

A Comparison of Regularization Methods for few-body interactions

S. Mikkola

Turku University Observatory, 21500 Piikkiö, Finland.

Abstract

In this work several methods to regularize the computation of few-body motions are discussed. These include the global regularization and the CHAIN-algorithm, which utilize the KS-transformation, and two other methods, the logarithmic Hamiltonian (logH) and time transformed leapfrog (TTL), which use a regularizing algorithm rather than coordinate transformation. All these methods are useful to advance strongly interacting subsystems in N-body simulations. Facility of use and programming are considered and accuracy tests in various systems, including some with large mass ratios, are conducted.

1 Extended phase space and time transformations

Poincare's time-transformed extended phase space Hamiltonian can be written

$$\Gamma = g(p, q)(H(p, q, t) + B),$$

where B is the momentum of time (also: binding energy), for which the initial value $B(0) = -H(0)$ guarantees that $\Gamma = 0$ along the orbit. The equations of motion are

$$\begin{aligned} p' &= -\frac{\partial \Gamma}{\partial q} = -g \frac{\partial H}{\partial q} - \frac{\partial g}{\partial q}(H + B) \\ q' &= \frac{\partial \Gamma}{\partial p} = +g \frac{\partial H}{\partial p} + \frac{\partial g}{\partial p}(H + B) \\ t' &= \frac{\partial \Gamma}{\partial B} = g. \end{aligned}$$

where $H+B = 0$ along the orbit, but *these terms cannot be dropped*.

In N-body computations the time-transformation function $g(p, q) \sim r$, where r is the shortest distance. Functions having such quality in the N-body problem are

$$g = 1/U ; \quad U = \text{potential} \quad \text{or} \quad g = 1/L ; \quad L = \text{Lagrangian.}$$

A recommendable choice is $g = 1/L$ which gives

$$\Gamma = (H + B)/L = (H - E)/L.$$

2 Regularization methods

In addition to a time transformation coordinate transformations and/or Hamiltonian manipulation and selection of a suitable algorithm can be used to regularize N-Body integrations.

KS-transformation:

$$1\text{D: } q = Q^2$$

$$2\text{D: } x + iy = (Q_1 + iQ_2)^2 \quad (\text{Levi-Civita})$$

$$3\text{D: } ix + jy + kz = U^* i U \quad (\text{KS})$$

where $U = iQ_1 + jQ_2 + kQ_3 + Q_4$ is a [quaternion](#).

These lead, with a suitable time transformation, to harmonic oscillator type behavior in near collisions.

The logarithmic Hamiltonian

$$\Lambda = \ln(T + B) - \ln(U)$$

and the leapfrog algorithm provide regularization of two-body collisions.

Time-Transformed Leapfrog (TTL)

is a non-canonical generalization of the logH-method. Here the Newtonian equations

$$\dot{x} = v, \quad \dot{v} = F(x),$$

are transformed to

$$x' = v/W, \quad t' = 1/W, \quad v' = F(x)/\Omega(x), \quad W' = \frac{\partial \Omega}{\partial x} v$$

for which a leapfrog-like algorithm can be constructed and regularization of two-body collisions is obtained if $\Omega \sim 1/r$ near collision.

3 KS TRANSFORMATION

Matrix formulation

In three dimensions the KS-transformation of coordinates \mathbf{R} and momenta \mathbf{W} may be written

$$\mathbf{R} = \widehat{\mathbf{Q}}\mathbf{Q}; \quad \mathbf{W} = \widehat{\mathbf{Q}}\mathbf{P}/(2Q^2) \quad (1)$$

Here $\widehat{\mathbf{Q}}$ is the KS-matrix (e.g. Stiefel and Scheifele (1972) p. 24)

$$\widehat{\mathbf{Q}} = \begin{pmatrix} Q_1 & -Q_2 & -Q_3 & Q_4 \\ Q_2 & Q_1 & -Q_4 & -Q_3 \\ Q_3 & Q_4 & Q_1 & Q_2 \\ Q_4 & -Q_3 & Q_2 & -Q_1 \end{pmatrix}. \quad (2)$$

The KS momenta are given by

$$\mathbf{P} = 2\widehat{\mathbf{Q}}^t\mathbf{W}. \quad (3)$$

With the time transformation

$$\frac{dt}{ds} = R = \mathbf{Q}^2 \quad (4)$$

one obtains the Hamiltonian

$$\Gamma = R(H - E) = \frac{1}{8}\mathbf{P}^2 - K^2 - E\mathbf{Q}^2, \quad (5)$$

where $H = \frac{1}{2}W^2 - K^2/R$ is the [two-body Hamiltonian](#) and $E = H(0)$ is the numerical value of the energy.

4 Simple (1d) introduction to various regularization methods

The one-dimensional problem with the Hamiltonian

$$H = p^2/2 - 1/q.$$

can be integrated with various regularization methods:

1d-KS:

Transformation $q = Q^2$, $P = p/(2Q)$, $t' = Q^2$.

With $B_0 = -H(0)$ the new regular Hamiltonian is

$$\Gamma = t'(H + B_0) = \frac{1}{8}P^2 + B_0Q^2 - 1$$

and the equation of motion

$$\begin{aligned} P' &= -\frac{\partial \Gamma}{\partial Q} = -2P_0Q, \\ Q' &= \frac{\partial \Gamma}{\partial P} = P/4, \\ t' &= \frac{\partial \Gamma}{\partial B_0} = Q^2, \end{aligned}$$

can be integrated with any reasonable numerical method.

1d-LogH:

One may also use the logarithmic Hamiltonian

$$\Lambda = \ln(p^2/2 + b_0) - \ln(1/q),$$

with $b_0 = -H(0)$, and try to solve approximately the equations of motion

$$\begin{aligned}q' &= p/(p^2/2 + b_0) \\p' &= -1/q \\t' &= 1/(p^2/2 + b_0),\end{aligned}\tag{6}$$

using the leapfrog. Defining the two 'subroutines':

$$\begin{aligned}\mathbf{X}(\mathbf{s}) : & \tag{7} \\ \delta t &= s/(p^2/2 + b_0) \\ q &\rightarrow q + \delta t p \\ t &\rightarrow t + \delta t\end{aligned}$$

and

$$\begin{aligned}\mathbf{V}(\mathbf{s}) : & \tag{8} \\ p &\rightarrow p - s/q\end{aligned}$$

one can symbolize the leapfrog by

$$\mathbf{X}(\mathbf{h}/2)\mathbf{V}(\mathbf{h})\mathbf{X}(\mathbf{h}/2),$$

i.e. half step $\mathbf{X}(\mathbf{h}/2)$, then full step $\mathbf{V}(\mathbf{h})$, followed by a half step $\mathbf{X}(\mathbf{h}/2)$.

Surprisingly, this happens to produce the [correct trajectory](#), having only a phase error.

The toy code: 1dLogH.f

```
      implicit real*8 (a-h,o-z)
      q=1 ! initialize
      p=0
      b=-(p*p/2-1/q)
      h=0.1d0
1     continue
      call qm(h/2,p,q,b,t)
      call pm(h ,p,q)
      call qm(h/2,p,q,b,t)

c     diagno
      err=p*p/2-1/q+b ! Energy error
      erru=q*err      ! This should remain constant (h
      write(6,*)t,q,p*q,err,erru
      if(t.lt.100.)goto 1
      end

      subroutine qm(s,p,q,b,t)
      implicit real*8 (a-h,o-z)
      dt=s/(p*p/2+b)
      t=t+dt
      q=q+dt*p
      return
      end

      subroutine pm(s,p,q)
      implicit real*8 (a-h,o-z)
      p=p-s/q
      return
      end
```

1d-TTL:

The normal leapfrog method is often used for the Newtonian equations of motion, which in our simple case read

$$\dot{p} = -1/q^2, \quad \dot{q} = p.$$

The collision ($q \rightarrow 0$) is, however, singular and the basic leapfrog does not work.

The idea of TTL is to introduce a time transformation in such a way that a modified leapfrog can be constructed. Write

$$p' = \dot{p}/\Omega(q), \quad q' = \dot{q}/W, \quad t' = 1/W$$

where we consider W to be a 'velocity-like' variable and $\Omega(q)$ is some suitable function of q . Clearly we must have, [in numerical value](#), $W = \Omega(q)$, but to make the construction of a leapfrog possible we obtain the value of W from the differential equation

$$\dot{W} = \dot{\Omega}(q) = \frac{\partial \Omega}{\partial q} p, \quad \text{or} \quad W' = \dot{W}/\Omega(q).$$

In the 1d-case a reasonable choice is $\Omega = 1/q$, so that

$$q' = p/W, \quad t' = 1/W, \quad p' = -1/q, \quad W' = -p/q.$$

The equations

$$q' = p/W, \quad t' = 1/W$$

can be solved for constant p and W and

$$p' = -1/q, \quad W' = -p/q$$

are solvable for constant q , thus a [leapfrog can be constructed as a composite of these two pairs](#).

The 'subroutines' are

$$\begin{aligned} \mathbf{X}(\mathbf{s}) : & & (9) \\ \delta t &= s/W \\ q &\rightarrow q + \delta t p \\ t &\rightarrow t + \delta t \end{aligned}$$

and

$$\begin{aligned} \mathbf{V}(\mathbf{s}) : & & (10) \\ \delta p &= -s/q \\ p_a &= p + \delta p/2 \\ p &\rightarrow p + \delta p \\ W &\rightarrow W - sp_a/q \end{aligned}$$

giving the leapfrog

$$\mathbf{X}(\mathbf{h}/2)\mathbf{V}(\mathbf{h})\mathbf{X}(\mathbf{h}/2),$$

The toy code: 1dTTL:

```
implicit real*8 (a-h,o-z)
q=1
p=0
b=p*p/2-1/q ! =Energy
W=1/q      ! initial value
h=0.1d0
t=0

1  continue
   call qm(h/2,p,q,w,t)
   call pm(h ,p,q,w)
   call qm(h/2,p,q,w,t)

c   diagno
   err=p*p/2-1/q-b ! Energy error
   erru=q*err      !
   write(6,123)t,q,p*q,err,erru
123 format(1x,f10.4,2f10.4,1p,3g10.2)
   if(t.lt.100.)goto 1
   end

subroutine qm(s,p,q,w,t)
implicit real*8 (a-h,o-z)
dt=s/w
t=t+dt
q=q+dt*p
return
end

subroutine pm(s,p,q,w)
implicit real*8 (a-h,o-z)
dp=-s/q
pa=p+dp/2
p=p+dp
w=w-s*pa/q
return
end
```

Figure 1: ENERGY ERRORS IN 1dTTL and 1dLogH.

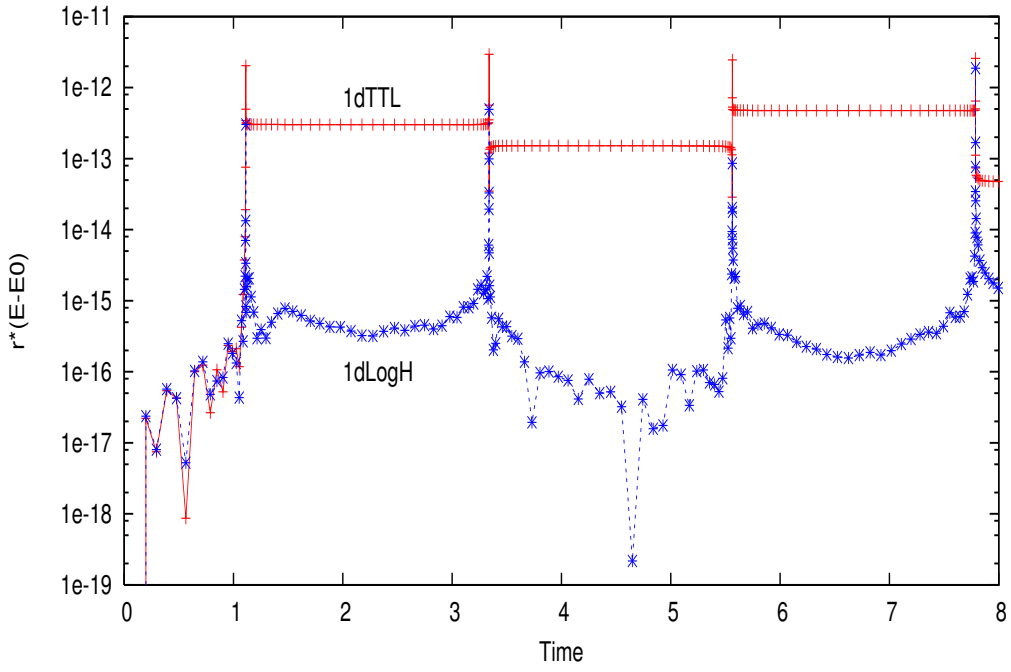
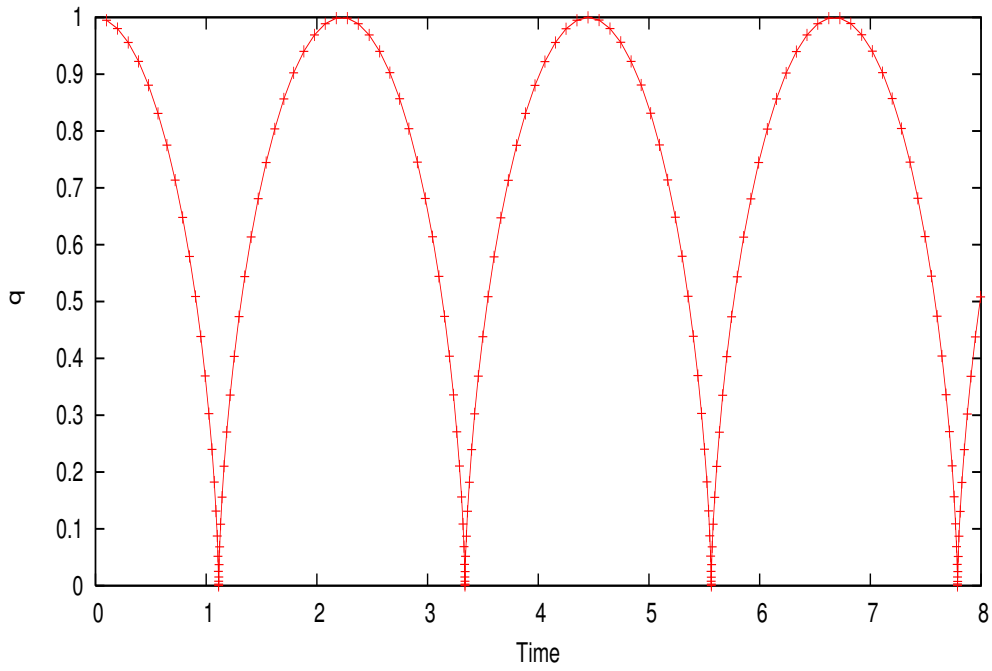


Figure 2: Motion of q



Some conclusions:

- Mathematically the logH and TTL methods are equivalent in this simple case [increment of w is equivalent to that of the kinetic energy in logH].
- The difference in precision is due to numerical effects: there is more round-off in updating the W -variable. One notes that in collision $W \rightarrow \infty$ and then it return back to 'normal'. All this with many increments that can be individually very large.
- In more general cases one may recommend the logH-method in case it is possible to use

$$\Omega = \text{potential.}$$

End of simplicity.....

N-Body system in one dimension

One-dimensional CHAIN

The Hamiltonian of the problem is given by

$$H = \sum_i \frac{p_i^2}{2m_i} - \sum_{i < j} \frac{m_i m_j}{|x_i - x_j|}.$$

New coordinates $X_k = x_{k+1} - x_k$ and the generating function

$$S = \sum_{k=1}^{N-1} W_k X_k = \sum_{k=1}^{N-1} W_k (x_{k+1} - x_k).$$

together with the canonical transformation

$$X_k = Q_k^2; \quad W_k = \frac{P_k}{2Q_k},$$

and a suitable time transformation

$$g = 1/L = 1/(T + U),$$

gives the new regularized Hamiltonian

$$\Gamma = (T - U - E_0)/(T + U),$$

where

$$T = \sum_{k=1}^{N-1} \frac{1}{8} \left(\frac{1}{m_k} + \frac{1}{m_{k+1}} \right) \frac{P_k^2}{Q_k^2} - \sum_{k=2}^{N-1} \frac{P_{k-1} P_k}{4Q_{k-1} Q_k m_k}$$

$$U = \sum_{k=1}^{N-1} \frac{m_k m_{k+1}}{Q_k^2} + \sum_{i < j+1} \frac{m_i m_j}{\sum_{k=i}^{j-1} Q_k^2}.$$

The equations of motion may be written

$$H = T - U; \quad g = 1/(T + U); \quad \Gamma = g(H - E),$$

$$\Gamma_T = (1 - \Gamma)g; \quad \Gamma_U = -(1 + \Gamma)g,$$

$$\Gamma_{P_k} = \Gamma_T T_{P_k},$$

$$\Gamma_{Q_k} = \Gamma_T T_{Q_k} + \Gamma_U U_{Q_k},$$

$$P_k' = -\Gamma_{Q_k},$$

$$Q_k' = \Gamma_{P_k}.$$

LogH and TTL (1, 2 and 3 dimensions....)

With

$$T = \sum_i \frac{p_i^2}{2m_i}, \quad U = \sum_{i<j} \frac{m_i m_j}{|x_i - x_j|} \quad (11)$$

and the binding energy $B = U - T$ one may define the logarithmic Hamiltonian

$$\Lambda = \ln(T + B) - \ln(U),$$

which gives the equations of motion

$$p'_k = \frac{1}{U} \frac{\partial U}{\partial x_k}, \quad x'_k = \frac{1}{T + B} \frac{\partial T}{\partial p_k}. \quad (12)$$

and **since p' only depends on x and x' only on p , a leapfrog is possible.** One expects this to give good approximations for close approaches because it is asymptotically exact for two-body collisions.

The TTL formulation is now

$$t' = 1/W, \quad x'_k = \frac{1}{W} \frac{\partial T}{\partial p_k} \quad (13)$$

$$p'_k = \frac{1}{\Omega} \frac{\partial U}{\partial x_k}, \quad W' = \sum_k \frac{\partial \Omega}{\partial x_k} \frac{\partial T}{\partial p_k} / \Omega \quad (14)$$

and the leapfrog for this system can be constructed.

Note that the soft potential

$$U = \sum_{i<j} m_i m_j / \sqrt{|x_i - x_j|^2 + \epsilon^2}, \quad (15)$$

is ‘regularized’ by these algorithms.

The toy code 1dNBTTTL.f

```
1dNBTTTL.f
implicit real*8 (a-h,o-z)
real*8 q(6),p(6)
common/softening/eps,U
read(5,*)NB,h,tmx,eps
read(5,*)(q(k),k=1,nb)
read(5,*)(p(k),k=1,nb)
call energy(nb,p,q,b,w)
goto 2
1  continue
   ts=t
   do kie=1,10 000
   call qm(nb,h/2,p,q,w,t) ! LEAPFROG
   call pm(nb,h ,p,q,w) !
   call qm(nb,h/2,p,q,w,t) !
   if(t.gt.ts+.01)goto 2
   end do
c   diagno
2   continue
   call energy(nb,p,q,bt,U)
   err=-bt+b
   erru=err/U
123 write(6,123)t,W/U-1,erru,(q(k),k=1,nb)
    format(1x,f10.3,1p,8g12.4)
    if(t.lt.tmx)goto 1
    end
subroutine qm(nb,s,p,q,w,t)
implicit real*8 (a-h,o-z)
real*8 q(6),p(6)
dt=s/w
t=t+dt
do k=1,nb
q(k)=q(k)+dt*p(k)
end do
return
end
subroutine pm(nb,s,p,q,w)
implicit real*8 (a-h,o-z)
real*8 q(6),p(6),A(6)
common/softening/eps,U
do k=1,nb
A(k)=0
end do
U=0
do i=1,nb-1
do j=i+1,nb
xij=q(i)-q(j)
rij=sqrt(xij**2+eps**2)
```

```

aij=-xij/rij**3
U=U+1/rij ! pot
a(i)=a(i)+aij ! acc_i
a(j)=a(j)-aij ! acc_j
end do
end do
dlt=s/U
do k=1,nb
pa=p(k)
p(k)=p(k)+dlt*a(k)
pa=(pa+p(k))/2
w=w+dlt*a(k)*pa
end do
return
end
subroutine energy(nb,p,q,b,w)
implicit real*8 (a-h,o-z)
real*8 q(6),p(6)
common/softening/eps,U
T=0
U=0
do k=1,nb
T=T+p(k)**2/2
end do
do i=1,nb-1
do j=i+1,nb
U=U+1/sqrt((q(i)-q(j))**2+eps**2)
end do
end do
b=U-T
w=U
return
end

```


Figure 3: Three-body motion in 1d (hard potential).

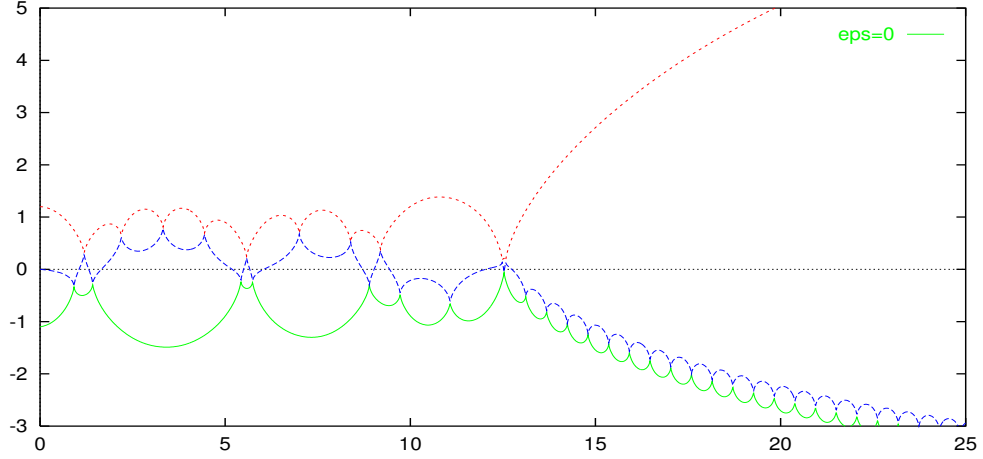


Figure 4: Three-body motion in 1d (soft potential).

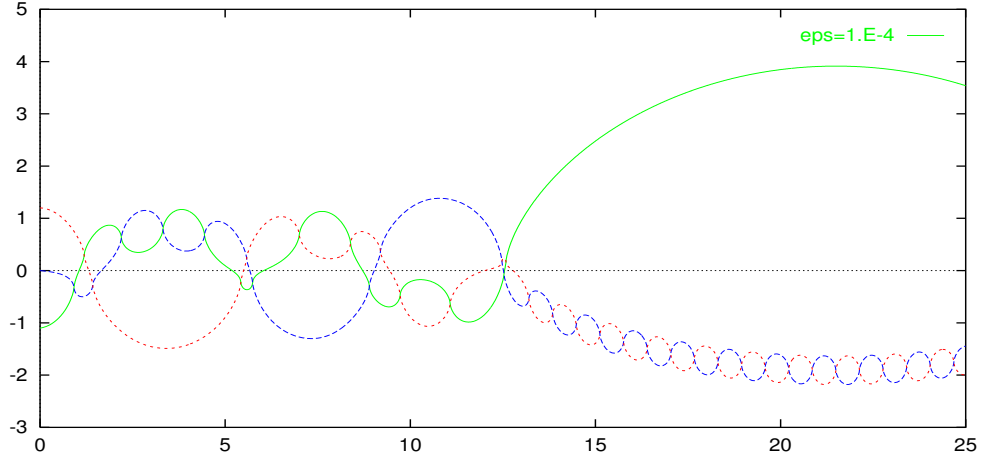
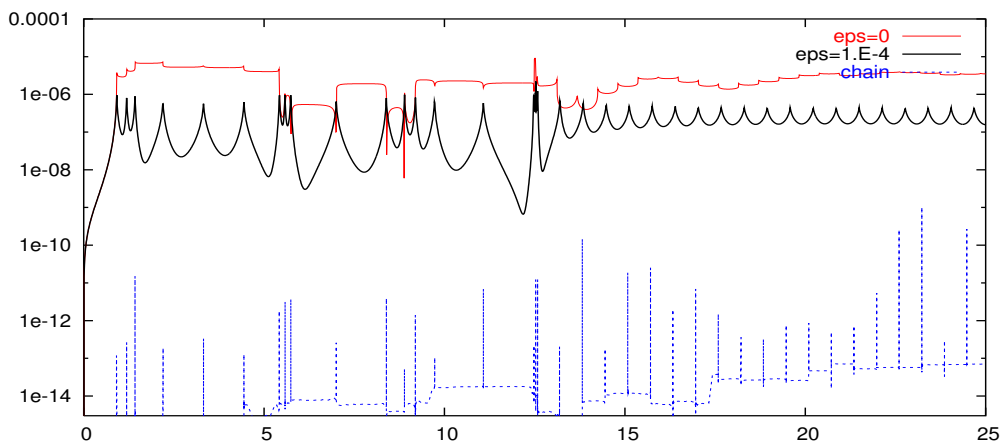


Figure 5: Energy errors in hard and soft potential calculations



REGULARIZATION IN THREE DIMENSIONS

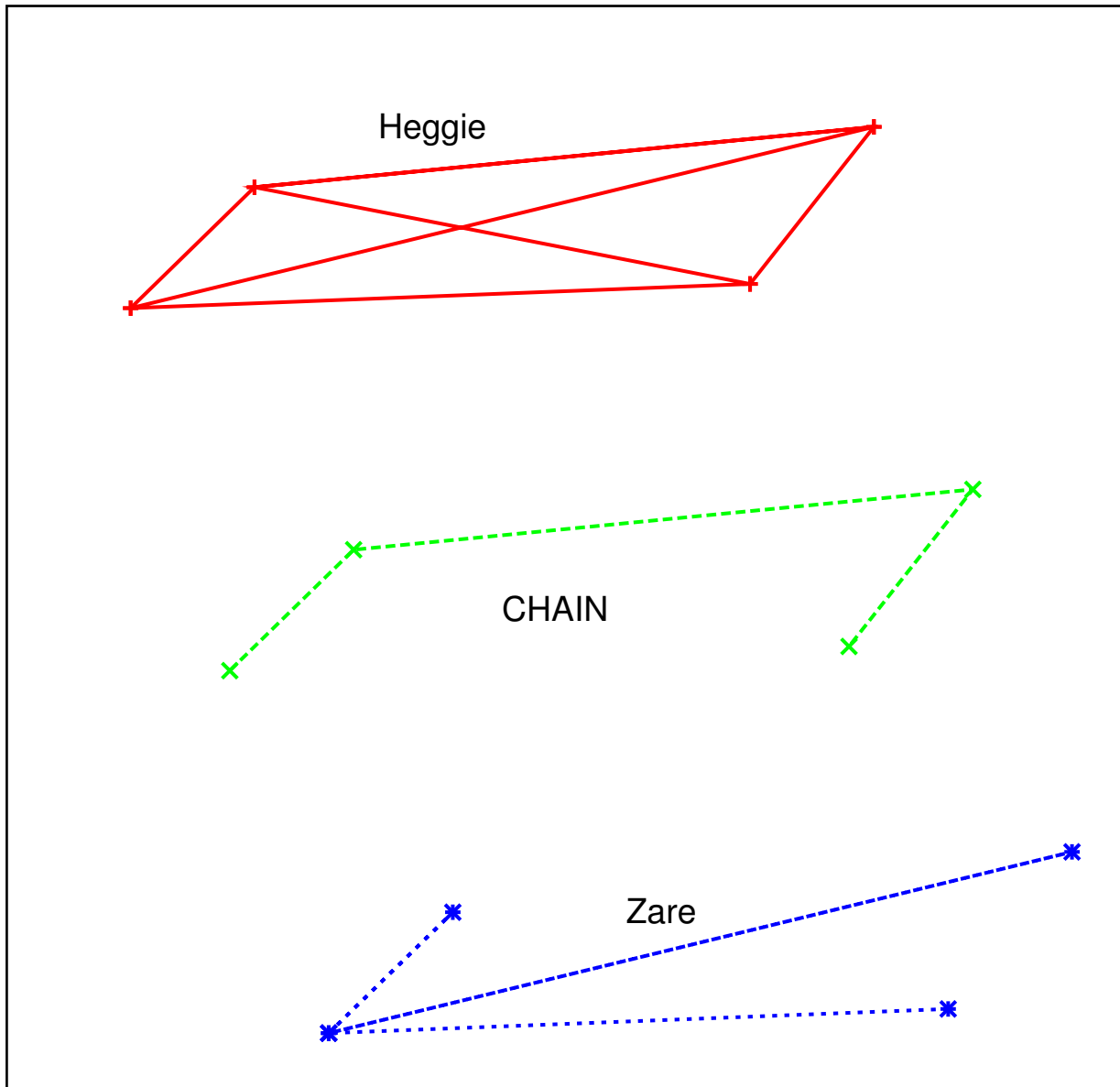


Figure 6: KS regularized interactions in the global method of Heggie, CHAIN-method and Zare's cartwheel method.

5 HEGGIE'S GLOBAL METHOD

Starting with the Hamiltonian

$$H = \sum_{i=1}^N \frac{1}{2m_i} w_i^2 - \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{m_i m_j}{r_{ij}} \quad (16)$$

Heggie (1974) introduces new independent variables $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. The generating function

$$S = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \mathbf{w}_{ij} \cdot (\mathbf{r}_j - \mathbf{r}_i), \quad (17)$$

defines the moments \mathbf{w}_{ij} . The resulting expression for the Hamiltonian can be simplified by renaming the variables $(\mathbf{w}_{ij}, \mathbf{r}_{ij})$ to $(\mathbf{p}_k, \mathbf{q}_k)$, (in a loop 'do i=1,N-1; do j=i+1,N; k=k+1 ...). The kinetic energy and potential can then be written

$$T = \sum_{\mu\nu} T_{\mu\nu} \mathbf{P}_\mu^t \widehat{\mathbf{Q}}_\mu^t \widehat{\mathbf{Q}}_\nu \mathbf{P}_\nu / (4Q_\mu^2 Q_\nu^2), \quad U = \sum_k M_k / Q_k^2, \quad (18)$$

and with $\Gamma = (T - U - E)/(T + U)$ a concise algorithm can be formulated:

(i) Calculate initial values and auxiliary quantities

$$\begin{aligned} k &= (i-1)N - i(i+1)/2 + j \\ \mathbf{q}_k &= \mathbf{r}_i - \mathbf{r}_j \\ \mathbf{p}_k &= (m_i \dot{\mathbf{r}}_i - m_j \dot{\mathbf{r}}_j) / N \\ M_k &= m_i m_j \\ a_{ik} &= 1, \quad a_{jk} = -1 \quad [\text{other components} = 0] \\ T_{\mu\nu} &= \frac{1}{2} \sum_{n=1}^N a_{n\mu} a_{n\nu} / m_n \end{aligned} \quad (19)$$

(ii) With the KS variables \mathbf{P}_k and \mathbf{Q}_k compute

$$\begin{aligned}
\mathbf{p}_k &= \widehat{\mathbf{Q}}_k \mathbf{P}_k / (2Q_k^2) \\
\mathbf{A}_k &= \sum_{\mu=1}^K T_{k\mu} \mathbf{P}_\mu \\
d_k &= A_k \cdot \mathbf{p}_k \\
T &= \sum_{k=1}^K d_k, \quad U = \sum_{k=1}^K M_k / Q_k^2 \\
T_{\mathbf{P}_k} &= \widehat{\mathbf{Q}}_k^t \mathbf{A}_k / Q_k^2 \\
T_{\mathbf{Q}_k} &= (\widehat{\mathbf{P}}_k^t \mathbf{A}_k^* - 4d_k \mathbf{Q}_k) / Q_k^2 \\
\mathbf{A}_k^* &= (A_1, A_2, A_3, -A_4)_k^t, \\
U_{\mathbf{Q}_k} &= -2M_k \mathbf{Q}_k / Q_k^4 \\
L &= T + U; \quad H = T - U; \quad \Gamma = (H - E) / L \\
\Gamma_T &= (1 - \Gamma) / L; \quad \Gamma_U = -(1 + \Gamma) / L. \\
\mathbf{Q}'_k &= \Gamma_T T_{\mathbf{P}_k} \\
\mathbf{P}'_k &= -\Gamma_T T_{\mathbf{Q}_k} - \Gamma_U U_{\mathbf{Q}_k} \\
t' &= 1 / L.
\end{aligned} \tag{20}$$

(iii) Transformation from the KS variables to the physical ones:

$$\begin{aligned}
\mathbf{r}_{ij} &= \mathbf{q}_k = \widehat{\mathbf{Q}}_k \mathbf{Q}_k \\
\mathbf{w}_{ij} &= \mathbf{p}_k = \widehat{\mathbf{Q}}_k \mathbf{P}_k / (2Q_k^2) \\
\mathbf{r}_i &= \left(\sum_{j=i+1}^N m_j \mathbf{r}_{ij} - \sum_{j=1}^{i-1} m_j \mathbf{r}_{ji} \right) / \sum_{n=1}^N m_n \\
\dot{\mathbf{r}}_i &= \left(\sum_{j=i+1}^N \mathbf{w}_{ij} - \sum_{j=1}^{i-1} \mathbf{w}_{ji} \right) / m_i.
\end{aligned} \tag{22}$$

6 THE CHAIN METHOD

Suppose a chain of vectors connecting N bodies has been selected and bodies relabeled 1, 2, ..., N along the chain. The generating function

$$S = \sum_{i=1}^{N-1} \mathbf{W}_k \cdot (\mathbf{q}_{k+1} - \mathbf{q}_k) \quad (23)$$

defines a transformation to new chain-coordinates $\mathbf{R}_k = \mathbf{q}_{k+1} - \mathbf{q}_k$ and momenta \mathbf{W}_k . The Hamiltonian takes the form

$$H = \sum_{k=1}^{N-1} \frac{1}{2} \left(\frac{1}{m_k} + \frac{1}{m_{k+1}} \right) \mathbf{W}_k^2 - \sum_{k=2}^N \frac{1}{m_k} \mathbf{W}_{k-1} \cdot \mathbf{W}_k - \sum_{k=1}^{N-1} \frac{m_k m_{k+1}}{R_k} - \sum_{1 \leq i \leq j-2} \frac{m_i m_j}{R_{ij}}. \quad (24)$$

The KS transformations $\mathbf{R}_k = \widehat{\mathbf{Q}}_k \mathbf{Q}_k$, $\mathbf{W}_k = \widehat{\mathbf{Q}}_k \mathbf{P}_k / (2Q_k^2)$ and the 'standard' Hamiltonian

$$\Gamma = g(H - E) = (T - U - E)/(T + U)$$

gives

$$\mathbf{P}_k' = -\partial\Gamma/\partial\mathbf{Q}_k, \quad \mathbf{Q}_k' = \partial\Gamma/\partial\mathbf{P}_k.$$

7 CHAIN SELECTION

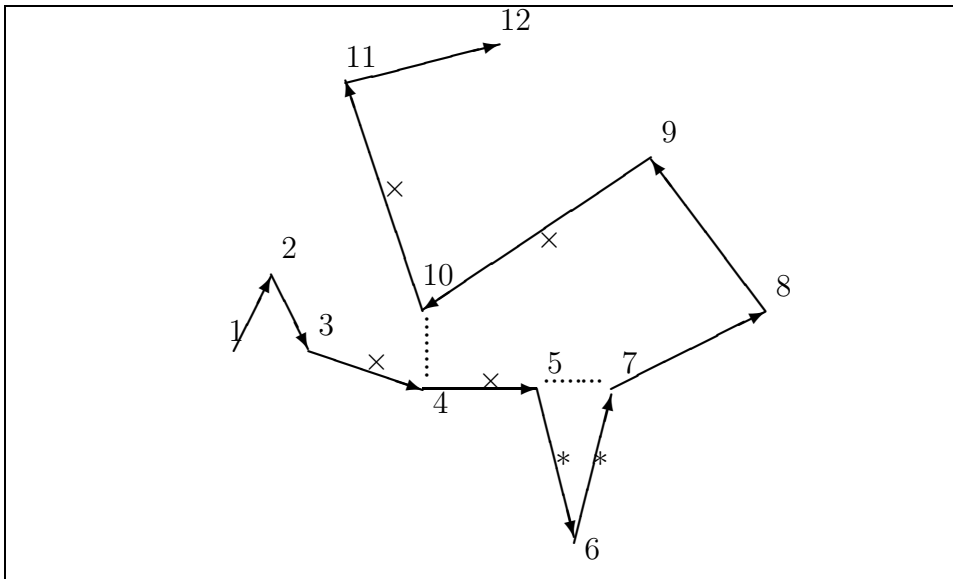


Figure 7: Illustration of the chain and the checking of switching conditions. Distances like $R_{5,7}$ are compared with the smaller of the two distances $R_{5,6}$ and $R_{6,7}$ (marked by *). Interparticle distances like $R_{4,10}$ are compared with the smallest of those in contact with the considered distance (marked by \times).

Chain selection begins by finding the shortest interparticle vector. This is taken as the first part of the chain under construction. One proceeds by searching for the particle which is closest to one or the other end of the presently known part of the chain. Once identified, this particle is added to the chain (to the end which is closer). This operation is repeated until all particles are included in the chain. There is also a check for need of a new chain after every integration step.

8 LogH and TTL in small clusters

8.1 logH:

With the usual notation

$$T = \sum_i \frac{p_i^2}{2m_i} \quad (25)$$

$$U = \sum_{i<j} \frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|} + U_{extr}(\mathbf{r}, t) \quad (26)$$

$$(27)$$

and the logarithmic Hamiltonian

$$\Lambda = \ln(T + B) - \ln(U),$$

the basic leapfrog subroutines $\mathbf{X}(s)$ and $\mathbf{V}(s)$ are

$$\mathbf{X}(\mathbf{s}) : \quad (28)$$

$$\delta t = s/(T + B)$$

$$\mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t \mathbf{p}_k$$

$$t \rightarrow t + \delta t$$

and

$$\mathbf{V}(\mathbf{s}) : \quad (29)$$

$$\mathbf{p} \rightarrow \mathbf{p}_k + \frac{s}{U} \frac{\partial U}{\partial \mathbf{r}_k}$$

$$B \rightarrow B + \frac{s}{U} \frac{\partial U}{\partial t}$$

and the leapfrog reads

$$\mathbf{X}(\mathbf{h}/2)\mathbf{V}(\mathbf{h})\mathbf{X}(\mathbf{h}/2),$$

Leapfrog alone is not satisfactory and it must to be used just as a basic algorithm to provide results for a **Gragg-Bulirsch-Stoer extrapolation method**.

A very simple two-body toy code (LogH-method)

```

      implicit real*8 (a-h,m,o-z)
      real*8 x(3),v(3)
      M=1.d0
      read(5,*)h,Tmx,x,v
      t=0
      r=sqrt(x(1)**2+x(2)**2+x(3)**2)
      vv=v(1)**2+v(2)**2+v(3)**2
      B=M/r-vv/2
c
c      Integration of the two-body motion
1      continue

      dt=h/(v(1)**2+v(2)**2+v(3)**2+2*B)
      do k=1,3
      x(k)=x(k)+dt*v(k) ! x-motion
      end do
      t=t+dt
c

      dtc=h/(x(1)**2+x(2)**2+x(3)**2)
      do k=1,3
      v(k)=v(k)-dtc*x(k) ! v-motion
      end do

      dt=h/(v(1)**2+v(2)**2+v(3)**2+2*B)
      do k=1,3
      x(k)=x(k)+dt*v(k) ! x-motion
      end do
      t=t+dt
      t=t-h**3/12/(M)**2 ! +O(h^5) time correction (only for small h)

c      diagnostics
      r=sqrt(x(1)**2+x(2)**2+x(3)**2)
      vv=v(1)**2+v(2)**2+v(3)**2
      Bt=M/r-vv/2
      write(6,*)t,x,Bt-B
      if(t.lt.Tmx)goto 1
      end
```


8.2 TTL:

The TTL equations

$$t' = 1/W \quad (30)$$

$$\mathbf{r}'_k = \frac{1}{W} \frac{\partial T}{\partial \mathbf{p}_k} \quad (31)$$

$$\mathbf{v}'_k = \frac{1}{\Omega} \mathbf{A}_k \quad (32)$$

$$W' = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \mathbf{v}_k / \Omega \quad (33)$$

where \mathbf{v} , \mathbf{A} are the velocity and acceleration, can be used to construct the leapfrog $\mathbf{X}(\mathbf{h}/2)\mathbf{V}(\mathbf{h})\mathbf{X}(\mathbf{h}/2)$, where

$$\mathbf{X}(\mathbf{s}) : \quad (34)$$

$$\delta t = s/W$$

$$\mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t \mathbf{p}_k$$

$$t \rightarrow t + \delta t$$

$$\mathbf{V}(\mathbf{s}) : \quad (35)$$

$$\delta \mathbf{v}_k = \frac{s}{\Omega} \mathbf{A}_k$$

$$W \rightarrow W + \frac{s}{\Omega} \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot (\mathbf{v}_k + \frac{1}{2} \delta \mathbf{v}_k)$$

$$\mathbf{v}_k \rightarrow \mathbf{v}_k + \delta \mathbf{v}_k$$

For Ω **one may use any suitable function**, usually

$$\Omega = \sum_{i < j} \frac{\Omega_{ij}}{r_{ij}},$$

where

$$\Omega_{ij} = 1, \quad \text{or} \quad \Omega_{ij} = \mathbf{m}_i \mathbf{m}_j,$$

the latter choice being recommendable iff the masses are comparable. Finally: **use extrapolation method.**

9 ON PROGRAMMING

-Facility of programming (in author's opinion)

LogH < TTL < Heggie < CHAIN

where $A < B$ means 'A is simpler than B'.

- The **algorithmic regularization methods LogH and TTL** are fairly simple to program: differences compared with the most straightforward programming are minor in the simplest implementations.
- Comparing the methods that use KS-transformation suggests that in fact the Heggie's global method is simpler than CHAIN. This is because all interactions are treated in the same way.
- The CHAIN has some complicated bookkeeping which make it harder to implement, however, it may be worth of it!

10 NUMERICAL

Figure 8: dE/L in a 3-body system; comparable masses

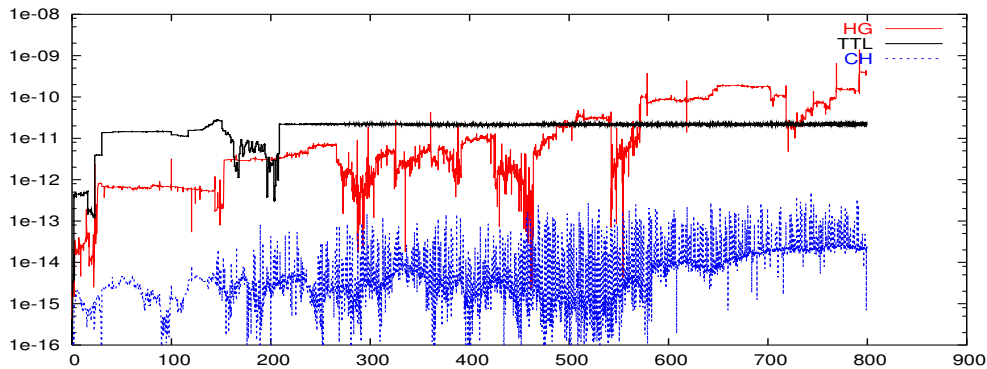
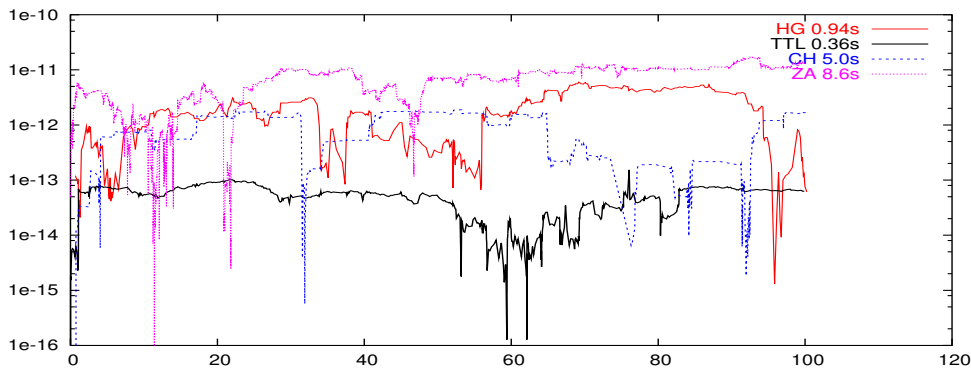


Figure 9: dE/L in a 7-body system; $\max/\min=10^6$



11 CONCLUSION

- For **comparable masses** the **CHAIN** algorithm is clearly preferable.
- If extreme **many-body encounters are excluded**, then **LogH** may compete with CHAIN. It is also enormously simpler to code!
- For **very large mass ratios** it becomes evident that **TTL** is more accurate and faster.

Literature

1. Aarseth, S. J. and Zare, K.: 1974, “A Regularization of the Three-Body Problem”, *Cel. Mech.*, **10**, 185–205.
2. Alexander, M. E.: 1986, “Simulations of binary-single star and binary-binary scattering”, *J. Comp. Phys.*, **64**, 195–219.
3. Bulirsch, R. and Stoer, J.: 1966, “Numerical Treatment of Differential Equations by Extrapolation Methods”, *Num. Math.*, **8**, 1–13.
4. Heggie, D. C.: 1974, “A Global Regularisation of the Gravitational N-Body Problem”, *Cel. Mech.*, **10**, 217–241.
5. Preto, M. and Tremaine, S.: 1999, ‘A class of symplectic integrators with adaptive timestep for separable Hamiltonian systems’, *Astron J.* **118**, 2532–2541.
6. Mikkola, S.: 1985, “A Practical and Regular Formulation of the N-Body Equations”, *Mon. Not. R. Astr. Soc.*, **215**, 171–177.
7. Mikkola, S. and Aarseth, S. J.: 1993, “An Implementation of N-body Chain Regularization”, *Celest. Mech. Dyn. Astron.* **57**, 439
8. Mikkola, S. and Aarseth, S. J.: 1996, “A Slow-Down Treatment of Close Binaries”, *Celest. Mech. Dyn. Astron.*
9. S. Mikkola & K. Tanikawa, (1999) *Algorithmic regularization of the few-body problem*, Monthly Notices of the Royal Astronomical Society, **310**, 745-749.
10. S. Mikkola & K. Tanikawa, (1999) *Explicit symplectic algorithms for time-transformed Hamiltonians*. *Celestial Mechanics and Dynamical Astronomy*, **74**, 287-295.
11. Mikkola, S. and Aarseth, S. (2002) *A Time Transformed Leapfrog Scheme: Integration of few-body systems with large mass ratios.*, *Celestial Mechanics and Dynamical Astronomy*, **84**: 343-354.
12. Stiefel, E. L. and Scheifele, G.: 1971, *Linear and Regular Celestial Mechanics*, Springer, Berlin.
13. Szebehely, V. and Zare, K: 1975, “Time transformations in the extended phase-space”, *Cel. Mech.*, **11**, 469.
14. Zare, K.: 1974, “A Regularization of Multiple Encounters in Gravitational N-body Problems” *Cel. Mech.*, **10**, 207–215.