x is the sign of $T_o - T$, and it is used to distinguish two possibilities.

Suppose first that $x > 0$. Then we can approximate to the contour $T(x, q, m = 0)$, for the given q (regarded as fixed), by the bilinear curve

$$T = T_o^2/(T_o + 4x). \tag{10}$$

The rationale for this is that Equation (10) satisfies the following three constraints: first, T tends to zero as x tends to infinity; secondly, $T = T_o$ for $x = 0$; and finally, $T' = -4$ for $x = 0$. (This is a fixed value, independent of q, when $|q| < 1$.) The merit of the bilinear approximation is that it is immediately invertible, the inverse function being also bilinear; thus from (10) we get, as our starter for $x > 0$,

$$x_0 = T_o(T_o - T)/4T. \tag{11}$$

If $x < 0$, we proceed in the same way, this time approximating the $T(x, q, 0)$ contour by the bilinear curve that is most naturally expressed as

$$T = T_o - 4x/(x + 1). \tag{12}$$

The first of the three constraints that lead to (12) is that T tends to infinity as x tends to -1; the other two constraints are the same as for the case $x > 0$. On inverting, to get x as a bilinear function of T, we have

$$x_{01} = -(T - T_o)/(T - T_o + 4). \tag{13}$$

This formula, (13), is as simple as the complementary one, (11), but it was found to work much less well and there was a need to patch it; this is why the left-hand side has been written x_{01} rather than x_0. The patching was of a complicated nature, with some

* The zero suffix in T_o is written differently from the zero suffix in x_0, to reduce possible confusion between the two different meanings.

arbitrary features, but it will be summarized (in the next three paragraphs) for completeness.

Equation (13) actually requires two patches. The necessity for the first became apparent for values of φ greater than about 1.999π, and it is ascribable to the proximity of the (left-hand half of the) curve for $\varphi = 2\pi$ ($q = -1$), for which the value of T'_o should actually be zero (as opposed to -4). Now T'_o would automatically be zero for a curve that is bilinear in x^2 instead of x. Hence we are led to consider the alternative approximation

$$T = T_o(1 + \tfrac{1}{2}x^2)/(1 - x^2), \tag{14}$$

in which only the term $\frac{1}{2}x^2$ requires explanation – it gives T''_o its correct value (6π) for $\varphi = 2\pi$, where $T_o = 2\pi$. (The behaviour of T'' is worth noting: if $|q| < 1$, it is defined and finite for all x, but unbounded as $|q|$ approaches unity and x approaches zero; if $|q|$ is constrained to be unity, on the other hand, T'' is bounded in the neighbourhood of $x = 0$, and though undefined at $T = 0$ may be regarded as 'effectively defined' since it has the same limiting value for an approach to the limit from either side.) The inversion of (14) may be written

$$x_{02} = -\sqrt{\{(T - T_o)/(T + \tfrac{1}{2}T_o)\}}. \tag{15}$$

At this stage we have two possible formulae for our desired starter, the first (and simpler) being normally to be preferred; thus the possibility of a weighted combination arises. We start by computing an empirical criterion, given by

$$W = x_{01} + 1.7\sqrt{(2 - \varphi/\pi)}, \tag{16}$$

in which, since $x_{01} < 0$, the two terms are of opposite sign (and φ is derived from Equation (20), to be presented). If $W \geqslant 0$, we use x_{01} unmodified, but if $W < 0$ we use a weighted combination of x_{01} with x_{02}, so that the starter is potentially of the form

$$\left.\begin{aligned} x_{03} &= x_{01} && \text{if} \quad W \geqslant 0 \\ x_{03} &= x_{01} + w(x_{02} - x_{01}) && \text{if} \quad W < 0 \end{aligned}\right\}, \tag{17}$$

where (still empirically) the weight w is $(-W)^{1/16}$; the transition from the pure x_{01} to the weighted x_{03} is clearly continuous.

The second limitation of the simple starter, (13), was associated with (true) values of x close to -1, the potential starter, now x_{03}, giving a value much *too* close to -1. This flaw has been dealt with by applying an empirical factor λ to x_{03}, so that the final starter (for $x < 0$) is given by

$$x_0 = \lambda x_{03}. \tag{18}$$

The formula used for λ is

$$\lambda = 1 + c_1 x_{03}(1 + x_{01}) - c_2 x_{03}^2\sqrt{(1 + x_{01})}, \tag{19}$$

with (empirical) values of c_1 and c_2 that are available from the listing of Appendix B.

A point concerning the patching of (13) in the vicinity of $(x, q) = (0, -1)$ is worth

noting. We see from (16) that W is zero at this point, so the patching associated with (17) can have little effect in its vicinity; moreover, the effect of the patching associated with (18) is very slight for $x \approx 0$, so (13) is essentially unpatched. But the x_{02}/x_{03} patching was only introduced because x_{01} has the 'wrong' left-hand derivative at $(0, -1)$, so there is an apparent paradox here. The paradox is resolved if we bear two points in mind. First, the basing of (13) on the value of T_o means that there can never be a problem when $x = 0$. Secondly, so long as q is not *exactly* -1, the derivative actually has the 'right' value (-4) at $(0, q)$, no matter how close φ is to 2π. Thus, for a value of φ such as 1.99999π, the unpatched starter fails in the vicinity of an x-value around -0.05; much closer to zero, however, all is well again. In this context, the inflexion-point curve of Lancaster and Blanchard (1969) is relevant, though their associated statement, that it is necessary to abandon the Newton–Raphson process in the region in question, has been found to be quite unwarranted.

7.3. MULTIREVOLUTION STARTERS

For $m > 0$, we can again make use of the value of T_o (corresponding to $x = 0$, for the given q), but this is not now the primary source of information in devising starting formulae. The primary source is T_M, the minimum value of T for the given q, together with x_M, the associated value of x, where T_M and x_M are always uniquely defined; further, $x_M = 0$ when $q = 1$, and otherwise $x_M > 0$. Once T_M has been determined, we immediately know whether the Lambert problem will have two solutions ($T > T_M$), one ($T = T_M$), or none at all ($T < T_M$), but the evaluation of T_M (and x_M) is no mean task, itself requiring an iterative process – a solution is required to the equation $T' = 0$. For this subsidiary problem, the same iteration process has been adopted as for the Lambert problem (with the resulting need for T'''), so two points remain to be discussed: the starter for this subsidiary problem; and the formulae for the two Lambert-problem starters that will be needed as soon as values for T_M and x_M are available (assuming $T_M < T$).

In regard to the starter for x_M, we need the value of φ_r, which can be recovered from q, since, from (1) and (9),

$$\tfrac{1}{2}\varphi_r = \arg(2q, 1 - q^2), \tag{20}$$

where $\arg(x, y)$ is $\tan^{-1}(y/x)$ with two arguments, computed in Fortran by the ATAN2 function. It can be shown that for $\varphi_r = \pi$ (i.e. $q = 0$) a good approximation to x_M is given by $4/\{3\pi(2m + 1)\}$, which we denote by $x_{M,\pi}$. From this it has been found, empirically, that for $\varphi_r < \pi$, a good starting formula is (omitting the formal zero-suffix)

$$x_M = x_{M,\pi}(\varphi_r/\pi)^{1/8}; \tag{21}$$

for $\varphi_r > \pi$, it is (symmetrically)

$$x_M = x_{M,\pi}\{2 - (2 - \varphi_r/\pi)^{1/8}\}. \tag{22}$$

Having determined x_M and the corresponding T_M, we now suppose that $T_M < T$, so

that the Lambert problem has a pair of solutions with a starting formula required for both. For one solution we have $x_M < x$, so x is certainly positive, and this leads to the simpler of the two starters, since T_o is not used. In principle we get x_0 (the required starter) by inverting the particular bilinear approximation to T, as a function of $(x - x_M)^2$, that (i) gives $T = T_M$ when $x = x_M$, (ii) makes T tend to infinity as x tends to unity, and (iii) has the correct second derivative (T''_M) at $x = x_M$. The formula for this starter, for which a λ-type patch is necessary, is given in the Report.

For the other solution we have $x < x_M$, so we base the starter on the sign of $T - T_o$ to distinguish between two possibilities. If $T > T_o$, then $x < 0$ and we proceed exactly as when $m = 0$ and $x < 0$; thus we just use T_o, not requiring T_M, and we patch the elementary formula for x_0 in two different ways as before. Finally, if $T < T_o$, we have $0 < x < x_M$, and now we use both T_o and T_M. We base x_0 on the particular bilinear approximation to T, as a function of $(x - x_M)^2$, that (i) passes through the points with (x, T) equal to (x_M, T_M) and $(0, T_o)$, and (ii) has the correct second derivative (T''_M) at the former point. The formula for the starter appears in the Report, no patching being necessary.

8. Completion of Convergence

It was decided (again following the work on the two Kepler equations) to aim at a fixed number of iterations for x, rather than employ a convergence test. In program testing, carried out on a PR1ME computer providing 14 decimal digits in double precision, it was found, for $m = 0$, that three iterations always sufficed for the determination of x to 13 digits, suggesting that (for levels of precision up to this) no truncation error remained; it was legitimate, therefore, to fix the number of iterations at three. The foregoing is an over-simplified statement of the accuracy achieved, however, and an amplified version now follows.

It was arranged, in testing, that the relative error in x (true value assumed known – see Section 10) after a pre-set number of iterations should be computed, as well as the relative error (residual) in the associated value of T; the *smaller* of these two relative errors, ε say, was recorded for each test case, and after three iterations it was found that ε never exceeded 10^{-13}. (The rationale here is that the accuracy in a numerical solution of the general equation $f(x) = Y$ should always be assessed in terms of the numerically smaller of the relative error in x and the relative residual in Y; assessment in terms of just the former could amount to a demand for the impossible.) For completeness, it is remarked that when the process was reduced to two iterations, the maximum value of ε was about 2×10^{-6}, this being the relative error in x for $q \approx -0.62$ ($\varphi \approx 1.71\pi$) and $x \approx -0.9944$. For a single iteration, on the other hand, the maximum value of ε was 4.3×10^{-3}, occurring for φ close to 2π and $x \approx 2.28$. Finally, the maximum value of ε prior to any iteration, i.e. due to the starter itself, was about 0.5, being associated with the possible over-estimate (in x_0) by 100%, when φ is small, noted in Section 6.

For $m > 0$, the three-iteration value of ε was not limited to 10^{-13} in all cases,

though for $m = 1$ the maximum value was only 1.1×10^{-13}. For $m > 1$, the value grows slowly with m, the explanation being attributable to an imperfection in the patching, with effects described in more detail in the Report; the imperfection must be regarded as rather academic, however, the surprising thing being, perhaps, that the multirevolution starters perform as well as they do.

We also have to consider the convergence of x_M in the 'subsidiary problem' associated with $m > 0$. For this, it was soon clear that a fixed number of iterations would not be appropriate. Something of a dilemma was involved here, since, on the one hand, great accuracy in x_M should not really be required (as x_M is only a step to the starter for the main problem); in critical cases, on the other hand, an incorrect x_M could lead to the erroneous conclusion that the particular Lambert problem possesses no solution. It was eventually decided to operate on the (conservative) basis that iteration would be terminated as soon as the value of x_M changed by less than 3 parts in 10^7 during the current iteration. With the cubic convergence of Halley's method, this meant (in principle) that, in a hypothetical further iteration, x_M would not change by more than 3 parts in 10^{20}; this expectation was confirmed when it was found that there was rarely any change in the value computed for T_M if the 'further iteration' was actually performed. Tests for values of φ covering the full range of q were carried out. The maximum number of iterations needed to satisfy the x_M criterion was found to be nine, but the number was only three in the vast majority of cases; this was true even for large values of φ, the convergence being essentially dependent on q rather than m. To provide a guaranteed exit, however, it was arranged that if the criterion was not satisfied within twelve iterations, then the process would be abandoned and a flat set; reference to this flag was made in Section 6.

It is worth remarking on how the convergence for x_M is affected if the Halley process is replaced by the Newton–Raphson process. For most of the q-range, the effect is to add only a single iteration, taking the total from three to four. For values of q approaching 1, however, i.e. for correspondingly small values of φ_r, there is a steady rise in the number of iterations required by the Newton–Raphson process – the maximum number experienced in the PR1ME testing was 16, as against the nine for the Halley process.

9. Computation of Velocity

We now come to the solution of an individual Lambert problem. It is assumed that we have obtained a value of x (and x^+ also, when appropriate) for the Lambert-equivalent class of problems, and it remains to compute, for both of the points P_1 and P_2, the velocity components (V_R and V_T) corresponding to the specific quantities (r_1, r_2, θ and Δt) of the original problem.* This computation is performed by the overall

* To solve the Lambert problem as originally stated (in Section 1), we must first derive θ_r from the coordinates of P_1 and P_2, and we have to decide for which θ (defined by θ_r) we actually require solutions. We also have to express our solution(s) three-dimensionally, of course.

Lambert-solving subroutine, VLAMB, which calls XLAMB (which calls TLAMB); VLAMB is listed in Appendix C and it will be seen that its arguments are precisely the quantities summarized (for the full Lambert solution procedure) at the end of Section 2.

A formula for $V_{R,1}$ was given by Lancaster and Blanchard (1969). This can be improved upon, however, to make it more accurate and efficient. The resulting formula is

$$V_{R,1} = \gamma\{(qz - x) - \rho(qz + x)\}/r_1, \tag{23}$$

where $\gamma = \sqrt{(\mu s/2)}$ and $\rho = (r_1 - r_2)/c$. The formula for $V_{R,2}$, similarly, is

$$V_{R,2} = -\gamma\{(qz - x) + \rho(qz + x)\}/r_2. \tag{24}$$

We must have $|\rho| \leqslant 1$, of course, except that ρ is indeterminate when $c = 0$ (P_1 and P_2 coincide), in which case VLAMB arbitrarily sets $\rho = 0$. But if $c = 0$, $q = \pm 1$, so $z = |x|$ by (8); then for $x \neq 0$ there are two possibilities, depending on whether q and x are of the same or opposite sign. Only when the signs are the same do $V_{R,1}$ and $V_{R,2}$ themselves become indeterminate, and then the arbitrary value of ρ yields conventional values for $V_{R,1}$ and $V_{R,2}$. There is no indeterminacy when q and x have opposite signs, since then $qz + x = 0$; thus $V_{R,1}$ and $V_{R,2}$ are independent of ρ, the orbits being rectilinear.

The formula for $V_{T,1}$ given by Lancaster and Blanchard is indirect, involving the orbital elements a and e, with a serious threat to accuracy in awkward cases. The direct formula is

$$V_{T,1} = \gamma\sigma(z + qx)/r_1, \tag{25}$$

where, in principle, σ is defined as $(1 - \rho^2)^{1/2}$; to minimize rounding error, however, in practice we compute $\sigma = 2(r_1 r_2/c^2)^{1/2} \sin \frac{1}{2}\theta_r$, setting $\sigma = 1$ (compatible with $\rho = 0$) when $c = 0$. The formula for $V_{T,2}$, similarly, is

$$V_{T,2} = \gamma\sigma(z + qx)/r_2. \tag{26}$$

Naturally, $V_{T,1}$ and $V_{T,2}$ are indeterminate in the same circumstances as $V_{R,1}$ and $V_{R,2}$.

For use in Equations (23)–(26), the values of $qz - x$, $qz + x$ and $z + qx$ are given, to optimum accuracy, as special-case output from the subroutine TLAMB, in a post-XLAMB direct call by VLAMB. The special case is signalled by the setting of the fifth input argument to -1, which causes the quantities in question to be computed in place of T', T'' and T'''.

10. Testing Rationale and Results

The subroutines TLAMB, XLAMB and VLAMB were tested for a wide range of data, starting with TLAMB, which is a self-contained procedure implementing a direct algorithm (no iteration necessary). Problems of accuracy with this subroutine seemed most likely to arise with input for which q was close to ± 1, or else x was close to ± 1

or zero, so testing was particularly thorough for data in these categories. Further, for $m = 0$ there are two transitions (from ellipse through 'series' to hyperbola), one each side of $x = 1$, and the regions in the vicinity of these transitions were tested the most carefully of all. The tests were entirely satisfactory, and Figure 1 was based on the output – its consistency with the Figure given by Lancaster *et al.* was an additional comfirmation of correct behaviour.

The testing of XLAMB followed a natural procedure for validating the iterative solution of an equation. Thus, the parameters of this subroutine are q, m and T, but instead of ranging over T the test data actually ranged over x. Then each 'true' x (for a given q and m) was the source of a nominal test value of T (via TLAMB), after which the testing of XLAMB could proceed, with the true x used merely as an accuracy evaluator, only one solution (in two-solution cases) being relevant to this evaluation. The testing was thorough, with values of q taken very close (and even equal) to ± 1, and (as already remarked in Section 8) for $m = 0$ it has shown that 13-digit accuracy is always achieved within three iterations.

A particular case was examined in detail, namely, for $q = -0.99$, $m = 0$ and $x \approx 0.159$ (nominal-test $T = 5.0$ and $\varphi \approx 1.9936\pi$), input for which Battin (1987, p. 342) found the worst convergence behaviour of the cases he tabulated. In fact, 14 iterations were found to be necessary in the modified version of Gauss's method that Battin used, just to get a solution to 8-digit accuracy; some comments on the modified-Gauss procedure appear in the Report. (Boltz (1984) claims to achieve faster convergence than Battin, but does no better in the worst cases compared.) Battin claims that methods of the Newton–Raphson type fail for such examples, but this is not so, in spite of the original remark of Lancaster and Blanchard cited for the claim (already commented upon here in Sections 6 and 7.2). Thus XLAMB gives values of ε (defined in Section 8), after zero, one and two iterations, respectively, of 0.043, 1.3×10^{-4} and 5.2×10^{-12}; after three iterations, ε is too small to detect, but it would evidently be around 4×10^{-34} in the absence of rounding error. (And the results for this example are no worse if the Halley process is replaced by the Newton–Raphson process.) However, this conspicuously good behaviour is largely due to the bilinear starter, so it was worth seeing what would happen if, for compatibility with Battin, x_0 was taken from its value for the circular orbit through P_1 and P_2. The formula for this is $x_0 = q(1 + q^2)^{-1/2}$, giving about -0.704 here. Then the values of ε, with the built-in Halley process, rose to 2.5, 0.23, 0.15 and 0.0023, for iterations up to the third, suggesting that two more iterations would be required for our usual criterion to be met. (Further, when the Halley process was replaced by the Newton–Raphson, the convergence was much slower still.) For this example it is clear that a good starting value is essential. (A different conclusion was reached in the Report, because the negative sign was overlooked in the value of x_0 given by a circular orbit.)

The testing of VLAMB was based on the idea that, for each Lambert solution, the output $V_{R,1}$ and $V_{T,1}$ of the subroutine can be combined with r_1, together with a polar reference angle (taken as zero), to constitute the four components of data (associated with the point P_1) appropriate for input to the subroutine PV2ELS (Gooding, 1988b).

This subroutine then generates the corresponding set of four universal two-dimensional orbital elements, one of which is τ, the time from pericentre (a conventional point if the orbit is circular). When τ is updated to $\tau + \Delta t$, with the other three elements unchanged, a nominal position (plus velocity) for the point P_2 can be obtained from ELS2PV, the subroutine inverse to PV2ELS. This position is specified as a radius vector and a polar angle, so the performance of VLAMB, and hence the success of the overall solution procedure, can be assessed by a direct comparison of these quantities with the input values r_2 and θ; we denote the (absolute values of) the differences, in r_2 and θ, by δr and $\delta\theta$. Ideally, looking at relative errors, we should like $\delta r/r_2$ and $\delta\theta/\theta$, which we can denote by δ_r and δ_θ, to be no greater than, say, 5×10^{-13} (on the PR1ME computer).

For a number of reasons, this 'ideal' requirement is too exacting. First, if the velocity at P_2 is very large, then a correspondingly large magnification of the relative rounding error may occur that is quite unavoidable. For δ_r, this magnification will be allowed for, in principle, if we do not automatically divide δr by r_2, but instead use r_R, where

$$r_R = \max(r_2, |V_{R,2}|\Delta t). \tag{27}$$

This does not fully compensate for the velocity effect, however, since Δt may have a large derivative with respect to the value of x determined by XLAMB; this secondary effect is allowed for if we replace Δt in (27) by $\max(\Delta t, |x\, \mathrm{d}(\Delta t)/\mathrm{d}x|)$.

With the velocity at P_2 allowed for in this way, the modified value of δ_r satisfied the 'ideal requirement' for all tests with $r_1 \geqslant r_2$, but a further complication with the test criteria was observed in tests with $r_2 > r_1$. The difficulty may be understood in relation to situations with $r_2 \gg r_1$, and with a value of x corresponding to a semi-major axis with a $\gg r_2$, since the determination of μ/a by PV2ELS is then inevitably inaccurate. A further empirical relaxation in the definition of δ_r (and similarly δ_θ) was introduced to compensate for this, and (as described in the Report) the difficulty was thereby resolved.

With δ_r and δ_θ adjusted in line with the last two paragraphs, it was found that δ_r always satisfied the 'ideal requirement', but that δ_θ could fail to do so when θ was less than about 1 radian. Replacement of the relative error (δ_θ) by the absolute error ($\delta\theta$) then always led to the requirement being satisfied, but this was hardly surprising, and in view of the desirability of maintaining relative accuracy for Lambert problems with small values of θ and Δt, it was important to know whether accuracy was lost in the Lambert procedure itself, or in the test procedure via PV2ELS and ELS2PV. Tests for very small values of θ and Δt have indicated that the accuracy loss was always in the test procedure. The explanation is that PV2ELS and ELS2PV cannot, by their design, maintain full accuracy in moving between close points on a given orbital path; in particular, this results from the reference made to pericentre, a point that is likely to be remote from the closely neighbouring points assumed.

With all the modifications to δ_r and δ_θ that have been described, their values remained below 5×10^{-13} in all the VLAMB tests carried out. In this testing, the same

range of values for θ was used as was (effectively) used for φ in testing XLAMB, with the same range of input-x used as source for the nominal values of Δt. Separate values of r_1 and r_2 were now provided, with values for r_2/r_1 covering the range from 10^{-6} to 10^6. Since the modified δ_r and δ_θ did not exceed 5×10^{-13}, over the entire range of x, θ and r_1/r_2, the VLAMB testing was considered to have been successful.

11. Conclusion

It is believed that the procedure described here is the first one published that solves Lambert's problem to maximal accuracy in all circumstances not involving a large number of elliptic revolutions. The achievement of this accuracy is largely due to the attention given to optimal starting formulae for the iteration variable (a topic that previous papers have almost totally neglected), and to the use of Halley's method for the actual iteration (which then converges rapidly in all circumstances, contrary to common belief). Care has also been taken to minimize rounding-error effects throughout the procedure.

The procedure is extremely efficient, the efficiency stemming largely from the computation of flight-time by the algorithm devised by Lancaster *et al.* (1966 and 1969). An important feature of this algorithm is that formulae particular to the ellipse and hyperbola are not eschewed when their use is appropriate. Power-series expansions are only used, to avoid rounding error, when the iteration variable lies within the range $\sqrt{0.6} < x < \sqrt{1.4}$. (There might be some gain in efficiency if, following the expertise of Battin (1987), we used continued fractions instead of power series, but for such a limited range of x the gain would be marginal.) It should be noted that the distinction between types of orbit is only made within the innermost subroutine of the procedure, so there is no loss of the universality that is inherent in the Lambert problem; this remark is in keeping with the author's published views on universal elements (Gooding, 1988b).

In testing the procedure it has been found that three iterations of the iteration process suffice for the effective elimination of error, when working to an accuracy of not more than 13 significant figures. When only 5- or 6-figure accuracy is needed, two iterations are adequate, and in most cases they will be a great deal more than merely 'adequate'.

Appendix A: Fortran-77 Subroutine TLAMB

```
SUBROUTINE TLAMB (M, Q, QSQFM1, X, N, T, DT, D2T, D3T)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
LOGICAL LM1, L1, L2, L3
DATA PI, SW /3.141592653589793D0, 0.4D0/
LM1 = N.EQ.-1
L1 = N.GE.1
L2 = N.GE.2
L3 = N.EQ.3
QSQ = Q*Q
XSQ = X*X
```

```
      U = (1D0 - X)*(1D0 + X)
      IF (.NOT.LM1) THEN
C           (NEEDED IF SERIES, AND OTHERWISE USEFUL WHEN Z = 0)
        DT = 0D0
        D2T = 0D0
        D3T = 0D0
      END IF
      IF (LM1 .OR. M.GT.0 .OR. X.LT.0D0 .OR. DABS(U).GT.SW) THEN
C         DIRECT COMPUTATION (NOT SERIES)
        Y = DSQRT(DABS(U))
        Z = DSQRT(QSQFM1 + QSQ*XSQ)
        QX = Q*X
        IF (QX.LE.0D0) THEN
          A = Z - QX
          B = Q*Z - X
        END IF
        IF (QX.LT.0D0 .AND. LM1) THEN
          AA = QSQFM1/A
          BB = QSQFM1*(QSQ*U - XSQ)/B
        END IF
        IF (QX.EQ.0D0.AND.LM1 .OR. QX.GT.0D0) THEN
          AA = Z + QX
          BB = Q*Z + X
        END IF
        IF (QX.GT.0D0) THEN
          A = QSQFM1/AA
          B = QSQFM1*(QSQ*U - XSQ)/BB
        END IF
        IF (.NOT.LM1) THEN
          IF (QX*U.GE.0D0) THEN
            G = X*Z + Q*U
           ELSE
            G = (XSQ - QSQ*U)/(X*Z - Q*U)
          END IF
          F = A*Y
          IF (X.LE.1D0) THEN
            T = M*PI + DATAN2(F, G)
           ELSE
            IF (F.GT.SW) THEN
              T = DLOG(F + G)
             ELSE
              FG1 = F/(G + 1D0)
              TERM = 2D0*FG1
              FG1SQ = FG1*FG1
              T = TERM
              TWOI1 = 1D0
    1         TWOI1 = TWOI1 + 2D0
              TERM = TERM*FG1SQ
              TOLD = T
              T = T + TERM/TWOI1
              IF (T.NE.TOLD) GO TO 1
C           (CONTINUE LOOPING FOR INVERSE TANH)
            END IF
          END IF
          T = 2D0*(T/Y + B)/U
          IF (L1 .AND. Z.NE.0D0) THEN
            QZ = Q/Z
            QZ2 = QZ*QZ
            QZ = QZ*QZ2
            DT = (3D0*X*T - 4D0*(A + QX*QSQFM1)/Z)/U
            IF (L2) D2T = (3D0*T + 5D0*X*DT + 4D0*QZ*QSQFM1)/U
            IF (L3) D3T = (8D0*DT + 7D0*X*D2T - 12D0*QZ*QZ2*X*QSQFM1)/U
          END IF
         ELSE
          DT = B
          D2T = BB
          D3T = AA
        END IF
       ELSE
C         COMPUTE BY SERIES
        U0I = 1D0
```

```
        IF (L1) U1I = 1D0
        IF (L2) U2I = 1D0
        IF (L3) U3I = 1D0
        TERM = 4D0
        TQ = Q*QSQFM1
        I = 0
        IF (Q.LT.5D-1) TQSUM = 1D0 - Q*QSQ
        IF (Q.GE.5D-1) TQSUM = (1D0/(1D0 + Q) + Q)*QSQFM1
        TTMOLD = TERM/3D0
        T = TTMOLD*TQSUM
C          (START OF LOOP)
    2   I = I + 1
        P = I
        U0I = U0I*U
        IF (L1 .AND. I.GT.1) U1I = U1I*U
        IF (L2 .AND. I.GT.2) U2I = U2I*U
        IF (L3 .AND. I.GT.3) U3I = U3I*U
        TERM = TERM*(P - 0.5D0)/P
        TQ = TQ*QSQ
        TQSUM = TQSUM + TQ
        TOLD = T
        TTERM = TERM/(2D0*P + 3D0)
        TQTERM = TTERM*TQSUM
        T = T - U0I*((1.5D0*P + 0.25D0)*TQTERM/(P*P - 0.25D0)
     A                                                - TTMOLD*TQ)
        TTMOLD = TTERM
        TQTERM = TQTERM*P
        IF (L1) DT = DT + TQTERM*U1I
        IF (L2) D2T = D2T + TQTERM*U2I*(P - 1D0)
        IF (L3) D3T = D3T + TQTERM*U3I*(P - 1D0)*(P - 2D0)
        IF (I.LT.N .OR. T.NE.TOLD) GO TO 2
C          (END OF LOOP)
        IF (L3) D3T = 8D0*X*(1.5D0*D2T - XSQ*D3T)
        IF (L2) D2T = 2D0*(2D0*XSQ*D2T - DT)
        IF (L1) DT = -2D0*X*DT
        T = T/XSQ
      END IF
      RETURN
      END
```

Appendix B: Fortran-77 Subroutine XLAMB

```
      SUBROUTINE XLAMB (M, Q, QSQFM1, TIN, N, X, XPL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.141592653589793D0, TOL=3D-7, C0=1.7D0)
      PARAMETER (C1=0.5D0, C2=0.03D0, C3=0.15D0, C41=1D0, C42=0.24D0)
      D8RT(X) = DSQRT(DSQRT(DSQRT(X)))
      THR2 = DATAN2(QSQFM1, 2D0*Q)/PI
      IF (M.EQ.0) THEN
C          SINGLE-REV STARTER FROM  T (AT X = 0) & BILINEAR (USUALLY)
        N = 1
        CALL TLAMB (M, Q, QSQFM1, 0D0, 0, T0, DT, D2T, D3T)
        TDIFF = TIN - T0
        IF (TDIFF.LE.0D0) THEN
          X = T0*TDIFF/(-4D0*TIN)
C           (-4 IS THE VALUE OF DT, FOR X = 0)
         ELSE
          X = -TDIFF/(TDIFF + 4D0)
          W = X + C0*DSQRT(2D0*(1D0 - THR2))
          IF (W.LT.0D0)
     A     X = X - DSQRT(D8RT(-W))*(X + DSQRT(TDIFF/(TDIFF + 1.5D0*T0)))
          W = 4D0/(4D0 + TDIFF)
          X = X*(1D0 + X*(C1*W - C2*X*DSQRT(W)))
        END IF
       ELSE
C          WITH MULTIREVS, FIRST GET T(MIN) AS BASIS FOR STARTER
        XM = 1D0/(1.5D0*(M + 5D-1)*PI)
        IF (THR2.LT.5D-1) XM = D8RT(2D0*THR2)*XM
        IF (THR2.GT.5D-1) XM = (2D0 - D8RT(2D0 - 2D0*THR2))*XM
C          (STARTER FOR TMIN)
        DO 1 I=1,12
```

```
      CALL TLAMB (M, Q, QSQFM1, XM, 3, TMIN, DT, D2T, D3T)
      IF (D2T.EQ.0D0) GO TO 2
      XMOLD = XM
      XM = XM - DT*D2T/(D2T*D2T - DT*D3T/2D0)
      XTEST = DABS(XMOLD/XM - 1D0)
      IF (XTEST.LE.TOL) GO TO 2
    1 CONTINUE
      N = -1
      RETURN
C        (BREAK OFF & EXIT IF TMIN NOT LOCATED - SHOULD NEVER HAPPEN)
C        NOW PROCEED FROM T(MIN) TO FULL STARTER
    2 CONTINUE
      TDIFFM = TIN - TMIN
      IF (TDIFFM.LT.0D0) THEN
        N = 0
        RETURN
C         (EXIT IF NO SOLUTION WITH THIS M)
       ELSE IF (TDIFFM.EQ.0D0) THEN
        X = XM
        N = 1
        RETURN
C         (EXIT IF UNIQUE SOLUTION ALREADY FROM X(TMIN))
       ELSE
        N = 3
        IF (D2T.EQ.0D0) D2T = 6D0*M*PI
        X = DSQRT(TDIFFM/(D2T/2D0 + TDIFFM/(1D0 - XM)**2))
        W = XM + X
        W = W*4D0/(4D0 + TDIFFM) + (1D0 - W)**2
        X = X*(1D0 - (1D0 + M + C41*(THR2 - 0.5D0))/(1D0 + C3*M)*
     C   X*(C1*W + C2*X*DSQRT(W))) + XM
        D2T2 = D2T/2D0
        IF (X.GE.1D0) THEN
          N = 1
          GO TO 3
        END IF
C         (NO FINITE SOLUTION WITH X > XM)
      END IF
C
    END IF
C       (NOW HAVE A STARTER, SO PROCEED BY HALLEY)
    5 CONTINUE
      DO 4 I=1,3
      CALL TLAMB (M, Q, QSQFM1, X, 2, T, DT, D2T, D3T)
      T = TIN - T
      IF (DT.NE.0D0) X = X + T*DT/(DT*DT + T*D2T/2D0)
    4 CONTINUE
      IF (N.NE.3) RETURN
C       (EXIT IF ONLY ONE SOLUTION, NORMALLY WHEN M = 0)
      N = 2
      XPL = X
C       (SECOND MULTI-REV STARTER)
    3 CALL TLAMB (M, Q, QSQFM1, 0D0, 0, T0, DT, D2T, D3T)
      TDIFF0 = T0 - TMIN
      TDIFF = TIN - T0
      IF (TDIFF.LE.0) THEN
        X = XM - DSQRT(TDIFFM/(D2T2 - TDIFFM*(D2T2/TDIFF0
     A                                          - 1D0/XM**2)))
       ELSE
        X = -TDIFF/(TDIFF + 4D0)
        W = X + C0*DSQRT(2D0*(1D0 - THR2))
        IF (W.LT.0D0) X =
     A   X - DSQRT(D8RT(-W))*(X + DSQRT(TDIFF/(TDIFF + 1.5D0*T0)))
        W = 4D0/(4D0 + TDIFF)
        X = X*(1D0 + (1D0 + M + C42*(THR2 - 0.5D0))/(1D0 + C3*M)*
     A                                  X*(C1*W - C2*X*DSQRT(W)))
        IF (X.LE.-1D0) THEN
          N = N - 1
C         (NO FINITE SOLUTION WITH X < XM)
        IF (N.EQ.1) X = XPL
      END IF
      END IF
      GO TO 5
      END
```

Appendix C: Fortran-77 Subroutine VLAMB

```
      SUBROUTINE VLAMB (GM, R1, R2, TH, TDELT, N, VR11, VT11,
     1 VR12, VT12, VR21, VT21, VR22, VT22)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI = 3.141592653589793D0, TWOPI = 2D0*PI)
      M = TH/TWOPI
      THR2 = TH/2D0 - M*PI
      DR = R1 - R2
      R1R2 = R1*R2
      R1R2TH = 4D0*R1R2*DSIN(THR2)**2
      CSQ = DR**2 + R1R2TH
      C = DSQRT(CSQ)
      S = (R1 + R2 + C)/2D0
      GMS = DSQRT(GM*S/2D0)
      QSQFM1 = C/S
      Q = DSQRT(R1R2)*DCOS(THR2)/S
      IF (C.NE.0D0) THEN
        RHO = DR/C
        SIG = R1R2TH/CSQ
       ELSE
        RHO = 0D0
        SIG = 1D0
      END IF
      T = 4D0*GMS*TDELT/S**2
      CALL XLAMB (M, Q, QSQFM1, T, N, X1, X2)
C           PROCEED FOR SINGLE SOLUTION, OR A PAIR
      DO 1 I=1,N
      IF (I.EQ.1) THEN
        X = X1
       ELSE
        X = X2
      END IF
      CALL TLAMB (M, Q, QSQFM1, X, -1, UNUSED, QZMINX, QZPLX, ZPLQX)
      VT2 = GMS*ZPLQX*DSQRT(SIG)
      VR1 = GMS*(QZMINX - QZPLX*RHO)/R1
      VT1 = VT2/R1
      VR2 = -GMS*(QZMINX + QZPLX*RHO)/R2
      VT2 = VT2/R2
      IF (I.EQ.1) THEN
        VR11 = VR1
        VT11 = VT1
        VR12 = VR2
        VT12 = VT2
       ELSE
        VR21 = VR1
        VT21 = VT1
        VR22 = VR2
        VT22 = VT2
      END IF
    1 CONTINUE
      RETURN
      END
```

References

Battin, R. H.: 1987, *An Introduction to The Mathematics and Methods of Astrodynamics*, AIAA, New York.

Boltz, F. W.: 1984, 'Second-order p-Iterative Solution of the Lambert/Gauss Problem', *J. Astr. Sci.*, **32**, 475–485.

Gooding, R. H.: 1988a, 'On the Solution of Lambert's Orbital Boundary-Value Problem', RAE Technical Report 88027.

Gooding, R. H.: 1988b, 'On Universal Elements, and Conversion Procedures to and from Position and Velocity', *Celest. Mech.*, **44**, 283–298.

Gooding, R. H. and Odell, A. W.: 1988, 'The Hyperbolic Kepler Equation (and the Elliptic Equation revisited)', *Celest. Mech.*, **44**, 267–282.

Lancaster, E. R. and Blanchard, R. C.: 1969, 'A Unified Form of Lambert's Theorem', NASA Technical Note D-5368.

Lancaster, E. R., Blanchard, R. C. and Devaney, R. A.: 1966, 'A Note on Lambert's Theorem', *J. Spacecraft Rockets*, **3**, 1436–1438.

Odell, A. W. and Gooding, R. H.: 1986, 'Procedures for Solving Kepler's Equation', *Celest. Mech.*, **38**, 307–334.

Sarnecki, A. J.: 1988, '"Minimal" Orbital Dynamics', *Acta Astronautica*, **17**, 881–891.

Sun, F. T., Vinh, N. X. and Chern, T. J.: 1987, 'Analytical Study of the Solution Families of the Extended Godal's Time Equation for Lambert's Problem', *J. Astr. Sci.*, **35**, 213–234.

Taff, L. G. and Randall, P. M. S.: 1985, 'Two Locations, Two Times and the Element Set', *Celest. Mech.*, **37**, 149–159.

Volk, O.: 1980, 'Johann Heinrich Lambert and the Determination of Orbits for Planets and Comets', *Celest. Mech.*, **21**, 237–250.

Wintner, A.: 1941, *The Analytical Foundations of Celestial Mechanics*, Princeton, New Jersey.