ON UNIVERSAL ELEMENTS,
AND CONVERSION PROCEDURES
TO AND FROM POSITION AND VELOCITY

R. H. GOODING

Royal Aerospace Establishment, Farnborough, Hants, England

Abstract. An element set is advocated that is familiar (in traditional terms), and yet applicable to every type of conic-section orbit without loss of accuracy. It is not free of singularity, but this is not a serious deficiency. Conversion procedures, to and from position and velocity, are outlined, with Fortran-77 listings appended. Tests have indicated that the errors in the pair of procedures are minimal, accuracy being limited only by computer precision and the (fixed) number of iterations used in the Kepler-equation solutions.

1. Introduction

There has long been the goal, in celestial mechanics, of subsuming the solutions of the classical two-body problem, particular to the ellipse, parabola and hyperbola, in a universal solution, valid for the rectilinear orbit of each type as well as the general orbit. Steps towards this goal have been taken by Sundman (1912), Stumpff (1947), Goodyear (1965), Herrick (1965), Pitkin (1965) and Shepperd (1985), among others, and some elegant mathematical formulations have been achieved. The aesthetic attractions of the universal approach have to be weighed against a number of computational disadvantages, however, and the latter are often disregarded.

An important distinction must be made between universality associated with orbital elements, on the one hand, and the formulae and working variables of computing procedures, on the other. The case for universal elements is undeniable, but it does not automatically extend to algorithms such as those involved in the conversion of elements to and from position and velocity; the user of such an algorithm is naturally concerned with its generality, as well as with its accuracy and efficiency, but a requirement for generality does not imply that the algorithm has to be internally 'blind' to orbit type.

The foregoing distiction is at the heart of the present paper, which is a shortened version of a recent RAE Report (Gooding, 1987). Section 2 discusses what is actually meant by a set of universal (or universally applicable) elements, and introduces the set to be used in the rest of the paper; this set suffers from certain singularities, but the singularities are found to lead to no real difficulty in the conversion algorithms. Based on the assumed element set, Section 3 describes a computing procedure for the conversion to position and velocity; this involves, in particular, separate algorithms for solving Kepler's equation and the corresponding hyperbolic equation, as currently described by Gooding and Odell (1988). Section 4 describes the reverse procedure; here the algorithms for ellipse and hyperbola need to be formulated with great

care, since serious inaccuracy can arise if a formula for hyperbolic orbits is based on direct transmutation of the corresponding elliptic formula. Section 5 of the paper is concerned with the testing and performance of the two conversion procedures, each of them having been implemented by a pair of Fortran-77 subroutines, one to cover the two-dimensional (in-plane) part of the conversion and the other the three-dimensional aspects.

There are situations in which the orbital-element singularities present more difficulty than in the conversion algorithms. Section 6 indicates how such difficulties may be dealt with.

2. A Set of Universally Applicable Elements

We seek to define a set of universally applicable elements for motion in unperturbed orbits about a centre of Newtonian attraction of strength μ. We restrict our attention to non-redundant element sets, for convenience denoting by ζ the sextuple of quantities that such a set must comprise; further, we make the usual assumption that the first five elements of ζ define the orbital path, whilst the sixth is a phase specifier. Since our main concern is with computing procedures for converting from orbital elements to position and velocity, and vice versa, we let $\mathbf{x}$ denote the sextuple of components of position and velocity in a convenient system of rectangular coordinates, with origin at the attraction centre; we introduce f as the function converting from ζ to $\mathbf{x}$, so that

$$\mathbf{x} = f(\zeta). \tag{1}$$

We shall also refer to the Jacobian (partial-derivative) matrix of $\mathbf{x}$ with respect to ζ, and denote it by $\mathbf{J}$.

If a particular element set can be chosen that covers every type of orbit, then in principle we regard these elements as universal. It is implied that the function f is surjective, with range covering all possible $\mathbf{x}$, but this is not enough. We also require that $\mathbf{J}$ is defined (exists) over the domain of valid ζ, with no occurrence of discontinuity or infinity.

Ideally, f would be injective as well as surjective, so that it would have a unique inverse, with the matrix $\mathbf{J}$ never singular. Since non-singularity seems to be incompatible with universality, however, we must give up the extra requirement, but for many purposes (in solving Lambert's problem, for example) this is of little consequence. From the surjective property, we can always define a (non-unique) function, f^{-1}, such that the composition of f^{-1} followed by f is the identity over $\mathbf{x}$-space, and this is enough to demand of the function f^{-1}. Looking ahead to Section 5, however, we can express this relation more conveniently (and symmetrically) as a property based on ζ-space, *viz.*

$$ff^{-1}f = f. \tag{2}$$

We now proceed to the identification of a particular set of universally applicable

elements. Our starting point is the set $(a, e, i, \Omega, \omega, M)$ of familiar elements used for elliptic orbits. The elements i, Ω and ω, which operate as Euler angles relating the orbital plane to the chosen axis system, are already universal, applying as well to parabolas and hyperbolas as to ellipses. The element M is manifestly not universal, on the other hand, since it is identically zero in the parabolic limit. It becomes universal on divison by n (mean motion), however, the result being a new element (time from perifocus) that we denote by τ; clearly, τ is just the negative of the element T, traditionally used as an alternative to M. The four elements so far considered are all involved in indeterminacies (non-uniqueness) due to singularity (and the associated non-injectivity of the function f), the sources of singularity being rectilinear orbits, circular orbits, and orbits for which $i = 0$ or $i = \pi$.

It just remains to consider the first two of the original elements: the combination of a and e cannot be universal, since a parabola's perifocal distance, given by $q = a(1 - e)$, and parameter, given by $p = a(1 - e^2)$, would not then be defined. (There is, in fact, an unnecessary degradation of accuracy for any orbit with an eccentricity close to unity.) If we replace e by q, however, the new pair of elements (a and q) would be universal if we allowed a to be infinite. To avoid this difficulty, we replace a by α, where

$$\alpha = \mu/a, \tag{3}$$

and then the set $(\alpha, q, i, \Omega, \omega, \tau)$ is a universal set. It is noted that the definition of α by (3) maintains the desirable feature of the original element set that makes one element a function only of energy; indeed α is twice the negative energy per unit mass, since

$$V^2 - 2\mu/r = -\alpha. \tag{4}$$

The incorporation of the factor μ in (3) leads to certain simplifications in computing; as can be seen from (4), it also permits the element α to be well defined in the ultimate limiting case in which $\mu = 0$. It would not be satisfactory to replace q by p, since (because $\partial q/\partial p = 1/2e$, for a fixed value of a) a circular orbit would then involve infinities in $\mathbf{J}$.

In terms of α and q, the following complete classification of orbits becomes possible:

	$\alpha > 0$	$\alpha = 0$	$\alpha < 0$
$q > 0$	general ellipse	general parabola	general hyperbola
$q = 0$	rectilinear ellipse	rectilinear parabola	rectilinear hyperbola

Figure 1 shows how q varies with α/μ ($= 1/a$) for selected values of e. The curves are all rectangular hyperbolas, and it is easily seen how rectilinear orbits (points on the horizontal axis) may be distinguished from parabolic orbits (points on the vertical axis), though $e = 1$ in both cases.
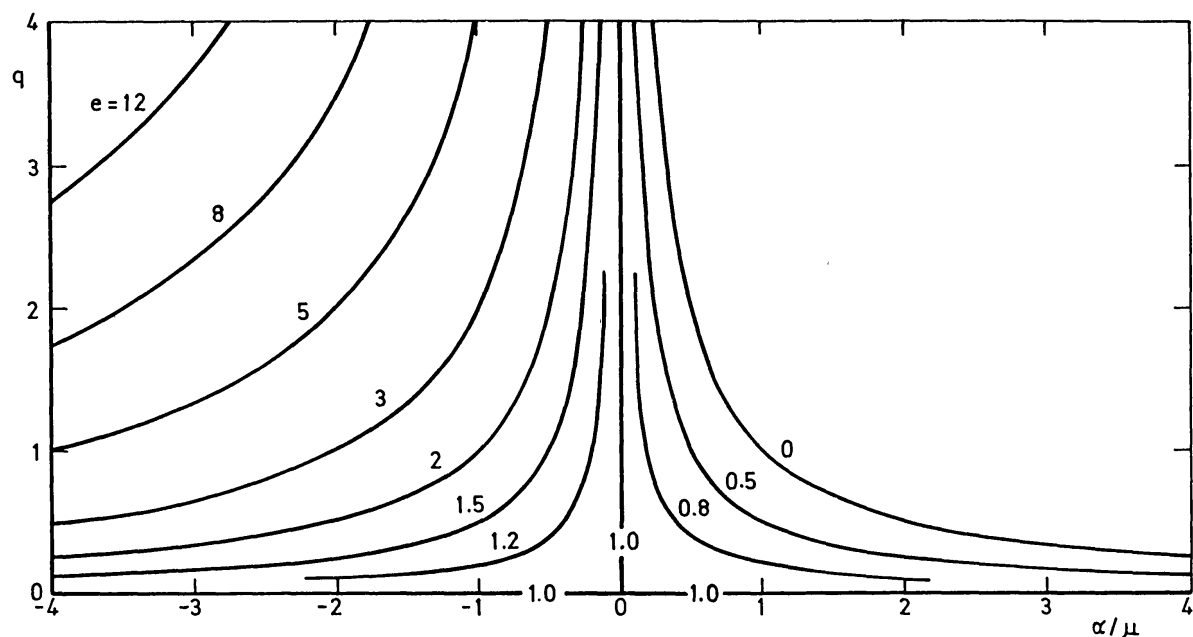
Fig. 1. Variation of q with $\alpha/\mu(=1/a)$ for selected values of e.

3. Conversion of Elements to Position and Velocity

The procedure to be described converts from the assumed element set $(\alpha, q, i, \Omega, \omega, \tau)$ to position and velocity $(x, y, z, \dot{x}, \dot{y}, \dot{z})$. It has been implemented by a pair of Fortran-77 subroutines, listed in Appendices A and B. Subroutine ELS2PV (Appendix A) covers the purely two-dimensional (in-plane) part of the conversion, in which the polar coordinates r and u (u being 'argument of latitude', assuming the normal interpretation of the axis system), together with the radial and transverse velocity components V_R and V_T, are derived from the elements α, q, ω and τ, plus μ. The definition of V_T is such that $V_T \geqslant 0$; also it is assumed that $r > 0$, avoiding the polar-coordinate singularity at $r = 0$, though ELS2PV operates accurately for all points of a rectilinear orbit other than the centre of attraction itself. Subroutine ELS3PV (Appendix B) first calls ELS2PV, and then uses the angles u, i and Ω to perform the appropriate rotations to complete the conversion.

Details of the computing procedure will be omitted here. The relevant formulae have been given, without proofs, by Gooding (1987), and further detail can be obtained from text-books. Some comments on particular aspects of the procedure are called for, however.

First, in much of ELS2PV, different algorithms are used for the ellipse, hyperbola and parabola, the control parameter being the sign of α. The function procedures EKEPL and SHKEPL are used to solve Kepler's equation and a reformulated hyperbolic equation, respectively, as described by Gooding and Odell (1988); the guaranteed accuracies (relative truncation error) for these procedures, in two iterations of a quartic convergence process, are 14 decimal digits for E (eccentric anomaly) and 20

digits for $S(=\sinh H$, where H is the hyperbolic anomaly). For the parabola, Barker's equation, in the form

$$d^3 + 6\mu qd = 6\mu^2\tau, \tag{5}$$

is solved directly for d, where $d = r\dot{r}$. Care has been taken to obtain maximum accuracy in the solution of this cubic equation, in particular by the use of a refined cube root.

The next point concerns the value of v (true anomaly), computed as a working variable of ELS2PV. For hyperbolas and parabolas, we must have $|v| < \pi$, so that the subroutine can accommodate an arbitrary value of ω unambiguously, with included multiples of 2π carried directly into u. For ellipses, on the other hand, an arbitrary value of v can arise, reflecting the number of orbital periods present in τ, so that the mapping $(\omega, \tau) \to u$ is no longer one-one. This is an aspect of the non-injective nature of the function f of Section 2, quite apart from the singularity issue, and it will be referred to again in Section 4.

As a third point, we note that, for the ellipse, r and v are computed from formulae in $\sin\frac{1}{2}E$ and $\cos\frac{1}{2}E$, but that the corresponding formulae (in $\sinh\frac{1}{2}H$ and $\cosh\frac{1}{2}H$) are not available for the hyperbola, as it is $S(=\sinh H)$ that is provided by SHKEPL. There is no difficulty, however, as both r and v can be expressed in terms of $C - 1$, where $C = \cosh H$; to minimize rounding error, we compute

$$C = \sqrt{(1 + S^2)},$$

followed by

$$C - 1 = S^2/(C + 1).$$

A final comment may be helpful in interpreting the listings in the Appendices (it applies in Section 4 as well as here). As a corollary of the appearance of μ in definition (3), an appropriate (integral) power of μ is naturally asociated with other quantities also; thus the Fortran variables E and M refer to μe and μM, respectively, not just e and M. (This scaling role of μ preserves the validity of both the conversion procedures when $\mu = 0$. When $\mu < 0$, implying the operation of a repulsive force with $e \leqslant -1$, the procedures still work appropriately, though a large rounding error can arise when $e \approx -1$; it would be necessary to replace the element q, equal to $a(1 - e)$, by $a(1 + e)$ to avoid this error.)

4. Conversion of Position and Velocity to Elements

We are now concerned with the procedure for converting from position and velocity $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ to the element set $(\alpha, q, i, \Omega, \omega, \tau)$. Whereas the procedure described in Section 3 constitutes the unambiguous function f of Section 2, the procedure to be described here merely defines a particular inverse function, f^{-1}, of the many satisfying (2). It has been implemented by a pair of Fortran-77 subroutines, listed in Appendices C and D, that are the natural counterparts of those from Section 3. Subroutine

PV2ELS (Appendix C) covers the purely two-dimensional part of the conversion, computing α, q, ω and τ from r, u, V_R and V_T (together with μ), whilst PV3ELS (Appendix D) implements the overall procedure, deriving i and Ω as well as the input required for PV2ELS, which it calls.

Mainly because of the semi-arbitrary nature of the procedure, the f^{-1} subroutines are a good deal more complicated than the f subroutines, and more extensive commentary is necessary – there is one simplification, on the other hand, since there is no longer a transcendental equation to be solved. The first comment refers to the control parameter, in PV2ELS, for the type of orbit. This is α (as in ELS2PV), computed from (4). To minimize arbitrariness, the orbit is only deemed to be parabolic if α is exactly zero, and this condition is very unlikely to occur in practice; because of rounding error, this is true even when f^{-1} follows an f that operated on an exactly parabolic orbit. There is no lessening of accuracy in the handling of near-parabolic ellipses and hyperbolas, however, so it is only the efficiency associated with the parabolic formulae that is lost; if this were considered important, then PV2ELS could be modified to decree 'parabolic orbit' whenever $|r\alpha/\mu|$ is less than some suitable criterion value.

A similar situation (still in PV2ELS) applies to the recognition of a circular orbit, which only happens if an exactly zero value of e (which is just a working variable) is computed. This special case has to be covered because it constitutes a singularity, with the values of ω and τ both becoming indeterminate; PV2ELS sets τ and v (another working variable) arbitrarily to zero, after which ω is set to $u - v$ as with every other type of orbit. There is a degree of arbitrariness associated with all elliptic orbits, of course, due to the non-injective nature of the mapping $(\omega, \tau) \rightarrow u$, as remarked in Section 3. The two obvious options, for PV2ELS, were to select for minimum $|\tau|$ or minimum $|\omega|$. The former is the better choice, because it reduces rounding error in certain circumstances, in particular for near-rectilinear ellipses, and this option has been implemented in the subroutine; the other option can be obtained, however, by changing the value of a built-in logical variable (L).

It was remarked in Section 1 that the use of directly parallel formulae, for the ellipse and hyperbola, can lead to inaccuracy. The reference was to rounding error, and a good example of such error arises with the computation of e, required so that q can be computed from

$$q = p/(1 + e). \tag{6}$$

If we define

$$c = rV^2/\mu - 1$$

and

$$s = rV_R|\alpha|^{1/2}/\mu,$$

then the elliptic interpretation is that $c = e \cos E$ and $s = e \sin E$, so that e is given by

$e^2 = c^2 + s^2$. For the hyperbola, however, the interpretations of c and s are as $e \cosh H$ and $e \sinh H$, so that the parallel formula is $e^2 = c^2 - s^2$. The potential sensitivity to rounding error, when $|H|$ is large, is obvious, yet this is a formula frequently given in textbooks. The formula appropriate for hyperbolas is simply $e^2 = 1 - \alpha p / \mu$, where (since α is negative) it is really an addition, not a subtraction, that is implied. This formula would, in turn, be inappropriate for ellipses (cf. the remark in Section 2 that α and p, as opposed to α and q, would be an unsatisfactory pair of elements for orbits approaching circularity), and PV2ELS uses the optimum formula for each case.

The remaining comments refer to the computation of the quantities i, Ω and u by subroutine PV3ELS, prior to the calling of PV2ELS. The main problem is singularity again, though (as in PV2ELS) there is the additional source of arbitrariness associated, in particular, with the convention that the value determined for Ω should always satisfy $-\pi < \Omega \leqslant \pi$.

The predominant singularity relates to rectilinear orbits, and it is deemed to arise only when the angular-momentum vector is exactly zero. As no 'orbital plane' is then defined, it was necessary, for the operation of PV3ELS, to fabricate one, and it was decided to do this, in principle, by taking the plane that contains the orbit and for which $i = \frac{1}{2}\pi$. For an 'axial orbit', perpendicular to the reference plane, there is a subsidiary singularity, however, with the general principle inadequate to define a unique plane; for definiteness in this case, the value of Ω is decreed to be zero.

For the general (non-rectilinear) orbit, superficially the only difficulty arises with the singularity that occurs when the orbital plane exactly coincides with the reference plane, and this is easily dealt with, again by decreeing Ω to be zero – this is just like setting ω and τ to zero (in PV2ELS) for an exactly circular orbit. But there is another difficulty; it arises with near-rectilinear orbits, for which the rounding error in computing the components of the angular-momentum vector, $\mathbf{h}$, can be serious. The problem here disappears at once, however, if instead of just computing $\mathbf{h}(= \mathbf{r} \times \dot{\mathbf{r}})$ we compute

$$r^2 \mathbf{h} = \mathbf{r} \times [(\mathbf{r} \times \dot{\mathbf{r}}) \times \mathbf{r}];$$

further detail has been given by Gooding (1987).

5. Testing of the Computing Procedures

The basic philosophy for testing was that the two conversion procedures, as implemented by the subroutines ELS3PV and PV3ELS, would be used to test each other. The validity of this philosophy emanated from two considerations: first, that the subroutines were essentially independent, with (for example) Kepler's equation only arising in a subroutine subordinate to ELS3PV and rectilinear orbits only having to be recognized in PV3ELS; secondly, that advantage could be taken of the different formulae used for different types of orbit, to make a careful study of continuity across the transition lines. The obvious property to test, in the notation of Section 2, was that

ff^{-1} (the composition of f^{-1} followed by f) should be the identity over **x**-space, with no corresponding requirement for $f^{-1}f$; but the need to make the testing systematic and efficient pointed to the use of input from ζ-space rather than **x**-space, and this is why the testing was actually based on the property specified by Equation (2). The testing has been restricted to the verification of (2), to within tolerable rounding error, at fixed instants in time, but this restriction does not limit the efficacy of the subroutines or the testing, since to proceed from **x** at t_0, say, to **x** at t_1, we merely have to increment the τ-component of $\zeta_0 = f^{-1}(\mathbf{x}_0)$, by $t_1 - t_0$, before deriving $\mathbf{x}_1$ from $f(\zeta_1)$.

To attach a meaning to 'tolerable rounding error', we separate the sextuple **x** into the pair of vectors **r** and **ṙ**. Also, for each 'input' **x** (derived from an actual input ζ), we write $ff^{-1}(\mathbf{x})$ as $\mathbf{x} + \Delta\mathbf{x}$. Then the relative errors in the final position and velocity are taken to be $|\Delta\mathbf{r}|/r$ and $|\Delta\dot{\mathbf{r}}|/V$, respectively, where $r = |\mathbf{r}|$ and $V = |\dot{\mathbf{r}}|$. It might be hoped that the ff^{-1} operation would not produce relative errors more than about a (decimal) order of magnitude greater than the limiting precision of the computer used; since this was a PRIME 750, of limiting accuracy (double precision) about 10^{-14}, it was therefore reasonable to look for a maximum relative error no greater than 2×10^{-13}. This goal has been met for $|\Delta\mathbf{r}|/r$ without difficulty, but for $|\Delta\dot{\mathbf{r}}|/V$ only in a modified manner. Before explaining this, we remark on the range of test cases covered.

There are seven degrees of freedom associated with the input to ELS3PV, since μ is an argument as well as the sextuple ζ. It was obviously impracticable to carry out exhaustive tests over a seven-dimensional space, so most of the testing concentrated on the in-plane algorithms, using fixed values of i, Ω and ω. In these tests, the quantities μ, α, q and τ had to be covered, but it was not necessary to vary them all independently, if the assumption was made that orbits of the same shape produce, at corresponding points, relative errors of the same order of magnitude. Thus, orbits having the same value of $\alpha q/\mu$ may be regarded as equivalent in shape, this being the value of $1 - e$, whilst points with the same value of $\alpha^3 \tau^2/\mu^2$ may be regarded as in correspondence, this being the value of M^2 for elliptic orbits. On this basis, it was legitimate, for non-rectilinear orbits, to test with fixed values of μ and q, taken (arbitrarily) as 64 and 1 respectively; these tests, with two degrees of freedom (α and τ), then had to be supplemented by one-degree tests (on α only) for rectilinear orbits, with q now set to zero and τ given a fixed value. A wide range of values for α and τ was covered: α varied from an extreme negative value of -10^{20} to its maximum permissible value of 64 (for the given μ and q), with test values clustered in particular around zero (parabolic orbit) and close to 64 (circular orbit); test values for τ also clustered about zero (perifocus), with a maximum value of 10^{20}.

The essentially in-plane tests, just described, were carried out with the fixed value of $\frac{1}{4}\pi$ for i. Tests for a wide range of i were obviously important, however, in particular with values clustered close to the limiting values of 0 and π; these tests were carried out with α varied over its full range, but τ (as well as the other quantities) held fixed. Finally, it seemed enough to carry out a limited (non-systematic) set of tests with the values of Ω and ω varied.

As already indicated, all the tests met the goal set for position error, as measured by $|\Delta\mathbf{r}|/r$. The goal was also met for velocity error, as measured by $|\Delta\dot{\mathbf{r}}|/V$, in most of the tests, but it was not met in a number of cases involving proximity to the apofocus of near-rectilinear ellipses. Since V is very small in such circumstances, the requirement becomes severe, but failure to achieve it can be ascribed to a specific technical point: when E is close to π, it is impossible to restrict the relative error in $\sin E$; but V_R is proportional to $\sin E$, so when $\dot{\mathbf{r}}$ is dominated by V_R (because $V_T \approx 0$) the loss of (relative) accuracy in $\dot{\mathbf{r}}$ is inevitable; another way of putting this point is that $\delta V/\delta\tau$ is large in comparison with V/τ, so the relative error in computing τ (within PV2ELS) gets magnified in the subsequent return to velocity. If we measure relative error by $|\Delta\dot{\mathbf{r}}|/W$, however, where W is $\max(V, \sqrt{\alpha})$, not just V, then the goal of 2×10^{-13} is met in all cases. A similar problem might have been expected with position error when r is very small; but for values of E close to zero, the relative error in $\sin E$ is kept low, whilst values of E close to 2π, 4π, etc, cannot arise with the composite function $f(f^{-1}(\mathbf{x}))$, since f^{-1} selects for minimum τ, as explained in Section 4.

6. On the Singularities

Not counting the rectilinear-orbit singularity (defined by $q = 0$, but of little practical interest), there are three singularities that have long been recognized as possible sources of difficulty in computational analysis based on the usual elliptic elements: the singularity at $e = 0$; and the pair of singularities at $i = 0$ and $i = \pi$. It is easy to transform the elements so as to eliminate any one of these singularities, and it is not hard to find a transformation that simultaneously eliminates the eccentricity singularity and one of the inclination singularities – this is achieved with the 'equinoctial elements' of Broucke and Cefola (1972), for example. The elimination of all three singularities together is much more difficult, however, though it was achieved by Cohen and Hubbard (1962) with the element set defined by

$$q_0 = p^{1/4} \cos\tfrac{1}{2}i \cos\tfrac{1}{2}(\Omega + \omega + \sigma), \qquad q_1 = p^{1/4} \sin\tfrac{1}{2}i \cos\tfrac{1}{2}(\Omega - \omega - \sigma),$$

$$q_2 = p^{1/4} \sin\tfrac{1}{2}i \sin\tfrac{1}{2}(\Omega - \omega - \sigma), \qquad q_3 = p^{1/4} \cos\tfrac{1}{2}i \sin\tfrac{1}{2}(\Omega + \omega + \sigma),$$

$$e_x = e \cos\sigma \qquad \text{and} \qquad e_y = -e \sin\sigma,$$

where σ is the mean anomaly at epoch.

The trouble with the partial or total elimination of singularity, in element sets such as the ones just referred to, is that it only applies to elliptic orbits and is incompatible with universal applicability. By the same token, there is no simple transformation of the universal elements recommended here that eliminates singularity whilst retaining universality. We have already seen, on the other hand, that singularity is of little consequence so far as conversion procedures are concerned. But up to now we have only been concerned with fixed values of the relevant elements; when variations in these elements are considered, we cannot dismiss the singularity question so lightly.

Suppose we have a large computer program involving differential changes in the values of a set of orbital elements subject to singularity; these changes may arise, in particular, from the evaluation of perturbation formulae, or from the use of observations in orbit determination. If the program is used with an element set in the vicinity of a singularity, it is likely that unacceptable error will result, and the first time this occurs it may seem that we have to rewrite the program in terms of a different (singularity free) element set. This could be a daunting prospect, and an especially unattractive one if it could be foreseen that the efficiency of the new progam would be significantly reduced by the complexity of the new elements. However, such a radical rewrite is almost certainly unnecessary, as it should be possible to maintain the original elements with only a short-term switch to non-singular equivalents.

To illustrate the principle, consider just the singularity associated with $e = 0$. For simplicity, we suppose that the computer program has been written in terms of the usual elliptic elements, rather than our universally applicable set, but this is a point of minor consideration. Unless e is exactly zero, we have been able to compute changes in all the elements, including superficially meaningless values of $\delta\omega$ and δM. Applying these changes directly could lead to serious error in e, ω and M, attributable to truncation effects in the underlying Taylor expansions, but such error could be avoided by just a temporary switch from the elements e, ω and M to ξ, η and U, where

$$(\xi, \eta, U) = (e \cos \omega, \, e \sin \omega, \, M + \omega);$$

the switch is implemented on the basis that e', ω' and M', the new elements we require, are recoverable from

$$e' \cos \omega' = \xi' = \xi + \delta\xi = (e + \delta e) \cos \omega - (e \, \delta\omega) \sin \omega,$$
$$e' \sin \omega' = \eta' = \eta + \delta\eta = (e + \delta e) \sin \omega + (e \, \delta\omega) \cos \omega$$

and

$$M' + \omega' = U' = U + \delta U = (M + \delta M) + (\omega + \delta\omega).$$

Truncation error can be entirely avoided by this artifice, with no difficulty even at the singularity itself if $e \, \delta\omega$ is available as a composite entity. There remains the lesser problem of rounding error, in particular in the computation of partial derivatives for use in orbit determination (Gooding and Tayler, 1968). Thus the quantity $e^{-1}(\partial\theta/\partial\omega - \partial\theta/\partial M)$, where θ is an arbitrary observed quantity, will be required in place of $\partial\theta/\partial\omega$, to reflect the introduction of U and the use of the composite $e \, \delta\omega$, but it should be possible to derive this combination derivative, to full accuracy, by minimal modification of the hypothetical computer program. In regard to the application of perturbation formulae, the procedure for singularity avoidance has been indicated before, by Gooding (1983).

7. Conclusion

The only truly 'universal' elements are perhaps the epochal components of position and velocity, and papers such as Shepperd's (1985)* give elegant algorithms for the propagation (or transition) of these from one instant of time to another. Only one element of the traditional type figures in such algorithms, namely, the energy-equivalent inverse of the semi-major axis. However, total universality is not available, even with an approach as general as this, since the generalized anomaly variable has to be range-restricted when an elliptic orbit is identified and more than a single revolution within this ellipse is involved. Though textbooks increasingly reflect the attractions of the very general approach, they also continue to recognize the utility of traditional element sets, and in particular of sets for which five of the six elements are independent of orbital position. It has been the main objective of the present paper to establish that the most familiar of all element sets is, with only slight modification, of universal application, so long as the necessary conversion procedures (to and from position and velocity) are carefully programmed. In particular, optimum numerical accuracy can only be maintained if the procedures respect the necessity for different types of formulae to be employed internally, according to the type of orbit.

The universality objective (for an element set) is not compatible with freedom from singularity, but in the majority of applications singularity is either of little consequence or can be dealt with easily. Hence the existence of singularities for a universally applicable element set should not be regarded as a major defect.

The four Fortran-77 subroutines that implement the conversion procedures described in the paper are listed in Appendices A to D. They could be used, for example, in the derivation of smoothly varying elements for a comet that changes its orbital type under the influence of Jupiter and Saturn. At the Royal Aerospace Establishment, however, the main application to date has been in the solution of Lambert's problem; the RAE solution (Gooding, 1988) yields accurate velocities at the two points specified in the problem, and the conversion procedures permit these to be related to each other via universal elements.

* Two points in this generally excellent paper illustrate the weaknesses of computing procedures that are almost totally blind to orbital type. First, Shepperd recommends the use of a certain universal variable, u, in terms of which an 'intermediate parameter' q, where $q = \alpha u^2/(1 + \alpha u^2)$, is the argument of a particular continued-fraction expansion, $G_5(q)$. For elliptic orbits, q is actually $\sin^2 \frac{1}{4}\Delta E$, where $\Delta E = E - E_0$, so that $q_{max} = 0.5$, as Shepperd remarks. But for this value of q, 18 iterations of the recursive evaluation of $G_5(q)$ are needed to give 14-decimal-digit accuracy, whereas the same accuracy can be obtained with much greater efficiency if the G_5 function is evaluated, in a specifically elliptic formulation, as $15(6\Delta E - 8 \sin \Delta E + \sin 2\Delta E)/96 \sin^5 \frac{1}{2}\Delta E$. The other point is more serious, in that Shepperd suggests that use of u as argument of a universal Kepler equation automatically eliminates the slow-convergence problem noted, in particular, by Odell and Gooding (1986). This is a misconception, however; with Newton–Raphson iteration, the problem can only be universally eliminated by careful choice of a starting formula.

Appendix A

SUBROUTINE ELS2PV

```
      SUBROUTINE ELS2PV (GM, AL, Q, OM, TAU, R, U, VR, VT)
C         ALGORITHM FOR TWO-DIMENSIONAL CONVERSION
C         FROM ORBITAL ELEMENTS TO POSITION AND VELOCITY.
C         INPUT ARGUMENTS ARE: GM (G*M), AL(PHA) (GM/A),
C         Q (PERI DISTANCE), OM(EGA) (ARG-PERI RELATIVE TO
C         ASSUMED REFERENCE DIRECTION) AND TAU (TIME FROM PERI).
C         OUTPUT ARGUMENTS ARE: R (RADIAL DISTANCE),
C         U (ANGLE FROM REFERENCE DIRECTION), VR (RADIAL VELOCIY)
C         AND VT (TRANSVERSE VELOCITY: .GE.0).
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI = 3.1415926535897932384626433832800,
     1                                    FOURPI = 4D0*PI)
      IF (AL.EQ.0D0) THEN
C         (PARABOLA - GM CANNOT BE ZERO)
        D = DCBSOL(0.5D0/GM, Q, 1.5D0*GM*TAU)
        R = Q + 0.5D0*D*D/GM
        H = DSQRT(2D0*GM*Q)
        V = 2D0*DATAN2(D, H)
      ELSE
C         (ELLIPSE OR HYPERBOLA)
        E1 = AL*Q
        E = GM - E1
        EP1 = GM + E
        H = DSQRT(Q*EP1)
        ALP = DABS(AL)
        RTAL = DSQRT(ALP)
C         (LAST 6 ITEMS COULD BE SAVED IF REPEATING GM, AL & Q)
        EM = TAU*ALP*RTAL
        IF (AL.GT.0D0) THEN
C           (ELLIPSE - GM CANNOT BE ZERO)
          EE2 = 0.5D0*EKEPL(EM/GM, E1/GM)
          S2 = DSIN(EE2)
          C2 = DCOS(EE2)
          R = Q + 2D0*E*S2*S2/AL
          D = 2D0*E*S2*C2/RTAL
          V = 2D0*DATAN2(EP1*S2, H*RTAL*C2)
          EMV = EM/GM - V
          V = V + FOURPI*DSIGN(DINT(DABS(EMV/FOURPI) + 0.5D0), EMV)
        ELSE
C           (HYPERBOLA)
          S = SHKEPL(EM/E, -E1/E)
          S2 = S*S
          C = DSQRT(1D0 + S2)
          S2 = S2/(C + 1D0)
          R = Q - E*S2/AL
          D = E*S/RTAL
          V = DATAN2(S*H*RTAL, -GM*S2 - E1)
        END IF
      END IF
C       (ALL ORBITS)
      U = OM + V
      VR = D/R
      VT = H/R
      RETURN
      END
```

Appendix B

SUBROUTINE ELS3PV

```
      SUBROUTINE ELS3PV (GM, AL, Q, EI, BOM, OM, TAU,
     1    X, Y, Z, XDOT, YDOT, ZDOT)
C          ALGORITHM FOR THREE-DIMENSIONAL CONVERSION
C          FROM ORBITAL ELEMENTS TO POSITION AND VELOCITY.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CALL ELS2PV (GM, AL, Q, OM, TAU, R, U, VR, VT)
      C = DCOS(U)
      S = DSIN(U)
      X1 = R*C
      Y1 = R*S
      X2 = VR*C - VT*S
      Y2 = VR*S + VT*C
      C = DCOS(EI)
      S = DSIN(EI)
      Z = Y1*S
      Y1 = Y1*C
      ZDOT = Y2*S
      Y2 = Y2*C
      C = DCOS(BOM)
      S = DSIN(BOM)
      X = X1*C - Y1*S
      Y = X1*S + Y1*C
      XDOT = X2*C - Y2*S
      YDOT = X2*S + Y2*C
      RETURN
      END
```

Appendix C

SUBROUTINE PV2ELS

```
      SUBROUTINE PV2ELS (GM, R, U, VR, VT, AL, Q, OM, TAU)
C          ALGORITHM FOR TWO-DIMENSIONAL CONVERSION
C          FROM POSITION AND VELOCITY TO ORBITAL ELEMENTS.
C          INPUT ARGUMENTS ARE: GM (G*M), R (RADIAL DISTANCE),
C          U (ANGLE FROM ASSUMED REFERENCE DIRECTION),
C          VR (RADIAL VELOCITY) AND VT (TRANSVERSE VELOCITY: .GE.0).
C          OUTPUT ARGUMENTS ARE: AL(PHA) (GM/A), Q (PERI DISTANCE),
C          OM(EGA) (ARG-PERI RELATIVE TO REFERENCE DIRECTION)
C          AND TAU (TIME FROM PERI).
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL L
      PARAMETER (PI = 3.141592653589793238462643383280D0,
     1           TWOPI = 2D0*PI, SW = 0.25D0, L = .FALSE.)
C      (ALL ORBITS)
      VSQ = VR*VR + VT*VT
      AL = 2D0*GM/R - VSQ
      ALP = DABS(AL)
      RTAL = DSQRT(ALP)
      D = R*VR
      H = R*VT
      P = H*H
      ESQ1 = P*AL
```

```
        ES = D*RTAL
        ESES = ES*ES
        EC = R*VSQ - GM
        ECEC = EC*EC
        IF (AL.GT.0D0) THEN
C           (ONE ESQ FORMULA SUPERIOR FOR THE ELLIPSE)
          ESQ = ECEC + ESES
         ELSE
C           (DIFFERENT FORMULA SUPERIOR FOR THE HYPERBOLA)
          ESQ = GM*GM - ESQ1
        END IF
        E = DSQRT(ESQ)
        Q = P/(GM + E)
        IF (AL.EQ.0D0) THEN
C           (PARABOLA)
          TAU = D*(2D0*Q + R)/(3D0*GM)
          V = 2D0*DATAN2(VR, VT)
         ELSE IF (E.EQ.0D0) THEN
C           (CIRCLE)
          TAU = 0D0
          V = 0D0
         ELSE
C           (ELLIPSE OR HYPERBOLA)
          E1 = AL*Q
          IF (AL.GT.0D0) THEN
C             (ELLIPSE)
            EH = DATAN2(ES, EC)
            IF (GM*EH*EH/6D0 + E1 .GE. GM*SW) THEN
C               (GENERAL CASE)
              EM = GM*EH - ES
              ECESQ = GM*EC - ESQ
             ELSE
C               (FOR E1 & EH BOTH NEAR ZERO)
              EM = GM*EMKEP(E1/GM, EH)
              ECESQ = (ESQ1*ECEC - ESQ*ESES)/(ESQ + GM*EC)
            END IF
           ELSE
C             (HYPERBOLA)
            EH = DASINH(ES/E)
            IF (GM*EH*EH/6D0 - E1 .GE. GM*SW) THEN
C               (GENERAL CASE)
              EM = ES - GM*EH
              ECESQ = ESQ - GM*EC
             ELSE
C               (FOR E1 & EH BOTH NEAR ZERO)
              EM = E*SHMKEP(-E1/E, ES/E)
              ECESQ = -(ESQ1*ECEC + ESQ*ESES)/(ESQ + GM*EC)
            END IF
          END IF
C           (ELLIPSE OR HYPERBOLA STILL)
          EN = ALP*RTAL
          TAU = EM/EN
          V = DATAN2(ES*H*RTAL, ECESQ)
        END IF
C         (ALL ORBITS)
        OM = U - V
        IF (L .AND. AL.GT.0D0) THEN
C           (FOR ELLIPSE, ADJUST REVOLUTIONS IF REQUIRED (USING L))
          ADJ = TWOPI*DSIGN(DINT(DABS(OM/TWOPI) + 0.5D0), OM)
          OM = OM - ADJ
```

```
            TAU = TAU + ADJ/EN
         END IF
         RETURN
         END
```

Appendix D

SUBROUTINE PV3ELS

```
         SUBROUTINE PV3ELS (GM, X, Y, Z, XDOT, YDOT, ZDOT,
      1     AL, Q, EI, BOM, OM, TAU)
C            ALGORITHM FOR THREE-DIMENSIONAL CONVERSION
C            FROM POSITION AND VELOCITY TO ORBITAL ELEMENTS.
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         PARAMETER (PI=3.14159265358979323846264338328D0, HALFPI=PI/2D0)
         XSQYSQ = X*X + Y*Y
         RSQ = XSQYSQ + Z*Z
         R = DSQRT(RSQ)
         VR = (X*XDOT + Y*YDOT + Z*ZDOT)/R
         HX = Y*ZDOT - Z*YDOT
         HY = Z*XDOT - X*ZDOT
         HZ = X*YDOT - Y*XDOT
         HSQ = HX*HX + HY*HY + HZ*HZ
         IF (HSQ.EQ.0D0) THEN
C            (RECTILINEAR ORBIT)
           EI = HALFPI
           IF (XSQYSQ.EQ.0D0) THEN
C              (AXIAL ORBIT)
             BOM = 0D0
            ELSE
C              (GENERAL RECTILINEAR ORBIT)
             BOM = DATAN2(Y, X)
           END IF
           U = DATAN2(Z, DSQRT(XSQYSQ))
           VT = 0D0
          ELSE
C            (NON-DEGENERATE ORBIT)
           BX = HY*Z - HZ*Y
           BY = HZ*X - HX*Z
           BZ = HX*Y - HY*X
           HX = Y*BZ - Z*BY
           HY = Z*BX - X*BZ
           HZ = X*BY - Y*BX
           W = HX*HX + HY*HY
           H = DSQRT(W + HZ*HZ)
           EI = DATAN2(DSQRT(W), HZ)
           IF (W.EQ.0D0) THEN
C              (ORBIT IN REFERENCE PLANE)
             BOM = 0D0
             U = DATAN2(Y*DSIGN(1D0,HZ), X)
            ELSE
C              (GENERAL ORBIT)
             BOM = DATAN2(HX, -HY)
             U = DATAN2(H*Z, RSQ*BZ)
           END IF
           VT = H/(R*RSQ)
         END IF
         CALL PV2ELS (GM, R, U, VR, VT, AL, Q, OM, TAU)
         RETURN
         END
```

References

Broucke, R. A. and Cefola, P. J.: 1972, 'On the Equinoctial Orbit Elements', *Celest. Mech.* **5**, 303–310.

Cohen, C. J. and Hubbard, E. C.: 1962, 'A Nonsingular Set of Orbit Elements', *Astron. J.* **67**, 10–15.

Gooding, R. H.: 1983, 'Complete Second-Order Satellite Perturbations due to J_2 and J_3, Compactly Expressed in Spherical-Polar Coordinates', *Acta Astr.* **10**, 309–317.

Gooding, R. H.: 1987, 'Universal Procedures for Conversion of Orbital Elements to and from Position and Velocity (unperturbed orbits)', RAE Technical Report 87043.

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

Gooding, R. H. and Odell, A. W.: 1988, 'The Hyperbolic Kepler Equation (And the Elliptic Equation Revisited)', *Celest. Mech.*, this issue, pp. 267–282.

Gooding, R. H. and Tayler, R. J.: 1968, 'A PROP3 Users' Manual', RAE Technical Report 68299.

Goodyear, W. H.: 1965, 'Completely General Closed-Form Solution for Coordinates and Partial Derivatives of the Two-Body Problem', *Astron. J.* **70**, 189–192.

Herrick, S.: 1965, 'Universal Variables', *Astron. J.* **70**, 309–315.

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

Pitkin, E. T.: 1965, 'A Regularized Aproach to Universal Variables', *AIAA J.* **3**, 1508–1511.

Shepperd, S. W.: 1985, 'Universal Keplerian State Transition Matrix', *Celest. Mech.* **35**, 129–144.

Stumpff, K.: 1947, 'Neue Formeln und Hilfstafeln zur Ephemeriden Rechnung', *Astron. Nachrichten* **275**, 108–128.

Sundman, K. F.: 1912, 'Mémoire sur le Problème des Trois Corps', *Acta Math.* **36**, 105–179.