

Improved First Estimates to the Solution of Kepler's Equation

Richard J. Mathar*

Max-Planck Institute of Astronomy, Königstuhl 17, 69117 Heidelberg, Germany

(Dated: February 11, 2019)

The manuscript provides a novel starting guess for the solution of Kepler's equation for unknown eccentric anomaly E given the eccentricity e and the mean anomaly M of an elliptical orbit.

PACS numbers: 95.10.Jk, 95.75.Pq, 91.10.Ws

Keywords: Kepler's Equation, Inverse Problem, Orbital Elements

KEPLER'S EQUATION

Mean and Eccentric Anomaly

The track of the orbit for a 2-body potential proportional to the inverse distance of the two bodies leads to solutions which may be ellipses with eccentricity $0 \leq e \leq 1$. The time dependence is described by the parameter M of the mean anomaly, which is an angle measured from the center of the ellipse, and which is a product of a parameter n called the mean motion (essentially the square root of the coupling parameter in the numerator of the 2-body potential divided by the cube of the major semi-axis) and a time elapsed since some reference epoch:

$$M = n(t - t_0). \quad (1)$$

For the manuscript at hand, M and e are considered fixed parameters. To compute the circular coordinates of distance and true anomaly of the body at that time in the reference frame centered at the ellipse, one encounters Kepler's equation

$$E = M + e \sin E \quad (2)$$

$$E - M - e \sin E = 0. \quad (3)$$

E and M are angles measured in radian in the range $-\pi \leq E, M \leq \pi$. To simplify the notation, we discuss only the cases where $M \geq 0$, because the parity

$$E(-M) = -E(M), \quad (4)$$

—equivalent to flipping the entire orbit along the major axis of the ellipse—allows to recover solutions for negative M as well.

The Inverse Problem

The numerical problem considered here is to find the root of the function

$$f(E) \equiv E - M - e \sin E \quad (5)$$

in an efficient and numerically stable fashion.

The expansion of E in a Taylor series of e can be written as

$$E = \sum_{i \geq 0} m_i(M) e^i \quad (6)$$

supported by the table

i	$m_i(M)$
0	M
1	S
2	CS
3	$S(3C^2 - 1)/2$
4	$CS(8C^2 - 5)/3$
5	$S(125C^4 - 114C^2 + 13)/24$
6	$CS(162C^4 - 194C^2 + 47)/15$
7	$S(16087C^6 - 24915C^4 + 9369C^2 - 541)/720$
8	$CS(16384C^6 - 28950C^4 + 14838C^2 - 1957)/325$

where we have written $C \equiv \cos(M)$ and $S \equiv \sin M$ to tighten the notation. The standard problem with this series is that the m_i do not fall in magnitude as a function of i ; so the Taylor expansion is not converging well unless e is close to zero.

NEWTON METHODS

The simplest technique of solving (2) is a fixed point iteration

$$E^{(i+1)} = M + e \sin E^{(i)}; \quad E^{(0)} = M. \quad (7)$$

For faster convergence this is commonly replaced by a first-order Newton iteration

$$E^{(i+1)} = E^{(i)} - \frac{f}{f'} \quad (8)$$

or a second-order Newton iteration [1–5]

$$E^{(i+1)} = E^{(i)} - \frac{f}{f'(1 - \frac{ff''}{2f'^2})} = E^{(i)} - \frac{2ff'}{2f'^2 - ff''}, \quad (9)$$

where the function and its derivatives with respect to the unknown E are

$$f \equiv E - e \sin E - M; \quad f' \equiv 1 - e \cos E; \quad f'' \equiv e \sin E. \quad (10)$$

(8) is for example used in the `iauPlan94` IAU function for planets' ephemerides [6]. Note that, since the evaluation of the trigonometric functions is expensive compared to the fundamental operations [7, 8], the second-order iteration is preferred since $\sin E$ in f'' is already calculated in conjunction with f .

INITIAL VALUE PROBLEM

Standard Initial Guesses

If the initial guess is the second step of (7),

$$E^{(0)} = M + e \sin M, \quad (11)$$

and the iteration (8) is used with $e > 0.99$, a known problem is that the iterations may converge to secondary roots of the equation with the wrong sign [9]. This is basically triggered by starting with an underestimate of E as illustrated in Figure 1.

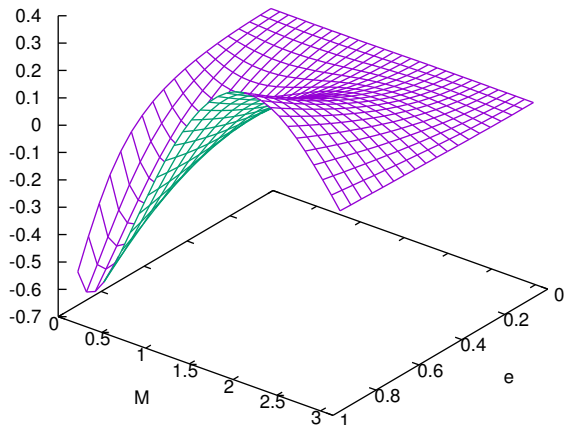


FIG. 1. Mismatch $E^{(0)} - E$ of the initial estimate (11).

A well-known remedy is to start with the initial guess

$$E^{(0)} = \pi \quad (12)$$

which is known to converge [10, 11]. The speed of convergence with the two basic Newton methods is illustrated in Figures 2 and 3.

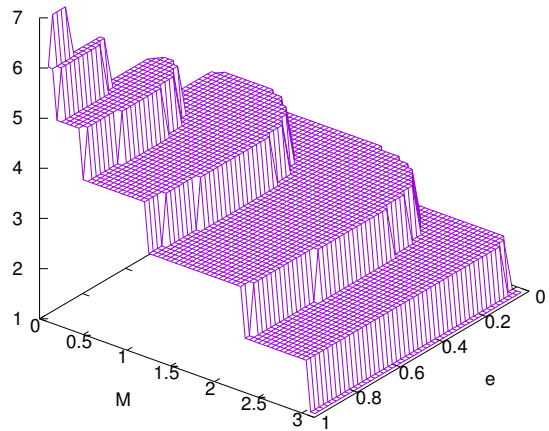


FIG. 2. The number of iterations needed for a relative accuracy of 10^{-12} in E starting from (12) iterating with (8).

Improved Initial Value, Version 1

A starting value of E is obtained by inserting the approximation

$$\sin E \approx 1 - \frac{4}{\pi^2}(E - \pi/2)^2 \quad (13)$$

into the equation. [Similar approximations could be obtained by truncating the Chebyshev series approximation of the $\sin E$ after the second term [12, 13].]

This leads to a quadratic equation for E

$$E^{(0)} = M + e \left[1 - \frac{4}{\pi^2}(E^{(0)} - \pi/2)^2 \right], \quad (14)$$

which is solved by

$$\bar{e} \equiv \frac{\pi}{4e} - 1; \quad (15)$$

$$E^{(0)} = \frac{\pi}{2} \bar{e} \left[\operatorname{sgn}(\bar{e}) \sqrt{1 + \frac{M}{e\bar{e}^2}} - 1 \right]. \quad (16)$$

The error of this estimate relative to the accurate solution is shown in Figure 4. It increases where $M/e \rightarrow 0$ and $\bar{e} \rightarrow 0$. The figure shows that the (16) has the same benefit as (12) of approximating the solution from above, therefore converging [10], but being more accurate. In consequence, the convergence is faster, as demonstrated in Figure 5 if compared with Figure 3.

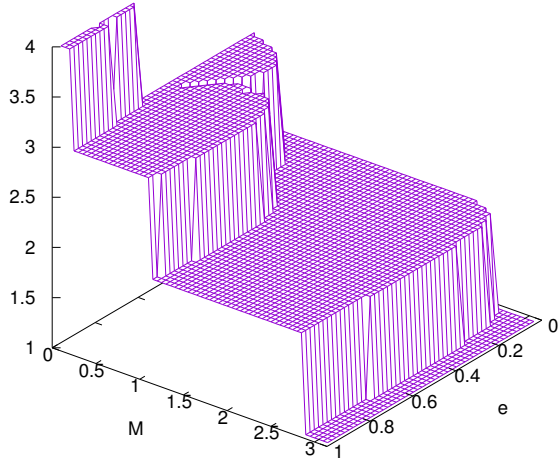


FIG. 3. The number of iterations needed for a relative accuracy of 10^{-12} in E starting from (12) iterating with (9).

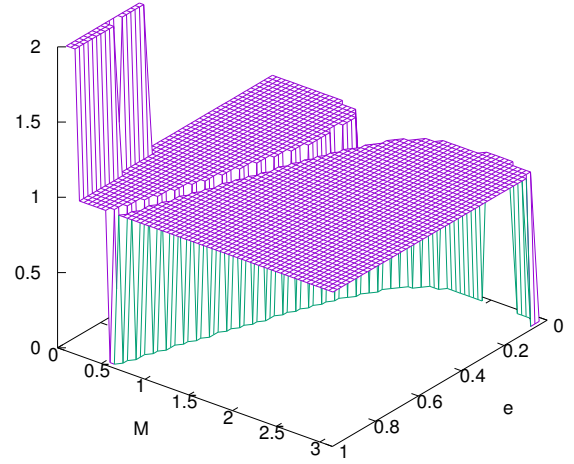


FIG. 5. The number of iterations needed for a relative accuracy of 10^{-12} in E starting from (16) iterating with (9).

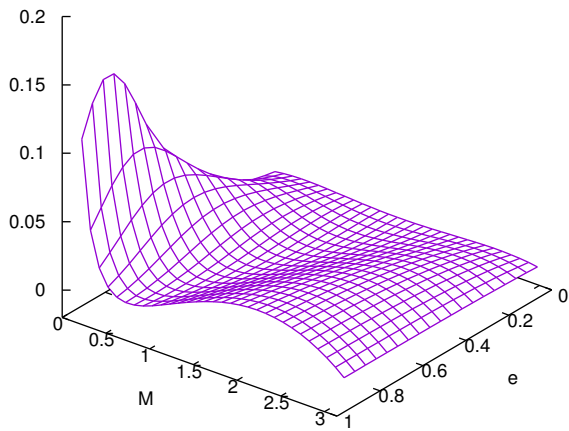


FIG. 4. Mismatch $E^{(0)} - E$ of the initial estimate (16).

Improved Initial Value, Version 2

If (3) is expressed as

$$E - M = e \sin(M + E - M), \quad (17)$$

both sides may be expanded in a Taylor series of $E - M$,

$$E - M \approx e \sin(M) + (E - M)e \cos M - \frac{(E - M)^2}{2} e \sin M + \dots \quad (18)$$

Keeping this series up to $O(E - M)$ yields the estimate

$$E^{(0)} = M + \frac{e \sin(M)}{1 - e \cos M}. \quad (19)$$

This is basically the estimate of the second step of the fixed point iteration (7) with an enhancement factor of the second term if e or $\cos M$ are large. As pointed out earlier [1], this is also obtained applying the Newton method to the estimator $E^{(0)} = M$.

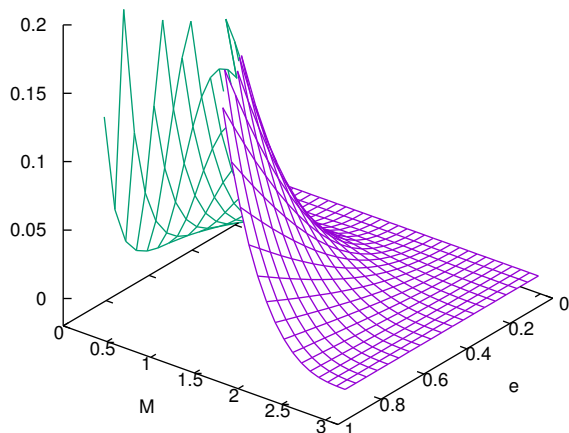


FIG. 6. Mismatch $E^{(0)} - E$ of the initial estimate (19).

If the series is kept up to $O((E - M)^2)$, the associated

quadratic equation proposes

$$\frac{e}{2} \sin M (E^{(0)} - M)^2 + (1 - e \cos M)(E^{(0)} - M) - e \sin M = 0. \quad (20)$$

This quadratic equation is solved by

$$E^{(0)} - M = \frac{1 - e \cos M}{e \sin M} \left[\sqrt{1 + \frac{2e^2 \sin^2 M}{(1 - e \cos M)^2}} - 1 \right]. \quad (21)$$

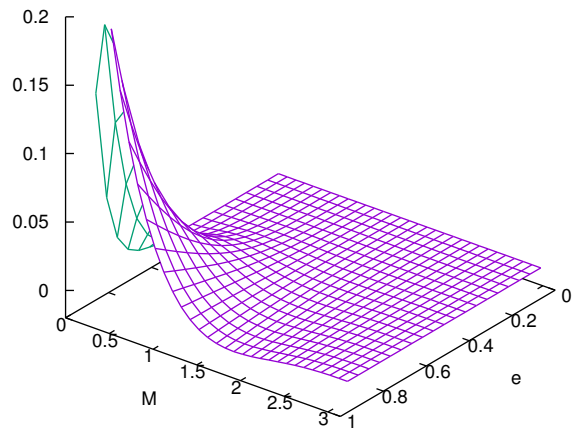


FIG. 7. Mismatch $E^{(0)} - E$ of the initial estimate (21).

Figures 6 and 7 show in comparison with Figure 4 that these approximations derived from the Taylor series of $E - M$ are not better than the one from the quadratic estimate of $\sin E$.

Expansion of (18) up to third order in $E - M$ yields a cubic equation for $E^{(0)} - M$, which is even closer to the exact solution as demonstrated in Figure 8. See Mikkola's paper for a similar approach [14–16].

SUMMARY

A initial value (16) combined with Halley's equation (9) leads to fast and stable convergence for the inverse problem of Kepler's equation for elliptic orbits. That initial value is simpler but generally *worse* than Markley's estimator [16].

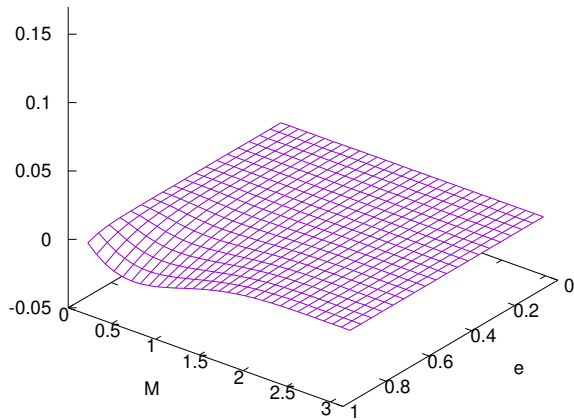


FIG. 8. Mismatch $E^{(0)} - E$ of the initial estimate of the cubic order of (18).

C++ Example

A reference implementation is reprinted here which implements (9) and (16). If compiled with the `-DTEST` pre-processor symbol, a test program is obtained which can be called with two command line arguments, e and M , to investigate the convergence with this and other approaches.

```
#include <cmath>
#include <iostream>

/** short cut for the sign of an expression
 */
#define SGN(a) ((a)>0?-1)

using namespace std;

/** Solve the Kepler equation starting at an improved guess.
 * The numeric routine proposed in vixra:1705.0216
 * @param e ellipticity in the range [0..1]
 * @param M mean anomaly in the range [-pi..pi]
 * @param eps required relative accuracy
 * @return eccentric anomaly in the range [-pi..pi]
 * @author R. J. Mathar
 * @since 2017-05-13
 */
double solveKepler(const double e, double M, const double eps)
{
    M = remainder(M,2.*M_PI) ;
    if ( M < 0.)
        return -solveKepler(e,-M,eps) ;

    const double aux =0.25*M_PI/e-1.0 ;
    double E=0.5*M_PI*aux*(
        SGN(aux)*sqrt(1+M/(e*aux*aux))-1.0) ;
    for (int itr= 1 ; itr < 10 ; itr++)
```

```

{
#ifdef HAVE_SINCOS
    double se,ce ;
    sincos(E, & se, & ce) ;
    const double f2 = e*se ;
    const double f1 = 1.-e*ce ;
#else
    const double f2 = e*sin(E) ;
    const double f1 = 1.-e*cos(E) ;
#endif /* HAVE_SINCOS */
    const double f = E-f2-M ;
    const double Ediff = f*f1/(f1*f1-0.5*f*f2) ;
    E -= Ediff ;
#ifdef TEST
    cout << e << " " << M << " " << itr
        << " " << E << " " << Ediff << endl ;
#else
    if ( fabs(Ediff) < eps*fabs(E) )
        break ;
#endif /* TEST */
}
return E ;
} /* solveKepler */

#ifdef TEST
/** Solve the Kepler equation starting at Pi.
 * This serves as a reference implementation which is known to
 * converge.
 * @param e ellipticity in the range [0..1]
 * @param M mean anomaly in the range [-pi..pi]
 * @return eccentric anomaly in the range [-pi..pi]
 */
double solveKeplerRef(const double e, double M)
{
    M = remainder(M,2.*M_PI) ;
    if ( M < 0.)
        return -solveKeplerRef(e,-M) ;

    double E=M_PI ;
    for (int itr= 1 ; itr < 10 ; itr++)
    {
        double f = E-e*sin(E)-M ;
        double f1 = 1.-e*cos(E) ;
        double f2 = e*sin(E) ;
        double Ediff = f*f1/(f1*f1-0.5*f*f2) ;
        E -= Ediff ;
        cout << e << " " << M << " " << itr
            << " " << E << " " << Ediff << endl ;
    }
    cout << endl ;
    return E ;
} /* solveKeplerRef */

/** Solve the Kepler equation starting at Mikkola's estimator.
 * An implementation according to S. Mikkola, Cel. Mech. 40 (1987) 329.
 * @param e ellipticity in the range [0..1]
 * @param M mean anomaly in the range [-pi..pi]
 * @return eccentric anomaly in the range [-pi..pi]
 * @author R. J. Mathar
 * @since 2019-02-11
 */
double solveKeplerMikkola(const double e, double M)
{
    M = remainder(M,2.*M_PI) ;

```

```

if ( M < 0.)
    return -solveKeplerMikkola(e,-M) ;

const double alpha = (1.0-e)/(4.0*e+0.5) ;
const double beta = 0.5*M/(4.0*e+0.5) ;
double z =sqrt(beta*beta+pow(alpha,3.0)) ;
if ( beta > 0.0)
    z = cbrt(beta+z) ;
else
    z = cbrt(beta-z) ;
const double s = z-alpha/z ;

double E=M+e*s*(3.0-4.0*s*s) ;
for (int itr= 1 ; itr < 10 ; itr++)
{
    double f = E-e*sin(E)-M ;
    double f1 = 1.-e*cos(E) ;
    double f2 = e*sin(E) ;
    double Ediff = f*f1/(f1*f1-0.5*f*f2) ;
    E -= Ediff ;
    cout << e << " " << M << " " << itr
        << " " << E << " " << Ediff << endl ;
}
cout << endl ;
return E ;
} /* solveKeplerRef */

/** Solve the Kepler equation starting at Markley's estimator.
 * An implementation according to F. L. Markley, Cel. Mech. Dyn. Astr. 63 (1995) 101
 * @param e ellipticity in the range [0..1]
 * @param M mean anomaly in the range [-pi..pi]
 * @return eccentric anomaly in the range [-pi..pi]
 * @author R. J. Mathar
 * @since 2019-02-11
 */
double solveKeplerMarkley(const double e, double M)
{
    M = remainder(M,2.*M_PI) ;
    if ( M < 0.)
        return -solveKeplerMarkley(e,-M) ;

    const double alpha = M_PI*(3.*M_PI+1.6*(M_PI-M)/(1.0+e))/(M_PI*M_PI-6.0) ;
    const double d = 3.0*(1.0-e)+alpha*e ;
    const double q = 2*alpha*d*(1.0-e)-M*M ;
    const double r = M*(3*alpha*d*(d-1.0+e)+M*M) ;
    const double w = pow(abs(r)+sqrt(q*q+r*r),2.0/3.0) ;

    double E=(2*r*w/(w*w+w*q+q*q)+M)/d ;
    for (int itr= 1 ; itr < 10 ; itr++)
    {
        double f = E-e*sin(E)-M ;
        double f1 = 1.-e*cos(E) ;
        double f2 = e*sin(E) ;
        double Ediff = f*f1/(f1*f1-0.5*f*f2) ;
        E -= Ediff ;
        cout << e << " " << M << " " << itr
            << " " << E << " " << Ediff << endl ;
    }
    cout << endl ;
    return E ;
} /* solveKeplerRef */

/** Compare the speed of convergence for 4 different estimators.
 * test usage:

```

```

*   tst e M
*/
int main(int argc, char *argv[])
{
    if (argc < 3 )
    {
        cout << "usage: tst e M" << endl ;
        return 1 ;
    }
    else
    {
        double e = atof(argv[1]) ;
        double M = atof(argv[2]) ;
        cout <<"standard" << endl ;
        solveKeplerRef(e,M) ;
        cout <<"Mikkola" << endl ;
        solveKeplerMikkola(e,M) ;
        cout <<"Markley" << endl ;
        solveKeplerMarkley(e,M) ;
        cout <<"Mathar" << endl ;
        solveKepler(e,M,1.e-14) ;
        return 0 ;
    }
}
#endif /* TEST */

```

* <http://www.mpia-hd.mpg.de/~mathar>

- [1] J. M. A. Danby and T. M. Burkardt, *Cel. Mech.* **31**, 95 (1983).
- [2] J. Gerlach, *SIAM Review* **36**, 272 (1994).
- [3] E. Hansen and M. Patrick, *Numer. Math.* **27**, 257 (1977).
- [4] B. Kalantari, I. Kalantari, and R. Zaare-Nahandi, *J. Comp. Appl. Math.* **80**, 209 (1997).
- [5] G. Alefeld, *Am. Math. Monthly* **88**, 530 (1981).
- [6] International Astronomical Union, *IAU SOFA Collection*, Tech. Rep.
- [7] T. Fukushima, *Cel. Mech. Dyn. Astr.* **66**, 309 (1997).
- [8] M. Palacios, *J. Comp. Appl. Math* **138**, 335 (2002).
- [9] B. A. Conway, in *24th Aerospace Sciences Meeting*, 86-0083 (AIAA, 1986).
- [10] E. D. Charles and J. B. Tatum, *Cel. Mech. Dyn. Astr.* **69**, 357 (1997).
- [11] L. Stumpf, *Cel. Mech. Dyn. Astr.* **74**, 95 (1999).
- [12] J. L. Schonfelder, *Math. Comp.* **34**, 237 (1980).
- [13] J. P. Boyd, *Appl. Num. Math.* **57**, 12 (2007).
- [14] S. Mikkola, *Cel. Mech.* **40**, 329 (1987).
- [15] A. Nijenhuis, *Cel. Mech. Dyn. Astr.* **51**, 319 (1991).
- [16] F. L. Markley, *Cel. Mech. Dyn. Astr.* **63**, 101 (1995).