

An efficient code to solve the Kepler equation

Hyperbolic case[★]

V. Raposo-Pulido^{**} and J. Peláez

Space Dynamics Group-Technical University of Madrid, 28040 Madrid, Spain
e-mail: v.raposo.pulido@upm.es, j.pelaez@upm.es

Received 4 June 2018 / Accepted 6 September 2018

ABSTRACT

Context. This paper introduces a new approach for solving the Kepler equation for hyperbolic orbits. We provide here the Hyperbolic Kepler Equation–Space Dynamics Group (HKE–SDG), a code to solve the equation.

Methods. Instead of looking for new algorithms, in this paper we have tried to substantially improve well-known classic schemes based on the excellent properties of the Newton–Raphson iterative methods. The key point is the seed from which the iteration of the Newton–Raphson methods begin. If this initial seed is close to the solution sought, the Newton–Raphson methods exhibit an excellent behavior. For each one of the resulting intervals of the discretized domain of the hyperbolic anomaly a fifth degree interpolating polynomial is introduced, with the exception of the last one where an asymptotic expansion is defined. This way the accuracy of initial seed is optimized. The polynomials have six coefficients which are obtained by imposing six conditions at both ends of the corresponding interval: the polynomial and the real function to be approximated have equal values at each of the two ends of the interval and identical relations are imposed for the two first derivatives. A different approach is used in the singular corner of the Kepler equation – $|M| < 0.15$ and $1 < e < 1.25$ – where an asymptotic expansion is developed.

Results. In all simulations carried out to check the algorithm, the seed generated leads to reach machine error accuracy with a maximum of three iterations (~99.8% of cases with one or two iterations) when using different Newton–Raphson methods in double and quadruple precision. The final algorithm is very reliable and slightly faster in double precision (~0.3 s). The numerical results confirm the use of only one asymptotic expansion in the whole domain of the singular corner as well as the reliability and stability of the HKE–SDG. In double and quadruple precision it provides the most precise solution compared with other methods.

Key words. methods: numerical – celestial mechanics – space vehicles

1. Introduction

The hyperbolic Kepler equation (HKE) is

$$M = e \sinh H - H \quad (1)$$

and it must be solved to obtain the value of the hyperbolic anomaly H in terms of the mean anomaly M and the eccentricity e of the orbit: $H = H(M, e)$ or $y = y(x, e)$. In this paper we use the following notation: the hyperbolic anomaly is denoted by H or y without distinction. Equally, the mean anomaly is denoted M or x .

Solving Kepler’s equation – in its elliptic or hyperbolic version – is not a new problem. For approximately the last three centuries many scientists have been interested in this resolution and many papers related to this topic have been published in each decade from the mid-seventeenth century. By 1960 problems of celestial mechanics and orbit determination were being reformulated, and indeed new fields, including astronautics and astrodynamics, were being built to study them (Colwell 1993). The introduction of rocket thrusters for space flight presented

possibilities of encountering trajectories with osculating orbits whose character changes significantly during the period of interest. In that sense, a low-acceleration escape trajectory might start in a nearly circular orbit and finish on a hyperbolic path. This aspect motivated researchers to employ a formulation free from the possibilities of indeterminateness, no matter what form the orbit takes, elliptic, parabolic, hyperbolic, or possibly a circular or linear limit (Pitkin 1965).

As a consequence, during that time there were efforts in several directions to reformulate laws of orbital motion, and in particular the Kepler equation, into universal variables which incorporated all eccentricities and allow rapid, accurate computation of positions and velocities. Some of these approaches can be found in Battin (1964), Goodyear (1965), Stumpff (1968), Herrick (1971), Gooding (1987), Sarnecki (1988), Sharaf & Sharaf (2003), Beust et al. (2016), or Wisdom & Hernandez (2015). Even analytical solutions have been found using universal variables (see Tokis 2014).

However, the interest for the hyperbolic version of the Kepler equation is based on many different reasons. First of all, the solution of any highly transcendental equation is interesting, in itself, from the point of view of applied mathematics. In the field of numerical analysis some powerful techniques have been devoted to solve transcendental equations; for example, the homotopy continuation methods (see Sharaf et al. 2007; Alhindi & Sharaf 2014).

[★] A copy of the C code model is available at the CDS via anonymous ftp to cdsarc.u-strasbg.fr (130.79.128.5) or via <http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/619/A129>

^{**} Present address: E.T.S.I. Aeronáutica y del Espacio, Pza. Cardenal Cisneros 3, 28040 Madrid, Spain.

Obviously the HKE is most important in the fields of astronomy, celestial mechanics, and astrodynamics. In their paper [Wisdom & Holman \(1991\)](#) strongly underlined the important role that long-term integrations play in investigations in dynamical astronomy. A clear example is the LONGSTOP project [Milani et al. \(1986\)](#), an international cooperative program with the purpose of investigating the stability of our solar system. In these kinds of integrations, it is of paramount interest to take care of the errors involved (see [Milani & Nobili 1987](#)). The imperfect convergence of an iterative algorithm – especially when we solve transcendental equations such as the Kepler equation – is a well known source of error. To minimize such errors is always an interesting issue that helps to achieve the main objective of the research. The Kepler equation must be solved in symplectic integration codes as for example in the *Wisdom–Holman method* ([Wisdom & Holman 1991](#)), where the orbital elements experience small changes at every step. It can be used for long-term gravitational simulations ([Rein & Tamayo 2015](#)), or for the collisional gravitational N-body problem ([Hernandez & Bertschinger 2015](#); [Dehnen & Hernandez 2016](#)). Beyond this, the correct determination of position and velocity in perturbed cometary orbits, which depends on the fast and accurate solution of the HKE, became an important issue. The introduction of computing machines gave impetus to the generation of algorithms that help in tasks of this kind from the very beginning. One example can be found in [Davidson \(1932\)](#), [Smiley \(1934\)](#), [Davidson \(1934\)](#) where a method is introduced in order to calculate the true anomaly and the time of perihelion passage for hyperbolic orbits. The authors gave an approximate solution of the HKE without explicitly mentioning it. Another example of algorithms designed to be used in a computing machine can be found in [Sitarski \(1968\)](#).

Some comets are originated outside of the solar system. Such comets enter the solar system coming from the interstellar space and may exit the solar system through hyperbolic trajectories. Recently the object 1I/2017 U1 reached its perihelion (0.2483 ua) following one of these hyperbolic trajectories. Frequently, the HKE is used to describe the hyperbolic anomaly of the comet. The determination of the radial distance – or the Cartesian coordinates – of such a comet requires somewhat accurate knowledge of the hyperbolic anomaly. [Faintich \(1972\)](#) shows how a near-parabolic comet in the Oort cloud can be perturbed by a passing star. The description of the perturbed comet orbit requires us to consider the hyperbolic motion of both stars around its common centre of mass and the time evolution of the whole system leads to a solution of the HKE.

There are, however, other fields in which the HKE plays a significant role. For example, [Rauh & Parisi \(2011\)](#) faces the problem of deriving macroscopic properties from the Hamiltonian of the hydrogen atom considering possible hyperbolic orbits. By using the Kustaanheimo–Stiefel (KS) transformation a pseudo-time σ is introduced into the problem. However, the relation between σ and the real time t is given by a particular version of the HKE. Thus in some quantum mechanics problems the HKE appears naturally.

Stellar dynamics is another field where the HKE also appears (see, for example; [Kurth 1978](#)). In [Yabushita \(1971\)](#) plan investigation is made of the possibility of a capture process in which a planet of negligible mass, initially in Keplerian orbit about a star, is perturbed by a passing star with hyperbolic velocity at great distance and becomes a planet of the star escaping to infinity with it. In order to integrate the governing equations, the HKE needs to be solved for each integration step. Since they did not

have a fast and reliable solver the authors opt to integrate an additional differential equation.

Thus in many integration processes the solution of the Kepler equation – elliptic or hyperbolic – could jeopardize the procedure as indicated in [Balaraman & Vranceanu \(2007\)](#). It follows, from here, the need for an efficient, fast and accurate calculation method to solve the HKE.

The HKE has been studied by several authors to develop optimal methods. In this regard, the main aspects to take into account are the accuracy and the computational performance, which are related to the method for determining the starting estimate H . The approaches in the paragraphs below represent some of the current methods used to solve (2).

The Gooding 1988 approach described in [Gooding & Odell \(1988\)](#): The starter is based on Lagrange’s theorem, where the equation is rewritten as a function of $S = \sinh H$ instead of H . First, Halley’s method is applied to obtain the corrector of S as well as of the function and its derivative. With these values, the Newton-Raphson method is applied to determine the solution of the equation. It is an iterative method which requires several transcendental function evaluations.

The Fukushima 1997 approach described in [Fukushima \(1997\)](#): Several intervals of $L = M/e$ are defined to select where the solution should be determined. The different methods and approximations chosen for each interval depend on the expected size solution. First, four cases in which the solution is large are selected. For these cases an asymptotic form of the main equation is considered and an approximate solution is found. Next, four other for which the solution is small cases are selected. In these latter case an iterative procedure is applied to solve the approximate forms of Kepler’s equation. In the remaining case, for which the solution interval becomes finite, a discretized Newton method is applied as well as a Newton method in which the functions are evaluated by Taylor series expansions. In that case, the starter value is the minimum of some upper bound of the solution prepared by using the Newton correction formula. This approach, depending on the case, uses or does not use an iterative method that requires several transcendental function evaluations.

The Avendaño 2015 approach described in [Avendaño et al. \(2015\)](#): The equation is rewritten as a function of $S = \sinh H$ instead of H , such that the starter S is a piecewise-defined function involving several linear expressions and one with cubic and square roots. Once the seed is estimated, a Newton’s method is applied, by using Smale’s α -theory to decide whether the starter gives the claimed convergence rate. It is an iterative method which requires several transcendental function evaluations.

The goal of the present study is the determination of a suitable seed to initialize the numerical method for solving the Kepler equation. For that we consider the optimization already tested of the Newton–Raphson method and extend the method described in [Raposo-Pulido & Peláez \(2017\)](#) for the elliptical Kepler equation to the HKE. Following previous work, the Maple symbolic manipulator is chosen to obtain the code in the C programming language. The code developed is called HKE–SDG. As a result we obtain a simple way to provide fast and accurate solutions.

2. Hyperbolic Kepler equation

The Kepler equation for the hyperbolic case

$$x = e \sinh y - y, \quad x \in] - \infty, \infty[, \quad y \in] - \infty, \infty[, \quad e > 1 \quad (2)$$

determines a nonlinear function $y = y(x, e)$ where y is the hyperbolic anomaly related with the true anomaly θ by means

of

$$\cos \theta = \frac{e - \cosh y}{e \cosh y - 1}, \quad \cosh y = \frac{e + \cos \theta}{e \cos \theta + 1}, \quad (3)$$

$$\sin \theta = \frac{\sqrt{e^2 - 1} \sinh y}{e \cosh y - 1}, \quad \sinh y = \frac{\sqrt{e^2 - 1} \sin \theta}{e \cos \theta + 1}, \quad (4)$$

and x is the equivalent to the mean anomaly in the elliptic motion:

$$x = \sqrt{\frac{\mu}{a^3}}(t - \tau). \quad (5)$$

The eccentricity is larger than one. As a consequence x and y have the same sign, that is, $y > 0 \Leftrightarrow x > 0$. Since the right hand side of (2) is an odd function, for negative values of x the following change of variables leads to

$$x = -x, \quad y = -y \quad \Rightarrow \quad x = e \sinh y - y \quad (6)$$

where x and y are positives. Therefore, we focus the analysis in the resolution of Eq. (2) for positive values: $(x, y) \in [0, \infty[\times]0, \infty[$.

The solution of the Kepler equation can be seen as a root finder problem of the function

$$f(y) \equiv e \sinh y - y - x = 0 \quad (7)$$

for given values of e and x . In such a case, the Kepler equation univocally determines a bijective function $y = y(x, e)$ which is defined on \mathbb{R}_+ . This property can be deduced from Bolzano's theorem and the fact that function $f(y)$ in Eq. (7) is strictly increasing. Indeed, $f(y)$ is a continuous function on the closed interval $[y_1, y_2] = [0, \frac{x}{e-1}]$ and it takes values of the opposite sign at the extremes

$$f(0) = -x < 0 \quad (8)$$

$$f\left(\frac{x}{e-1}\right) = e \sinh\left(\frac{x}{e-1}\right) - \frac{x}{e-1} - x = e \left(\frac{1}{3!} \left(\frac{x}{e-1}\right)^3 + \frac{1}{5!} \left(\frac{x}{e-1}\right)^5 + \dots \right) > 0. \quad (9)$$

Applying Bolzano's theorem to function $f(y)$, there exist an intermediate value $y^* \in [y_1, y_2]$ which verifies $f(y^*) = 0$. Therefore $f(y)$ vanishes at least once in the stated interval. However, the derivative of $f(y)$, $f'(y)$, is always positive

$$\frac{df}{dy} = e \cosh y - 1 \geq e - 1 > 0. \quad (10)$$

That is, $\forall y_2 > y_1$ with $y_1, y_2 \in [0, \infty[$, $f(y_2) \geq f(y_1)$ and $f(y)$ is strictly increasing. Consequently, $f(y)$ is zero for only one value of $y \in [y_1, y_2]$. In summary, for given x , $\exists ! y$ such that $f(y) = 0$.

3. Newton–Raphson algorithms

Following the procedure developed and validated in Raposo-Pulido & Peláez (2017), we use the Newton–Raphson algorithms to solve the HKE. The root searched is solution of Eq. (7). To find this root we wish to establish a successive approximation method starting from a seed y_0 . If y_n is one of the terms of the sequence, the next term will be $y_{n+1} = y_n + \Delta y_n$, where Δy_n is given by applying a root-finding method of

Laguerre (1834–1886) to the solution of the HKE (Conway 1986)

$$\Delta y_n = \frac{-mf(y_n)}{f'(y_n) \pm \sqrt{|(m-1)[(m-1)f'^2(y_n) - mf(y_n)f''(y_n)]}} \quad (11)$$

in which the sign (+) is selected when $f'(y_n)$ is positive; if $f'(y_n) < 0$ the sign (–) must be chosen. The introduction of absolute value in the Eq. (11) is useful to avoid failures when the square root is a complex number. It does not affect the convergence of the algorithm which is practically independent of m , the degree of the polynomial.

Three different algorithms are be considered in this study: the classical Newton–Raphson (CNR) algorithm ($m = 1$); the modified Newton–Raphson (MNR) algorithm ($m = 2$); and the Conway algorithm ($m = 5$).

All of three algorithms use an initial seed to initialize the convergent sequence. However, its behavior, related with the reliability, is going to mark the difference. While the MNR method always converges to the real solution independently of the quality of the seed, the CNR is less stable. Therefore CNR method is more correlated with the initial value than the Conway or the MNR methods. A way to escape from this disadvantage and assure the reliability of the method is by using a very good seed as initial point. In that case the CNR and the MNR algorithms converge quickly to the real solution, as is shown in Raposo-Pulido & Peláez (2017) for the elliptic Kepler equation.

4. Seed value

As we briefly explain in the introduction, the core of this work is to generate a very good initial seed in all possible cases. If this is satisfied, then we can benefit from the convergent property of the Newton–Raphson algorithms (CNR, MNR, or Conway) to determine the right solution of the Kepler equation. This approach has been used with success in the elliptic case (see Raposo-Pulido & Peláez 2017). In this paper, we show that in the hyperbolic case too, improving the initial seed to feed a Newton–Raphson algorithm is much more cost-effective than searching for new algorithms better than the classic Newton–Raphson algorithms.

We must take into account that the accuracy of the Newton–Raphson algorithms is jeopardized when $x = M \ll 1$ and $e-1 \ll 1$; This region of the plane (e, M) is called the singular corner. In such a region y is also small, and the expansion of the equation for small values of y provides

$$y \approx \frac{x}{e-1} \quad (12)$$

that is, the solution is badly determined from a numerical point of view since it is the ratio of two infinitesimals.

In order to determine the optimal seed, we need to introduce two different scenarios, depending on whether we are inside or outside of the singular corner. From a practical point of view we define the singular corner as the region in which $x = M < 0.15$ and $e < 1.25$. These values have been chosen arbitrarily: firstly inside the singular corner ($x = M < 0.15$ and $e < 1.25$): three regions (inner, intermediate and outer) are defined in terms of the size of x and y . In these regions we estimate the starter through an asymptotic expansion. And secondly outside the singular corner ($x = M \geq 0.15$ or $e \geq 1.25$): two regions are defined in terms of the infinite y -domain $[0, \infty[$ to estimate the starter. In the finite

interval $[0, 5[$ the seed is determined through a fifth degree polynomial, while in the infinite interval $[5, \infty[$ the seed is estimated through an asymptotic approximation.

As we see below, the polynomial approximation works very well in most cases, especially when $x = M$ is close to the ends of the interval where the polynomial is defined. In such a case the seed practically leads to the solution of the Kepler equation. However, the singular behavior of the Kepler equation makes the approximation of $y(x)$ through this procedure ineffective and a special analysis should be implemented.

5. Inside the singular corner

To describe the solution inside the singular corner we introduce the value $\varepsilon = e - 1$ assuming that $\varepsilon \ll 1$:

$$x = (1 + \varepsilon) \sinh y - y = (\sinh y - y) + \varepsilon \sinh y. \quad (13)$$

The objective is to obtain an approximation to the exact solution $y_v(x)$ of Eq. (13) with enough accuracy to be the *seed* used to start the Newton–Raphson convergent process. To this end, an asymptotic expansion in power of the small parameter ε will be introduced.

However we must distinguish between several regimes inside the singular corner:

- the inner region, where $x \approx O(\varepsilon^2)$ and $y \approx O(\varepsilon)$. In this inner region the most important term on the right hand side of Eq. (13) is $\varepsilon \sinh y$.
- the intermediate region, where $x \approx O(\varepsilon^{\frac{3}{2}})$ and $y \approx O(\varepsilon^{\frac{1}{2}})$. In this intermediate region both terms on the right hand side of Eq. (13) are of the same order: $\sinh y - y \approx \varepsilon \sinh y$.
- the outer region, where $x \approx O(\varepsilon^{\frac{3}{4}})$ and $y \approx O(\varepsilon^{\frac{1}{4}})$. In this outer region the most important term on the right hand side of Eq. (13) is $\sinh y - y$.

In each of these regions we obtain an asymptotic expansion of the exact solution $y_v(x)$ of equation Eq. (13) in terms of some power of the small parameter ε ; the change of variable Eq. (43) provides a very good seed S_0 to feed the Newton–Raphson convergent process.

5.1. Inner region

In the inner region we use new variables given by the following relations:

$$x = \varepsilon^2 \xi, \quad y = \varepsilon \eta, \quad \text{with} \quad \xi, \eta \approx O(1), \quad (14)$$

and where (ξ, η) are both positive amounts. In Eq. (13) we introduce the expansion

$$\xi = \eta + a_1 \varepsilon + a_2 \varepsilon^2 + a_3 \varepsilon^3 + \dots \quad (15)$$

The coefficients $a_i, i = 1, 2, \dots$ can be obtained expanding appropriately the right hand side of Eq. (13). The solution is

$$\xi = \eta + \frac{1}{6} \eta^3 \varepsilon + \frac{1}{6} \eta^3 \varepsilon^2 + \frac{1}{120} \eta^5 \varepsilon^3 + \frac{1}{120} \eta^5 \varepsilon^4 + \dots \quad (16)$$

We must invert this solution to obtain η as a function of ξ ; to this end we introduce the expansion

$$\eta = \xi + b_1 \varepsilon + b_2 \varepsilon^2 + b_3 \varepsilon^3 + \dots \quad (17)$$

To calculate the coefficients $b_i, i = 1, 2, \dots$, which are functions of ξ , we introduce this expansion in (16) and require that the different coefficients of each order match. The result is

$$\eta = \xi - \frac{1}{6} \xi^3 \varepsilon + \frac{1}{12} \xi^3 (\xi^2 - 2) \varepsilon^2 - \frac{1}{360} \xi^5 (20 \xi^2 - 57) \varepsilon^3 + \dots \quad (18)$$

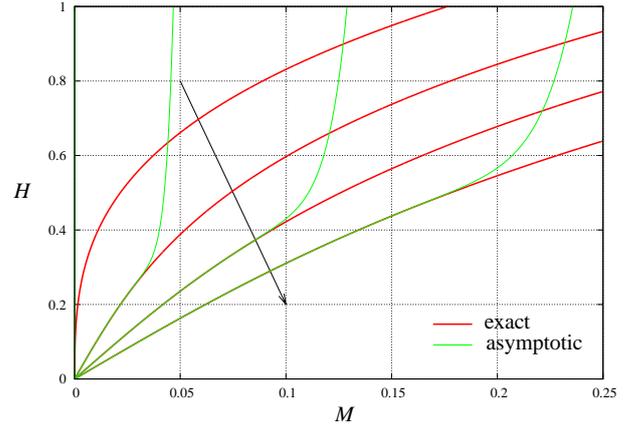


Fig. 1. Exact solution of the Kepler equation and the asymptotic solution – up to sixth order – obtained in the inner region of the singular corner. The agreement is excellent for very small values of x . Here four values of ε have been considered: $\varepsilon = 0.001, 0.101, 0.201, 0.301$.

Figure 1 shows the behavior of the asymptotic solution (Eqs. (14)–(18)) in the inner region. The agreement with the exact solution is excellent for very small values of x .

5.2. Outer region

In the outer region we introduce the following change of variables:

$$x = \varepsilon^{\frac{3}{4}} \nu, \quad y = \varepsilon^{\frac{1}{4}} \mu, \quad \text{with} \quad \nu, \mu \approx O(1). \quad (19)$$

We note that (ν, μ) are both positive amounts. We look for a solution of Eq. (13) given by

$$\nu = a_0 + a_1 \varepsilon^{\frac{1}{2}} + a_2 \varepsilon + a_3 \varepsilon^{\frac{3}{2}} + \dots \quad (20)$$

where the coefficients $a_i, i = 1, 2, \dots$ can be obtained by appropriately expanding the right hand side of Eq. (13). The solution is

$$\nu = \frac{1}{6} \mu^3 + \frac{1}{120} \mu (\mu^4 + 120) \varepsilon^{\frac{1}{2}} + \frac{1}{5040} \mu^3 (\mu^4 + 840) \varepsilon + \dots \quad (21)$$

Let $\mu_0 > 0$ be the real and positive solution of the cubic equation

$$\frac{1}{6} \mu^3 - \nu = 0. \quad (22)$$

This root always exists and is univocally defined because ν is always positive. It takes the value

$$\mu_0 = (6\nu)^{\frac{1}{3}}. \quad (23)$$

Once the value of μ_0 is known it is possible to invert the solution (21) to obtain μ as a function of ν ; to do that we introduce the expansion

$$\mu = \mu_0 + b_1 \varepsilon^{\frac{1}{2}} + b_2 \varepsilon + b_3 \varepsilon^{\frac{3}{2}} + \dots \quad (24)$$

where the coefficients $b_i, i = 1, 2, \dots$ are functions of μ_0 that must be calculated. Introducing this expansion into Eq. (21) and requiring that the different coefficients of each order match it is possible to obtain the values of $b_i, i = 1, 2, \dots$. The result is

$$\mu = \mu_0 - \frac{\mu_0^4 + 120}{60 \mu_0} \varepsilon^{\frac{1}{2}} + \frac{\mu_0 (\mu_0^4 - 280)}{1400} \varepsilon - \frac{\mu_0^{12} - 162 \mu_0^8 - 25200 \mu_0^4 - 67200}{25200 \mu_0^5} \varepsilon^{\frac{3}{2}} + \dots \quad (25)$$

5.3. Intermediate region

In the intermediate region we introduce the following change of variables:

$$x = \varepsilon^{\frac{3}{2}} \chi, \quad y = \varepsilon^{\frac{1}{2}} \sigma, \quad \text{with} \quad \chi, \sigma \approx \mathcal{O}(1). \quad (26)$$

We note that (χ, σ) are both positive amounts. We look for a solution of Eq. (13) given by

$$\chi = a_0 + a_1 \varepsilon + a_2 \varepsilon^2 + a_3 \varepsilon^3 + \dots \quad (27)$$

where the coefficients $a_i, i = 1, 2, \dots$ can be obtained expanding appropriately the right hand side of Eq. (13). The solution is

$$\chi = \frac{1}{6} \sigma^3 + \sigma + \frac{1}{120} \sigma^3 (\sigma^2 + 20) \varepsilon + \frac{1}{5040} \sigma^5 (\sigma^2 + 42) \varepsilon^2 + \dots \quad (28)$$

The next step is to invert this solution to obtain σ as a function of χ . Let $\sigma_0 > 0$ be the real and positive solution of the cubic equation

$$f(\sigma) = \frac{1}{6} \sigma^3 + \sigma - \chi = 0. \quad (29)$$

This root always exists; in effect, the function $f(\sigma)$ verifies

$$f(0) = -\chi < 0 \quad (30)$$

$$f(\chi) = \frac{\chi^3}{6} > 0 \quad (31)$$

therefore in the interval $[0, \chi]$ there is a root of $f(\sigma) = 0$ at least. But there is only one root, because the derivative

$$f'(\sigma) = \frac{\sigma^2}{2} + 1 > 0 \quad (32)$$

is always positive in that interval. That root is given by

$$\sigma_0 = T - \frac{2}{T} \quad (33)$$

where $T = (\Lambda + 3\chi)^{\frac{1}{3}}$ and $\Lambda = \sqrt{8 + 9\chi^2}$. It should be noticed that expression (33) can present a rounding-error defect when $T \approx \sqrt{2}$ ($\sigma_0 \ll 1$). In order to minimize this error, we develop an alternative procedure to provide σ_0 . We rewrite the cubic Eq. (29) as

$$\sigma \left(\frac{\sigma^2}{6} + 1 \right) - \chi = 0 \Leftrightarrow \sigma = \frac{6\chi}{\sigma^2 + 6} \quad (34)$$

such that σ is replaced by (33) on the right hand side of the equation, obtaining

$$\sigma_0 = \frac{6\chi}{2 + T^2 + \frac{4}{T^2}} \quad (35)$$

which provides a smaller rounding-error when $\chi \ll 1$, that is, when $\Lambda \approx 2\sqrt{2}$.

Once the value of σ_0 is known it is possible to invert the solution (28) to obtain σ as a function of χ ; to do that we introduce the expansion

$$\sigma = \sigma_0 + b_1 \varepsilon + b_2 \varepsilon^2 + b_3 \varepsilon^3 + \dots \quad (36)$$

where the coefficients $b_i, i = 1, 2, \dots$ are functions of σ_0 that must be calculated. Introducing this expansion in (28) and requiring that the different coefficients of each order match it is possible to obtain the values of $b_i, i = 1, 2, \dots$. The result is

$$\sigma = \sigma_0 - \frac{\sigma_0^3 (\sigma_0^2 + 20)}{60(\sigma_0^2 + 2)} \varepsilon + \frac{\sigma_0^5 (\sigma_0^6 + 25\sigma_0^4 + 340\sigma_0^2 + 840)}{1400(\sigma_0^2 + 2)^3} \varepsilon^2 + \dots \quad (37)$$

5.3.1. Matching with the inner region

A detailed insight of the solution in this region shows that when (χ, σ) take small values the asymptotic solution tends to the inner solution obtained previously. In effect, when

$$\chi \approx \varepsilon^{1/2} \xi, \quad \sigma \approx \varepsilon^{1/2} \eta \quad (38)$$

the solution enters in the domain of the inner solution. For that values the parameter σ_0 given by (35) takes the value

$$\sigma_0 \approx \varepsilon^{1/2} \left(\xi - \frac{1}{6} \xi^3 \varepsilon + \frac{1}{12} \xi^5 \varepsilon^2 - \frac{1}{18} \xi^7 \varepsilon^3 + \dots \right). \quad (39)$$

If this value is introduced into Eq. (37) provides

$$\sigma = \varepsilon^{1/2} \eta = \varepsilon^{1/2} \left(\xi - \frac{1}{6} \xi^3 \varepsilon + \frac{1}{12} \xi^3 (\xi^2 - 2) \varepsilon^2 + \dots \right) \quad (40)$$

and this expression coincides with Eq. (18) corresponding to the inner solution. This is the reason why the asymptotic solution given by Eq. (37) also provides the solution in the inner region where x is in the domain of the inner solution.

5.3.2. Matching with the outer region

When (χ, σ) take large values the asymptotic solution tends to the outer solution obtained previously. In effect, when

$$\chi \approx \frac{\nu}{\varepsilon^{3/4}}, \quad \sigma \approx \frac{\mu}{\varepsilon^{1/4}} \quad (41)$$

the solution enters in the domain of the outer solution. For those values the parameter σ given by Eq. (37) takes the value

$$\sigma = \frac{\mu}{\varepsilon^{1/4}} = \frac{1}{\varepsilon^{1/4}} \left\{ \mu_0 - \frac{\mu_0^4 + 120}{60\mu_0} \varepsilon^{\frac{1}{2}} + \frac{\mu_0 (\mu_0^4 - 280)}{1400} \varepsilon - \frac{\mu_0^{12} - 162\mu_0^8 - 25200\mu_0^4 - 67200}{25200\mu_0^5} \varepsilon^{\frac{3}{2}} + \dots \right\} \quad (42)$$

and this expression coincides with Eq. (25) corresponding to the outer solution. This is the reason why the asymptotic solution given by Eq. (37) also provides the solution in the outer region where x is in the domain of the outer solution.

5.4. Summary of the singular corner

The analysis carried out above shows that the asymptotic expansion Eq. (37) corresponding to the intermediate region also describes the solution in the inner and outer regions of the singular corner. Figure 2 shows the exact solution of the HKE and the asymptotic solution – up to the fourth order – obtained in the intermediate region of the singular corner. The agreement is exceptional in the whole range of values of $x = M$. These numerical results confirm the theoretical analysis. The main consequence is that, from a practical point of view, the asymptotic expansion Eq. (37) can be used in the whole domain of the singular corner.

Therefore, only one asymptotic expansion is considered in HKE–SDG: that corresponding to the intermediate region. This approximate solution permits us to generate a very good seed to feed the Newton–Raphson convergent process for values of (e, M) inside the singular corner.

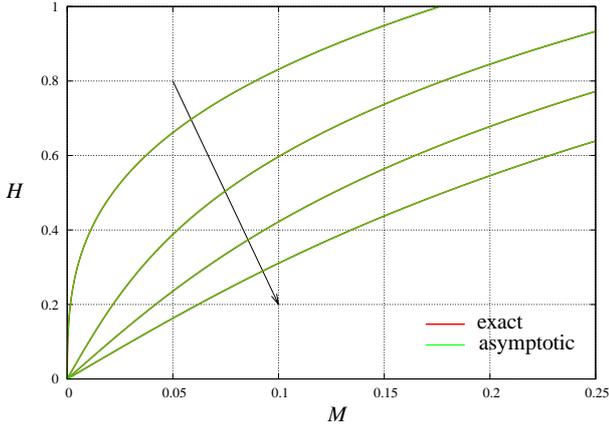


Fig. 2. Exact solution of the Kepler equation and the asymptotic solution – up to the fourth order – obtained in the intermediate region of the singular corner. The agreement is excellent for whole range. Here four values of ε have been considered: $\varepsilon = 0.001, 0.101, 0.201, 0.301$.

6. Outside the singular corner

We begin by performing the following change of variable

$$S = \sinh y \quad \Leftrightarrow \quad y = \ln|S + \sqrt{1 + S^2}| \quad (43)$$

which transforms the Kepler equation into the relation

$$x = eS - \ln|S + \sqrt{1 + S^2}| \quad (44)$$

with the variable S defined in the interval $[0, \infty[$. Due to the infinite domains of both variables (x, y) , the range of y is divided in two intervals: $[0, 5[$ and $[5, \infty[$ such that for each interval a different treatment will be applied. The limit value of five has been chosen arbitrarily; we note that the corresponding value of $S \approx 74.2$ is large compared with unity. In the interval $[0, 5[$ the seed will be generated with polynomials. In the interval $[5, \infty[$ the seed will be generated with the help of an asymptotic solution of the Eq. (44) for large values of $S \gg 1$ which correspond, basically, to large values of M .

6.1. Polynomial approximation

If $y \in [0, 5[$, the interval is divided into 25 even intervals; the lower ends of such intervals are given by

$$y_i = \frac{i-1}{5}, \quad i = 1, \dots, 26 \quad (45)$$

where the corresponding S_i is obtained from (43) and $M_i \equiv x_i$ is obtained from (44) (values attached in the code). This way, we define twenty five intervals $[x_i, x_{i+1}]$, $i = 1, \dots, 25$. In each of these intervals we introduce a fifth degree polynomial $p_i(x, e)$, $i = 1, \dots, 25$ to interpolate the hyperbolic sine of the hyperbolic anomaly, that is, S .

We note that the coefficients of the polynomials considered in the method depend on the eccentricity. In order to simplify the reading of the equation, we abuse notation by writing the coefficients only as a function of i ,

$$p_i(x) = a_0^{(i)} + a_1^{(i)}x + a_2^{(i)}x^2 + a_3^{(i)}x^3 + a_4^{(i)}x^4 + a_5^{(i)}x^5. \quad (46)$$

The polynomial $p_i(x)$ is only valid inside the interval $[x_i, x_{i+1}]$, $i = 1, \dots, 25$, where the function is approximated by the polynomial: $S(x) \approx p_i(x)$. To do that we follow the procedure developed in Raposo-Pulido & Peláez (2017) by imposing

six boundary conditions to determine the six coefficients of $p_i(x)$. These conditions assure that the function $S(x)$ and the polynomial $p_i(x)$ share the same values and the same first and second derivatives at the ends of the interval $[x_i, x_{i+1}]$.

Modern symbolic manipulators such as Maple or Mathematica allow us to obtain the polynomials $p_i(x)$, $i = 1, \dots, 25$, easily and only once. Moreover, they provide the FORTRAN or C code of the polynomials that will be appropriately stored in one module to be used later. In the HKE–SDG we use the version 18.02 of Maple (Raposo-Pulido & Peláez 2017) to perform this task.

6.2. Asymptotic approximation

We note that the HKE (44) can be expressed as follows:

$$S = f(S, x, e), \quad f(S, x, e) = \frac{1}{e} \ln(S + \sqrt{1 + S^2}) + \frac{x}{e} \quad (47)$$

where f is a function of the variable S ; the values of (x, e) appear as parameters. The derivative of f turns out to be

$$\frac{df}{dS} = \frac{1}{e} \frac{1}{\sqrt{1 + S^2}}. \quad (48)$$

If $y = H \in [5, \infty[$, the corresponding interval for S is given by $[74.203, \infty[$, where the value of x could be very large. In this range x and S tend to infinite, in a first approximation.

The derivative given by Eq. (48) is, obviously, lower than unity, when $y = H \in [5, \infty[$ because $S \gg 1$ and $e > 1$. As a consequence the function $f(S, x, e)$ given by Eq. (47) is a contraction mapping. The Banach fixed-point theorem can be applied and $f(S, x, e)$ admits a unique fixed-point S^* which verify: $S^* = f(S^*, x, e)$. This fixed-point is the root we are looking for.

The mathematical analysis shows that the following sequence S_n defined by

$$S_{n+1} = f(S_n, x, e) \quad (49)$$

which starts in the value $S_0 = x/e$, is convergent and it tends to the limit S^* , that is, to the unique fixed-point.

The speed of convergence of this sequence is larger than usual because of the large values of S in the interval $y \in [5, \infty[$. However, a detailed analysis shows that there are faster algorithms which are also reliable.

In effect, the second term of the right hand side of Eq. (44) becomes, with respect to the first term, very small. As a consequence we can consider the logarithm as a small parameter $\varepsilon = \ln|S + \sqrt{1 + S^2}|/x$, obtaining S by applying an asymptotic expansion in power of ε . Equation (44) takes now the expression

$$x = eS - \varepsilon x. \quad (50)$$

In the case $\varepsilon = 0$ the solution of Eq. (50) is $s_0 = \frac{x}{e}$. This function is known by the a priori data and therefore it is possible to obtain an asymptotic solution of Eq. (50) in the limit when ε tends to zero but for $\varepsilon \neq 0$

$$S = s_0 + s_1\varepsilon + s_2\varepsilon^2 + \dots \quad (51)$$

where the remaining coefficients s_i , $i = 1, 2, \dots$ can be easily obtained by introducing this expansion in (50) and requiring that the resulting series vanishes for every order in ε . The asymptotic solution obtained is

$$S = \frac{x}{e}(1 + \varphi), \quad \varphi = \frac{\varepsilon}{x} \ll 1. \quad (52)$$

We have now to obtain φ as a function of the known parameters x and e . Replacing Eqs. (52) into (44)

$$x = e \left(\frac{x}{e}(1 + \varphi) \right) - \ln \left(\frac{x}{e}(1 + \varphi) + \sqrt{1 + \frac{x^2}{e^2}(1 + \varphi)^2} \right) \quad (53)$$

and taking into account that $\varphi \ll 1$ when x tends to infinite, we compute the first order series expansion of the previous expression with respect to the variable φ

$$0 = -\ln(x + \sqrt{e^2 + x^2}) + \ln(e) + \frac{(\sqrt{e^2 + x^2} - 1)x\varphi}{\sqrt{e^2 + x^2}} \quad (54)$$

such that

$$\varphi = \frac{\sqrt{e^2 + x^2} (\ln(x + \sqrt{e^2 + x^2}) - \ln(e))}{(\sqrt{e^2 + x^2} - 1)x}. \quad (55)$$

Replacing (55) in (52) we obtain S as a function of x and e

$$S = \frac{\sqrt{e^2 + x^2} (\ln(x + \sqrt{e^2 + x^2}) - \ln(e) + x) - x}{(\sqrt{e^2 + x^2} - 1)e}. \quad (56)$$

In order to increase the accuracy of S , we can expand (52) by

$$S = \frac{x}{e} (1 + \varphi + \xi), \quad \xi \ll 1, \quad \xi \ll \varphi \quad (57)$$

where ξ should now be determined as a function of the known parameters x , e and φ . Replacing (57) into (44)

$$x = e \left(\frac{x}{e}(1 + \varphi + \xi) \right) - \ln \left(\frac{x}{e}(1 + \varphi + \xi) + \sqrt{1 + \frac{x^2}{e^2}(1 + \varphi + \xi)^2} \right) \quad (58)$$

and repeating the process for φ , we obtain

$$\xi = -\frac{1}{2x^3} \left[\ln\left(\frac{e}{2x}\right) \right]^2. \quad (59)$$

The final expression for S , replacing (59), is given by

$$S = \frac{1}{2ex^2(\sqrt{e^2 + x^2} - 1)} \left[\sqrt{e^2 + x^2} (2x^2 \ln(x + \sqrt{e^2 + x^2}) - 2x^2 \ln(e)) + (\sqrt{e^2 + x^2} - 1)(2x^3 - (\ln(e) - \ln(x))^2) + \ln(2)(2 \ln(e) - 2 \ln(x) - \ln(2)) \right]. \quad (60)$$

Now we can assess the accuracy of this asymptotic solution just studying the order of magnitude of the variables we have approximated. We note that φ and ξ decrease when e increases. Therefore $e = 1$ marks the lowest accuracy of solution (60), being better when $e > 1$ (see Table 1). If $M \geq 50$, then φ is smaller than 10^{-1} and ξ is smaller in absolute value than 10^{-4} . If $y \geq 5$, we obtain $M \geq 69.2$ when $e = 1$, which assures small values for φ and ξ .

Table 1. Order of magnitude of φ and ξ as a function of M when $e = 1$.

M	50	75	100	200	500
$\varphi \times 10^2$	9.40	6.77	5.35	3.01	1.38
$\xi \times 10^5$	-8.48	-2.98	-1.40	-0.224	-0.0191

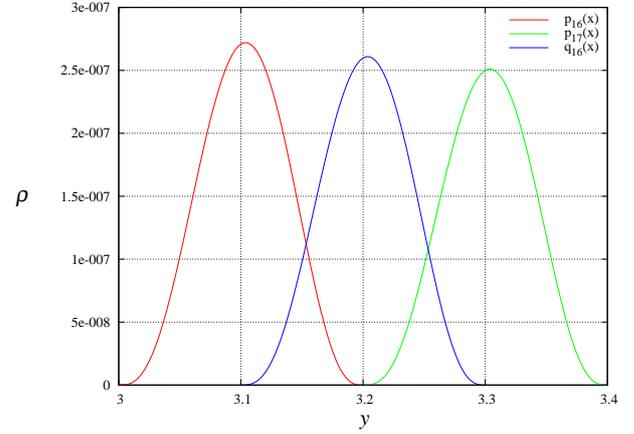


Fig. 3. Residual ρ versus the hyperbolic anomaly y , when the solution is determined by the polynomials $p_{16}(x)$, $p_{17}(x)$ and $q_{16}(x)$.

6.3. Improving the code

The interval $[0, 5]$ of the hyperbolic anomaly $H \equiv y$ has been discretized in 25 intervals. However, this number, related with the number of polynomials, can be modified having a significant impact on the accuracy of the algorithm but not jeopardizing the speed of the code. If we plot the residual ρ caused by the polynomial $p_i(x)$ associated to the interval $[y_i, y_{i+1}]$, we observe that it follows a Gaussian-like function reaching the maximum, approximately, at the centre of the interval (see Fig. 3). One option, to decrease the residual and obtain a better initial seed, is to increase the number of polynomials. But if we decrease the size of the intervals they become too small and the polynomials, which depend on the eccentricity e , can become numerically unstable. However, we can define a new group of polynomials $q_j(x)$ whose ends are defined by the mean points of the initial intervals $[y_i, y_{i+1}]$, $i = 1, \dots, 25$, by keeping the same size (see Fig. 3), that is

$$y_j = \frac{2j-1}{10}, \quad j = 1, \dots, 25, \quad (61)$$

where the corresponding S_j is obtained from Eq. (43) and $M_j \equiv x_j$ is obtained from Eq. (44) (values attached in the code). This way, we define twenty-four intervals $[x_j, x_{j+1}]$, $j = 1, \dots, 24$ and in each one of these intervals we introduce a fifth-degree polynomial $q_j(x)$, $j = 1, \dots, 24$ to interpolate the hyperbolic sine of the hyperbolic anomaly

$$q_j(x) = a_0^{(j)} + a_1^{(j)}x + a_2^{(j)}x^2 + a_3^{(j)}x^3 + a_4^{(j)}x^4 + a_5^{(j)}x^5. \quad (62)$$

The introduction of these new polynomials $q_j(x)$ is not necessary, that is, the algorithm functions normally with the original polynomials $p_i(x)$. However, the tools elaborated to create the polynomial $p_i(x)$ permit to generate the new polynomial $q_j(x)$ without additional effort. The benefit is that the accuracy and reliability of the code is increased.

7. The HKE–SDG algorithm

Given the values of M and e , to solve the hyperbolic Kepler equation we use the following algorithm.

7.1. Seed estimation

Inside the singular corner ($M < 0.15$ and $e < 1.25$). Use the sequence associated with Eq. (37) to estimate the starter value y_0 . The seed S_0 is obtained from the relation: $S_0 = \sinh y_0$.

Outside the singular corner ($M \geq 0.15$ or $e \geq 1.25$). The value of M permits to detect the ends x_k, x_{k+1} of the right interval – one of the intervals related to the polynomials $p_i(x)$ – which contains M , that is, $M \in [x_k, x_{k+1}]$. We note that the corresponding values $[S_k, S_{k+1}]$ are known and the seed S_0 satisfies: $S_0 \in [S_k, S_{k+1}]$. If the right interval is the last one: $[x_{26}, x_{27}]$, the algorithm uses the sequence associated with Eq. (60) to estimate the starter value S_0 (asymptotic approximation). If the right interval is not the last one, the algorithm identifies a new interval from the intervals related to the polynomials $q_j(x)$ containing M . Then it chooses the interval that has M closest to one of their ends. The seed S_0 is given by the corresponding polynomial (46) or (62) (polynomial approximation).

7.2. Root estimation

The seed S_0 is used to solve the equation

$$S - \frac{1}{e} \ln|S + \sqrt{1 + S^2}| - \frac{1}{e}x = 0 \quad (63)$$

by using the Newton–Raphson algorithm. The output of this algorithm provides, through the change in variable (43), the desired value of the hyperbolic anomaly $y(x)$.

A compact version of HKE–SDG has been made available to the reader in the way indicated by the journal under the license GPL Version 3.

8. Analysis of results

A full analysis of the algorithm was performed by considering two main aspects: speed and accuracy. The numerical analysis of HKE–SDG takes into account the ease of codification or the robustness of the procedure too. All these considerations are detailed below. The calculations have been carried out in a computer with an Intel(R) Xeon(R) E3-1535M v6 at 3.10 GHz microprocessor and the executables have been prepared with the version 17.0 of the Intel® C++ Compiler, under the Windows 10 operative system 64 bits.

8.1. Speed of the process

The Kepler equation for the hyperbolic case has been solved $\approx 4 \times 10^6$ times with the algorithm proposed in this paper by using a tolerance ε_{tol} . The eccentricity ranges in the interval $e \in [1, 10]$ and the values of $M \equiv x$ range in the interval $M \in [0, 100]$ (rad) (see Appendix A).

In each run we count the number of iterations needed to reach a solution with a residual $\rho = |e \sinh y - y - x|$ lower than the tolerance ε_{tol} . This number is taken as a gross measure of the speed of the procedure. Figures 4 and 5 show, in the plane (e, M) , the number of iterations associated with HKE–SDG when using the MNR scheme for double and quadruple precision. Notice that in the color code of the figures, green color corresponds to

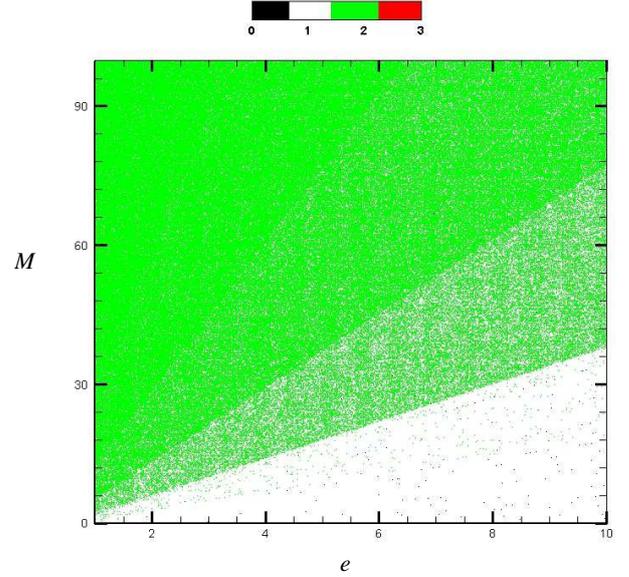


Fig. 4. Results with HKE–SDG using the MNR iteration scheme in double precision.

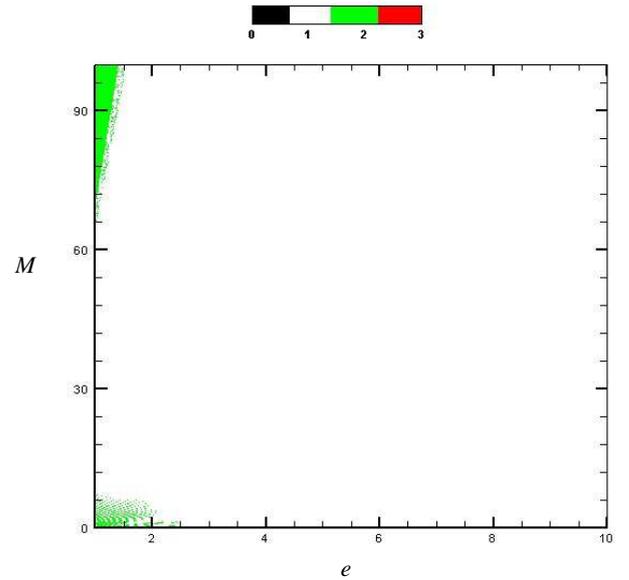


Fig. 5. Results with HKE–SDG using the MNR iteration scheme in quadruple precision.

two iterations, white color to one iteration and black color to zero iterations. The averaged value of iterations is 1.582 and 1.010 for double and quadruple precision respectively.

8.2. Accuracy analysis

As we show in the next section HKE–SDG reaches its best performances when the MNR algorithm is used in iterative scheme. Therefore, this section focuses the analysis in HKE–SDG with the MNR algorithm. In HKE–SDG the iteration ends when: (i) the residual $\rho = |e \sinh y - y - x|$ is lower than the tolerance ε_{tol} , or (ii) the increment $\Delta S_n/S_n$ in the n iteration of the MNR algorithm is lower than the zero of the machine. In order to assess the accuracy of our solution, we use the total absolute error ε_{abs} associated with the numerical solution (see Appendix B).

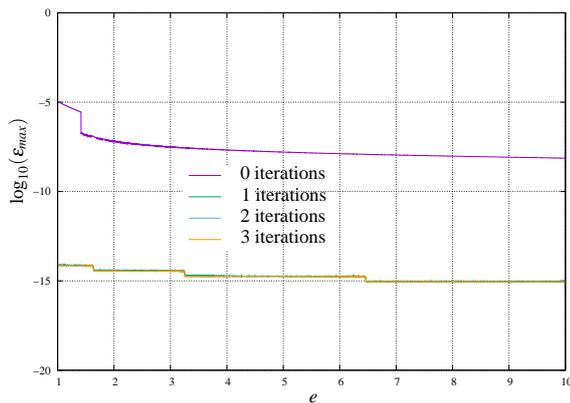


Fig. 6. Maximum total absolute error ε_{\max} versus the eccentricity e , for different number of iterations. Calculations in double precision.

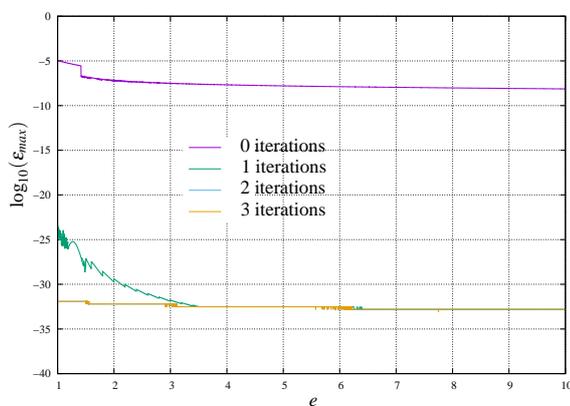


Fig. 7. Maximum total absolute error ε_{\max} versus the eccentricity e , for different number of iterations. Calculations in quadruple precision.

We fix the value of the eccentricity e , in the interval $e \in [1, 10]$; then we scan the whole interval $M \in [0, 100]$ calculating the residual ρ after zero iteration (the residual provided by the starting *seed*), after one iteration, after two iterations and so on. Let ρ_{\max} be the maximum residual that we find in the scanning of the whole interval $M \in [0, 100]$ for each iteration number. These maximum residuals allow us to calculate the maximum values of the absolute error ε_{\max} for different number of iterations, and these absolute errors can be associated with the value of the eccentricity e as it can be shown in Appendix C.

The values of such maximum absolute errors can be plotted versus the eccentricity e . This picture provides a very good idea of the accuracy involved in our calculations after zero, one, two, ... iterations. We note that the values plotted mark the worst accuracy of the numerical solution $y_c(x, e)$; in fact, for each value of e there are hundred of values of M where the accuracy is much better than the indicated by ε_{\max} . As we can see in Figs. 6 and 7 for double and quadruple precision, HKE–SDG saturates the capacity of the machine with a maximum of two iterations.

9. Comparison of results

The Kepler equation for the hyperbolic case has been solved with the method proposed in this paper – HKE–SDG – by using different Newton–Raphson algorithms (see Appendix C). The results for double and quadruple precision confirm that HKE–SDG is almost insensitive to the Newton–Raphson algorithm. The robustness of HKE–SDG is due to the strong optimization

Table 2. Number of iteration (percentage) and computational time obtained with HKE–SDG and Avendaño algorithms checked in double (upper) and quadruple (lower) precision.

Scheme	$i = 0$	$i = 1$	$i = 2$	$i \geq 3$	CPU time
MNR	0.126	41.597	58.277	0	1.398
	0.050	98.899	1.051	0	34.267
Avendaño	0	0.058	36.799	63.143	1.657
	0	0	7.163	92.837	31.422

of the initial seed performed in the algorithm; this optimization prevents the convergence problems which could appear in the Newton–Raphson iteration when the initial seed is far away from the real root. Moreover, the same analysis has been performed with typical solvers of the Kepler equation which have been compared with HKE–SDG when using the MNR method. In this section we collect the method described in Avendano et al. (2015) (see Table 2), which is the closest to our code. The comparisons with the remaining solvers can be seen in Appendix C. In each case the HKE has been solved under the conditions described in Sect. 8 and in Sect. 8.1, that is, we solve equation $\approx 4 \times 10^6$ times.

In double precision the HKE–SDG never requires, when using the MNR method, more than two iterations. Notice that the approach described in Avendano et al. (2015) requires more than three iterations for some particular cases, reaching a maximum value of 19 iterations. In double precision the Avendaño code requires 1.657 s with an averaged number of iterations of 2.672. In the case of HKE–SDG the average number was 1.582 with a CPU time of 1.398 s.

Table 2 also contains the results in quadruple precision. When using the MNR iteration scheme, HKE–SDG reaches the solution with only one or two iterations in the 99.950% of cases and in the 0.050% no iteration is necessary (see Fig. 5). The HKE–SDG keeps again the smallest number of iterations. In quadruple precision HKE–SDG reaches a maximum number of two iterations when applying the MNR method, while the method described in Avendano et al. (2015) reaches a maximum number of 21 iterations. The CPU time invested was 34.267 s with HKE–SDG and 31.422 s with Avendaño code. The averaged number of iterations was 1.010 and 2.976 for HKE–SDG and Avendaño respectively.

Figure 8 shows the number of significant digits obtained with the Avendaño and HKE–SDG codes when using the MNR method in double precision. Darker colors are associated with less precision as well as lighter colors with better precision. At first sight the two pictures are quite similar and only small differences can be appreciated. If we compare HKE–SDG with respect to the Avendaño approach, the results obtained with the last one show a lower precision for large values of eccentricity ($e > 8$) and small values of mean anomaly ($M \leq 5$).

Figure 9 shows the number of significant digits when quadruple precision is used in calculations. We note that the Avendaño approach shows radial beams with less precision than HKE–SDG. In fact, HKE–SDG provides the most precise solution with more than 33 significant digits in the whole plane (e, M) when quadruple precision is applied. As it is shown, HKE–SDG experiences an improvement of 20 significant figures for almost the complete plane when changing from double to quadruple precision. The results obtained with HKE–SDG when using different numerical methods are very similar and all of them assure the reliability and quality of the solution.

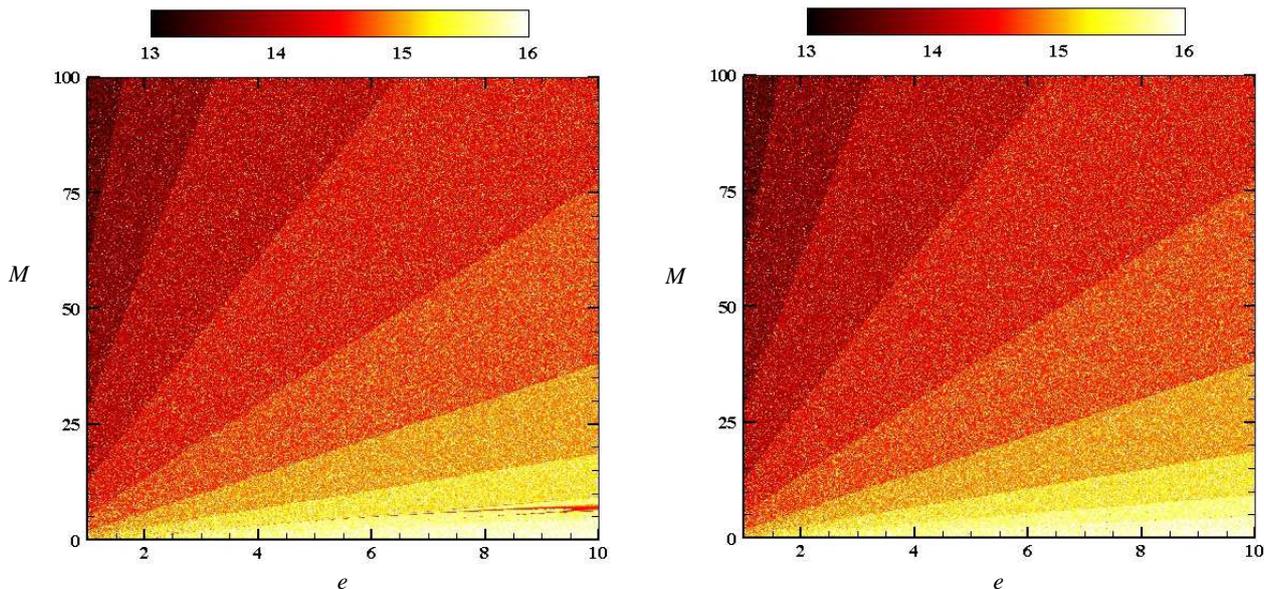


Fig. 8. Number of significant figures, approximately, obtained with the code of Avendano et al. (2015) (*left*) and HKE-SDG (*right*) in double precision.

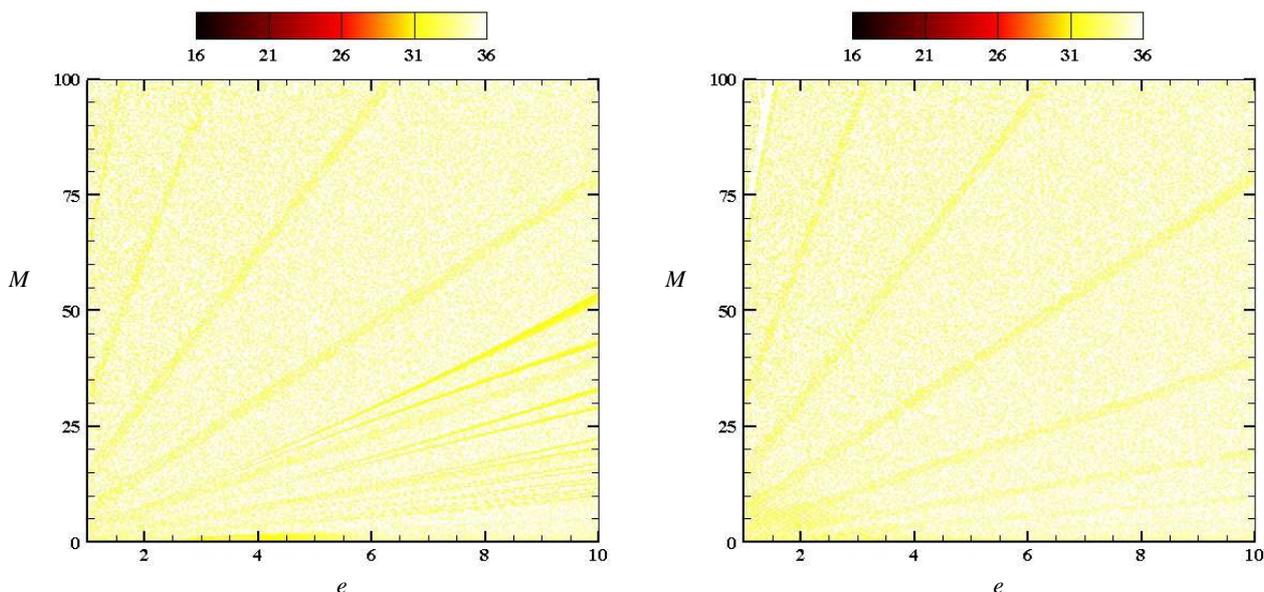


Fig. 9. Number of significant figures, approximately, obtained with the code of Avendano et al. (2015) (*left*) and HKE-SDG (*right*) in quadruple precision.

In summary, when double precision is used, HKE-SDG turns out to be the fastest method and it provides the most precise results. When quadruple precision is needed, HKE-SDG is the most accurate, stable and reliable even if it turns out to be a little slower.

10. Conclusions

Several conclusions can be drawn from the results obtained of our numerical research about the solution provided by the code HKE-SDG:

Firstly, convergence is always assured – as confirmed the thorough the analysis carried out in our computers – when the initial seed of the Newton-Raphson algorithm is optimized.

Secondly, the algorithm used in HKE-SDG is stable and reliable regardless of the Newton-Raphson algorithm used in the

numerical solution. We note that – with double precision – the classical Newton-Raphson provides the solution in the 99.866% of cases with only one or two iterations. This remarkable result has not been found in other algorithms used in the Astrodynamics community. Improving the algorithm by using the Conway or the modified Newton-Raphson permits to obtain the right solution with only one or two iterations in the 99.868% and 99.874% of cases respectively. The reliability is also improved. Gooding, Fukushima, and Avendaño codes require more than three iterations in 0.088%, 100% and ~63%–92% of cases respectively.

Thirdly, if a very precise answer is needed, HKE-SDG can be used with quadruple precision due to the low number of iterations required. HKE-SDG gives the maximum precision, it combines stability and reliability and the speed of the calculations is not jeopardized. Therefore, we obtain the benefit of a greater accuracy with a minimal cost.

Fourthly, if quadruple precision is used, the absolute error of the numerical solution provided by HKE–SDG is clearly under 10^{-24} after one iteration. In any case, after two iterations the error is always lower than 10^{-31} , including the region $e < 1.25$ (the singular corner).

Finally, the algorithm has a global reach since it successfully solves the Kepler equation in the singular corner, $M < 0.15$ and $e < 1.25$. The asymptotic expansions used to generate the initial seed assure the reliability and convergence of the Newton–Raphson iterative scheme. The convergence in this special region not always is assured by some of the algorithms usually considered in the literature. Gooding code shows a deterioration in a small region of the singular corner.

Acknowledgements. This work is part of the project entitled Dynamical Analysis of Complex Interplanetary Missions with reference ESP2017-87271-P, supported by Spanish Agencia Estatal de Investigación (AEI) (Spanish Research Agency) of Ministerio de Economía, Industria y Competitividad (MINECO) (Ministry of Economy, Industry and Competitiveness) and by European Found of Regional Development (FEDER). Authors thank to the referee for the constructive comments and suggestions which helped to improve the paper.

References

- Alhindi, F. A., & Sharaf, M. 2014, *J. Adv. Math.*, **10**, 3457
- Avendano, M., Martín-Molina, V., & Ortigas-Galindo, J. 2015, *Celest. Mech. Dyn. Astron.*, **123**, 435
- Balaraman, G., & Vrinceanu, D. 2007, *Phys. Lett. A*, **369**, 188
- Battin, R. 1964, *Astronautical Guidance* (New York: McGraw Hill)
- Beust, H., Bonnefoy, M., Maire, A.-L., et al. 2016, *A&A*, **587**, A89
- Colwell, P. 1993, *Solving Kepler’s Equation over three Centuries* (Richmond, Virginia: Willmann-Bell)
- Conway, B. A. 1986, *Celest. Mech.*, **39**, 199
- Davidson, M. 1932, *MNRAS*, **93**, 33
- Davidson, M. 1934, *MNRAS*, **95**, 66
- Dehnen, W., & Hernandez, D. M. 2016, *MNRAS*, **465**, 1201
- Fäintich, M. B. 1972, *Celest. Mech.*, **6**, 22
- Fukushima, T. 1997, *Celest. Mech. Dyn. Astron.*, **68**, 121
- Gooding, R. 1987, *Universal Procedures for Conversion of Orbital Elements to and from Position and Velocity (unperturbed orbits)*, Tech. rep., Royal Aircraft Establishment
- Gooding, R., & Odell, A. 1988, *Celest. Mech.*, **44**, 267
- Goodyear, W. H. 1965, *AJ*, **70**, 189
- Hernandez, D. M., & Bertschinger, E. 2015, *MNRAS*, **452**, 1934
- Herrick, S. 1971, *Astrodynamics: Orbit Correction, Perturbation Theory, Integration* (New York: Van Nostrand Reinhold)
- Kurth, R. 1978, *Q. Appl. Math.*, **36**, 325
- Milani, A., & Nobili, A. M. 1987, *Celest. Mech.*, **43**, 1
- Milani, A., Nobili, A., Fox, K., & Carpino, M. 1986, *Nature*, **319**, 386
- Pitkin, E. T. 1965, *AIAA J.*, **3**, 1508
- Raposo-Pulido, V., & Peláez, J. 2017, *MNRAS*, **467**, 1702
- Rauh, A., & Parisi, J. 2011, *Phys. Rev. A*, **83**, 042101
- Rein, H., & Tamayo, D. 2015, *MNRAS*, **452**, 376
- Sarnecki, A. 1988, *Acta Astronaut.*, **17**, 881
- Sharaf, M., Banajh, M., & Alshaary, A. 2007, *JApA*, **28**, 9
- Sharaf, M. A., & Sharaf, A. A. 2003, *Celest. Mech. Dyn. Astron.*, **86**, 351
- Sitarski, G. 1968, *Acta Astron.*, **18**, 197
- Smiley, C. 1934, *MNRAS*, **95**, 63
- Stumpff, K. 1968, *On the Application of Spinors to the Problem of Celestial Mechanics*, Tech. rep., National Aeronautics and Space Administration
- Tokis, J. N. 2014, *Int. J. Astron. & Astrophys.*, **4**, 683
- Wisdom, J., & Hernandez, D. M. 2015, *MNRAS*, **453**, 3015
- Wisdom, J., & Holman, M. 1991, *AJ*, **102**, 1528
- Yabushita, S. 1971, *MNRAS*, **153**, 97

Appendix A

By applying the MNR algorithm we have carried out the calculations in standard double precision and quadruple precision with a tolerance equal to: $\varepsilon_{\text{tol}} = 2.22 \times 10^{-16}$ and $\varepsilon_{\text{tol}} = 1.0 \times 10^{-32}$ respectively. The method used in this study is a numerical procedure based on successive approximations. It means if the tolerance considered is close to the zero of the machine, the method could experience artificial numerical chaos. A way to avoid this problem is by using an appropriate tolerance.

Figure 4 shows some black points; most of them on the axis $M = 0$. The red color (three iterations) does not appear in the figure.

Figure 5 shows the equivalent picture obtained with quadruple precision. Practically the code performs one iteration everywhere except in two zones where two iterations are needed: the singular corner and the zone with large values of M and small values of e . In both zones the seed is generated with asymptotic expansions. The algorithm never enters dead loops or any other kind of nonconvergent phase after many millions of runs. This fact confirms the reliability of the algorithm.

Appendix B

Let $y_v(x, e)$ be the true solution for given values of e and the mean anomaly $M \equiv x$. Let $y_c(x, e)$ be the numerical solution provided by HKE–SDG for these particular values. Due to truncation and round-off errors, both solutions are related by the equation

$$y_c = y_v + \varepsilon_{\text{abs}} \quad (\text{B.1})$$

where ε_{abs} is the total absolute error associated with the numerical solution $y_c(x, e)$. This absolute error is closely related with the residual ρ . In effect, for the numerical solution $y_c(x, e)$ the residual is given by

$$\rho = |e \sinh(y_v + \varepsilon_{\text{abs}}) - (y_v + \varepsilon_{\text{abs}}) - x|. \quad (\text{B.2})$$

Since ε_{abs} is small and taking into account that the true solution verifies

$$e \sinh y_v - y_v - x = 0 \quad (\text{B.3})$$

the residual takes the form

$$\rho = |\varepsilon_{\text{abs}}| |e \cosh y_v - 1| + \mathcal{O}(\varepsilon_{\text{abs}}^2). \quad (\text{B.4})$$

Therefore the absolute error can be approximated by

$$|\varepsilon_{\text{abs}}| = \frac{\rho}{|e \cosh y_v - 1|} \quad (\text{B.5})$$

and depends on the residual ρ . Something similar happens with the total relative error:

$$|\varepsilon_{\text{rel}}| = \frac{|y_c - y_v|}{|y_v|} = \frac{\rho}{|y_v(e \cosh y_v - 1)|}. \quad (\text{B.6})$$

Figure 6 shows the decimal logarithm of the maximum absolute error versus the eccentricity e . The initial seed – 0 iterations – provides an error which, in the worst cases, ranges from 10^{-5} to 10^{-8} . Since the calculations associated with Fig. 6 have been carried out in double precision, the residual obtained after one iteration practically saturates the capacity of the machine providing the solution with 14 or 15 significant figures. We emphasize that one additional iteration does not improve the accuracy of the numerical solution; figure shows that the lines corresponding to one, two, and three iterations are, practically, the same.

Table C.1. Number of iteration (percentage) and computational time obtained with HKE–SDG by using different Newton–Raphson methods checked in double (upper) and quadruple (lower) precision.

Scheme	$i = 0$	$i = 1$	$i = 2$	$i \geq 3$	CPU time
MNR	0.126	41.597	58.277	0	1.398
	0.050	98.899	1.051	0	34.267
Conway	0.126	41.498	58.370	0.006	1.297
	0.050	0.455	99.416	0.079	38.677
CNR	0.126	41.481	58.385	0.008	1.145
	0.050	0.430	99.427	0.093	35.733

Table C.2. Number of iteration (percentage) and computational time obtained with HKE–SDG, Gooding and Fukushima algorithms checked in double (upper) and quadruple (lower) precision.

Scheme	$i = 0$	$i = 1$	$i = 2$	$i \geq 3$	CPU time
MNR	0.126	41.597	58.277	0	1.398
	0.050	98.899	1.051	0	34.267
Gooding	0.050	0	99.862	0.088	1.808
	0.050	0	99.862	0.088	24.684
Fukushima	0	0	2×10^{-4}	100	28.669

Figure 7 is similar to Fig. 6 but with the calculations carried out in quadruple precision. It is clear that with one iteration we obtain more than 31 significant figures except for values of e close to unity, where the number of significant digits is about 24–31. With two iterations HKE–SDG saturates the precision capacities of the machine, providing more than 31 significant figures. It should be underlined that one additional iteration does not improve the accuracy of the numerical solution; figure shows that the lines corresponding to two and three iterations are, practically, the same.

Appendix C

Three different Newton–Raphson algorithms have been performed in the study to analyze the quality of the algorithm: the MNR method, the Conway method, and the CNR method.

Table C.1 summarizes the behavior of HKE–SDG in double precision by using these algorithms as well as the CPU time invested in the $\approx 4 \times 10^6$ of times that we solved the Kepler equation. When the MNR was used, in the 99.874% of cases we reach the solution with only one or two iterations and in the 0.126% no iteration is necessary. With the Conway or the CNR in a very small number of cases three iterations are needed; the averaged number of iterations was 1.583 for both methods. With any of the Newton–Raphson algorithms and using double precision, it took 1.3 s approximately to solve the whole set. The CPU time was 1.398 s when using the most reliable scheme: the MNR.

Table C.1 summarizes as well the behavior of HKE–SDG in quadruple precision. HKE–SDG reaches a maximum number of two iterations when applying the MNR method and three iterations ($\leq 0.1\%$) when working with the Conway and CNR methods. The CPU time invested was 34.267 s (MNR), 38.677 s (Conway) and 35.733 s (CNR), while the averaged number of iterations was 1.010, 1.995 and 1.996 respectively.

In addition to the Avendaño approach, two different solvers of the Kepler equation have been performed in the study

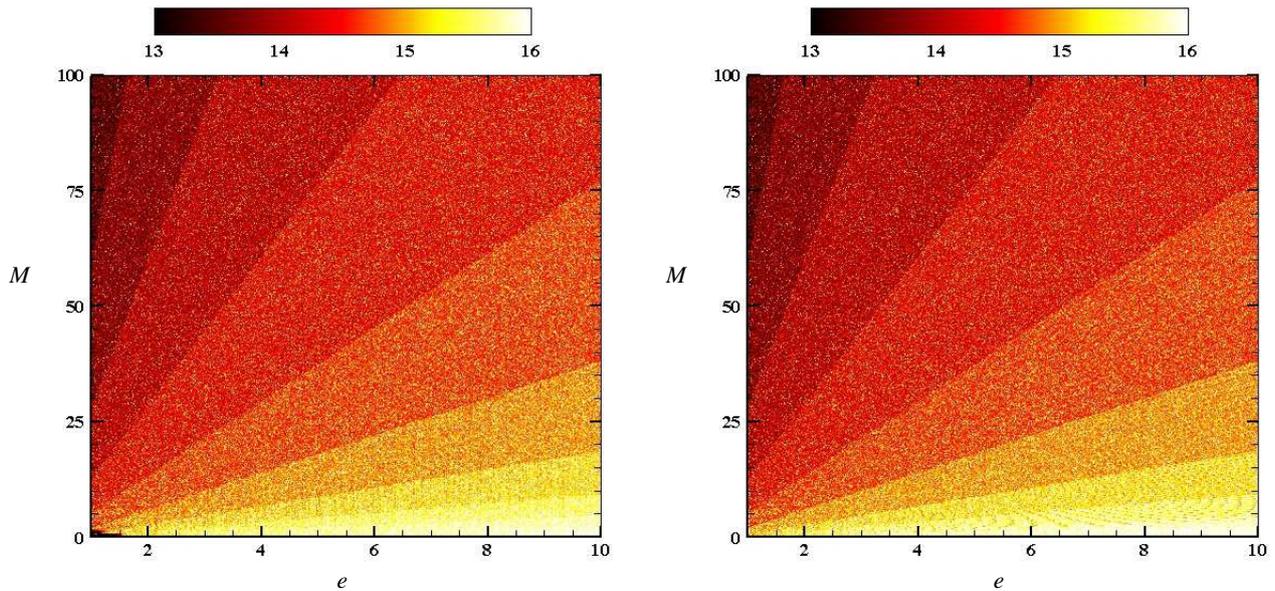


Fig. C.1. Number of significant figures, approximately, obtained with the code of Gooding & Odell (1988; *left*) and Fukushima (1997; *right*) in double precision.

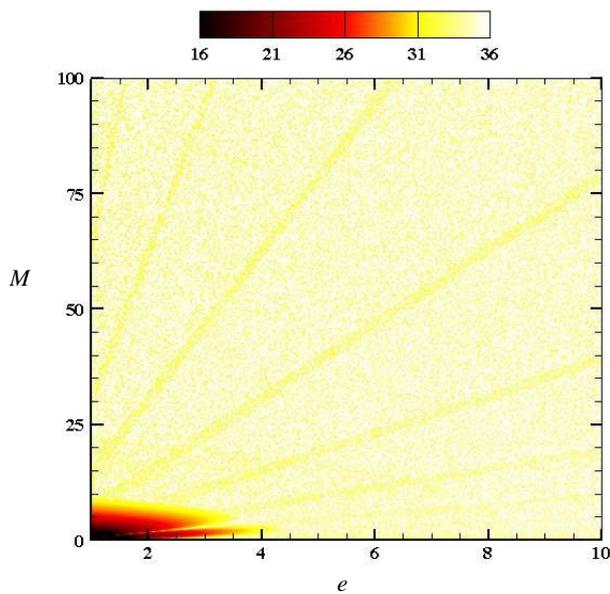


Fig. C.2. Number of significant figures, approximately, obtained with the code of Gooding & Odell (1988) in quadruple precision.

by using the MNR method. Firstly the method described in Gooding & Odell (1988), and secondly the method described in Fukushima (1997).

In double precision the approaches described in Gooding & Odell (1988) and Fukushima (1997) require more than three iterations for some particular cases, reaching a maximum value of 43 and 20 respectively. In the case of HKE-SDG the maximum is two. In double precision the Gooding code requires 1.808 s, while the Fukushima code requires 28.669 s. The averaged number of iterations was 2.014 and 6.117 respectively. In the case of HKE-SDG the average

number was smaller than two as well as the CPU time (see Table C.2). The reason why the Fukushima code takes much more time than the rest of the codes is the very strict tolerance that we use: the zero of the machine. The Fukushima algorithm has been developed to work in double precision by evaluating the Kepler equation by a Taylor series expansion with the appropriate number of terms. Due to that truncation, to reach the precision associated to the ε_{tol} turns out to be difficult and it requires more time than habitual. With not-so-strict tolerances the differences are much smaller.

Table C.2 also contains the results in quadruple precision. We note that the Fukushima code does not include quadruple precision. The use of quadruple precision would imply to extend the original series, however, the authors prefer to keep the algorithm as it is explained in Fukushima (1997). When using the MNR iteration scheme HKE-SDG reaches the solution with a maximum of two iterations. The method described in Gooding & Odell (1988) does not experiment almost any change, keeping the same percentages as in double precision. In quadruple precision the Gooding code reaches a maximum number of 94 iterations with a CPU time of 24.684 s and an averaged number of iterations of 2.035.

Figure C.1 shows the number of significant digits obtained with the Gooding and Fukushima codes when using the MNR method in double precision. If we compare HKE-SDG with respect to the other approaches, the results obtained with the Gooding code show a lower precision in a small region inside the singular corner. The number of digits in Fukushima is very similar to HKE-SDG, however, for some specific points in the yellow region ($M \leq 5$) the Fukushima code reaches less precision.

Figure C.2 shows the number of significant digits when quadruple precision is used in calculations. We note that the Gooding approach shows again a significant deterioration in the singular corner.