*Few-Body Problem: Theory and Computer Simulations*
*Annales Universitatis Turkuensis, Series 1A, Vol. 358, 2006*
*C. Flynn, ed.*

# Few-Body Regularizations

Seppo Mikkola

*Tuorla Observatory, University of Turku, Finland*

**Abstract.** The state-of-the-art in regularization of the few-body problem is concisely reviewed. For systems with comparable masses, the methods utilizing the Kustaanheimo-Stiefel (KS) transformation are generally the most practical. These include the global regularization of all interactions and the chain-method. Alternatively, one may use algorithmic regularizations, such as the logarithmic Hamiltonian leapfrog or the time transformed leapfrog. All these methods are useful for integration of strongly interacting few-body systems, but only the time transformed leapfrog can be used for arbitrary mass ratios.

## 1. Introduction

The well known $1/r^2$ singularity in the gravitational interaction is a major obstacle in N-body simulations. For strongly interacting few-body systems various regularization methods have been developed. The most successful of these is the Kustaanheimo-Stiefel (KS) transformation (Kustaanheimo & Stiefel 1965). This method was generalized for strongly interacting three-body systems by Aarseth and Zare (1974) (AZ) and further Heggie (1974) invented the global regularization method for N-body systems. This method was somewhat simplified by Mikkola (1985) using a more economic notation. Another N-body method, in which one body is a central reference body, was suggested by Zare (1974).

The Heggie method regularizes all the interactions at once but at the price of introducing a large number of extra degrees of freedom.

Later Mikkola and Aarseth (1993) generalized the AZ-method for N-bodies (CHAIN-method). This method uses a chain of interparticle vectors that are transformed using KS, thus regularizing the interactions. The advantage introduced was that there are no longer extra degrees of freedom. The changing structure of the system is taken into account by appropriately switching the chain.

Another way of regularizing the computation of motions is to use algorithmic regularization (Mikkola & Tanikawa 1999ab, Preto & Tremaine 1999, Mikkola & Aarseth 2002). These methods use a suitable time transformation and algorithm, but no coordinate transformations, except possibly linear ones.

## 2. Extended phase space and time transformations

A basic ingredient in regularization methods is the Poincare's transformation to extended phase space. The new Hamiltonian reads

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

where $B$ is the momentum of time with the initial value $B(0) = -H(0)$. The equations of motion are

$$p' = -\frac{\partial \Gamma}{\partial q};\ q' = \frac{\partial \Gamma}{\partial p};\ t' = \frac{\partial \Gamma}{\partial B} = g. \tag{1}$$

Szebehely & Zare (1975), and Alexander, (1986) recommend $g = 1/L$, where $L = T + U =$ Lagrangian. Thus

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

is a useful form for a time-transformed Hamiltonian.

## 3. Multiparticle regularizations by KS transformation

The KS-transformation of coordinates $\mathbf{r}$ and momenta $\mathbf{p}$ may be written

$$\mathbf{r} = \widehat{\mathbf{Q}}\mathbf{Q};\ \ \mathbf{p} = \widehat{\mathbf{Q}}\mathbf{P}/(2Q^2) \tag{2}$$

Here $\widehat{\mathbf{Q}}$ is the KS-matrix (e.g. Stiefel and Scheifele (1971) p. 24)

$$\widehat{\mathbf{Q}} = \begin{pmatrix} Q_1 & -Q_2 & -Q_3 & Q_4 \\ Q_2 & Q_1 & -Q_4 & -Q_3 \\ Q_3 & Q_4 & Q_1 & Q_2 \\ Q_4 & -Q_3 & Q_2 & -Q_1 \end{pmatrix}. \tag{3}$$

In the two-body case one may use

$$\frac{dt}{ds} = r = \mathbf{Q}^2 \tag{4}$$

to obtain the new Hamiltonian

$$\Gamma = r(H + B) = \frac{1}{8}\mathbf{P}^2 - M + B\mathbf{Q}^2, \tag{5}$$

where $H = \frac{1}{2}p^2 - M/r$ and $B = -H(0)$ is the numerical value of the binding energy.

When applied to more complicated systems, the final result is no longer an harmonic oscillator but close approaches are regularized anyway.

Let a multiparticle Hamiltonian in the centre-of-mass system be

$$H = \sum_{\nu} \mathbf{p}_\nu^2/(2m_\nu) - \sum_{i<j} m_i m_j / r_{ij}$$

and let us introduce new coordinates

$$\mathbf{X}_k = \mathbf{r}_{i_k} - \mathbf{r}_{j_k},$$

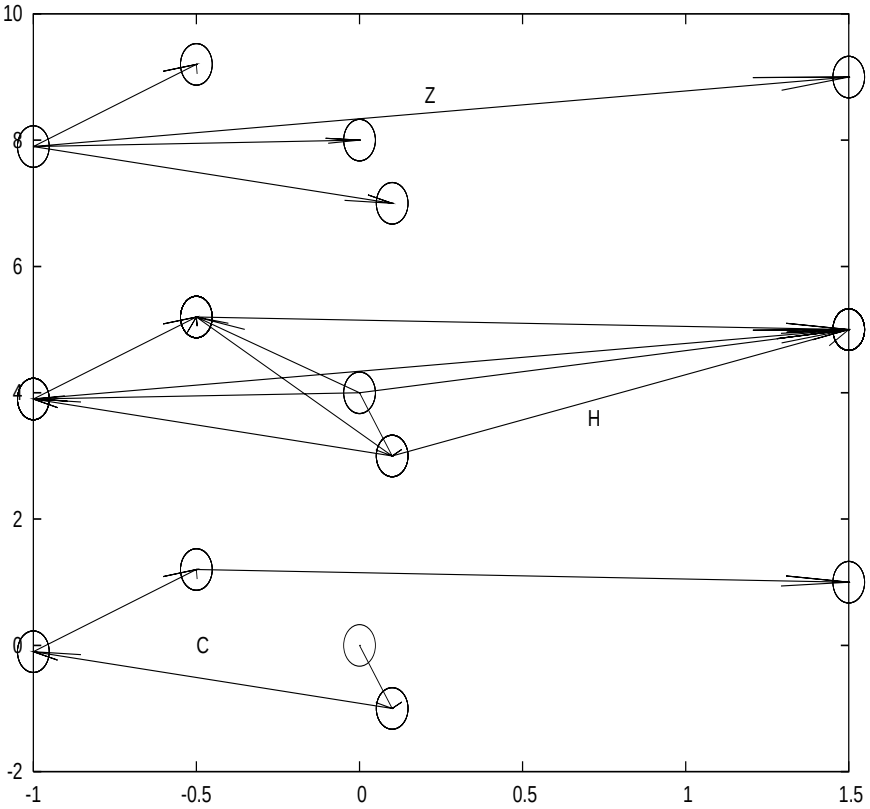


Figure 1. Regularized interactions (schematically) in the Zare method (Z), the global method of Heggie (H) and the chain method (C).

then we can use the generating function

$$S = \sum_k \mathbf{W}_k \cdot \mathbf{X}_k = \sum_k \mathbf{W}_k \cdot (\mathbf{r}_{i_k} - \mathbf{r}_{j_k}). \tag{6}$$

In terms of the new momenta $\mathbf{W}$ the old ones are $\mathbf{p}_\nu = \frac{\partial S}{\partial \mathbf{r}_\nu} = \sum_k \mathbf{W}_k \cdot (\delta_{\nu i_k} - \delta_{\nu j_k})$, where the $\delta$'s are the Kronecker symbols. Thus the new Hamiltonian reads

$$H = \frac{1}{2} \sum_{\alpha\beta} T_{\alpha\beta} \mathbf{W}_\alpha \cdot \mathbf{W}_\beta - \sum_k \frac{m_{i_k} m_{j_k}}{|\mathbf{X}_k|} - \sum_{i<j,\ (i,j)\notin\{i_k,j_k\}} \frac{m_i m_j}{r_{ij}}, \tag{7}$$

where

$$T_{\alpha\beta} = \sum_\nu \frac{1}{m_\nu} (\delta_{\nu i_\alpha} - \delta_{\nu j_\alpha})(\delta_{\nu i_\beta} - \delta_{\nu j_\beta}),$$

and the latter potential term

$$\sum_{i<j,\ (i,j)\notin\{i_k,j_k\}} \frac{m_i m_j}{r_{ij}}$$

contains all the distances $r_{ij} = r_{ij}(X_1, X_2...)$ that are not included among the $\mathbf{X}_k$. After application of the KS transformation to every momentum-coordinate pair $\mathbf{W},\ X \to P,\ Q$ one can obtain the regularized Hamiltonian $\Gamma = (T - U - E)/(T + U)$ and form the canonical equations of motion. Note that the number of new variables may exceed the number of the old ones. This, however, is no problem: all the physical results remain correct (Heggie, 1974).

The above formulation is completely general at least to the point that all the well known methods, the Zare method, Heggie's global regularization and the chain method are included. The vectors $\mathbf{X}$ of those methods are schematically illustrated in Figure 1.

Among these methods, Heggie's global regularization is actually simplest to use and program, since one needs no criteria for selecting the vectors ($\mathbf{X}$) to be regularized. Also the potential term due to the non-regularized potential do not appear in this method. In the other methods, however, one must somehow deem which interactions are the most critical and need regularization. As already mentioned, the present formulation is somewhat more general than any of the previous ones. One can regularize any interparticle vector. Thus any kind of branching and looping chains can be handled. This could be seen as an intermediate form between the Heggie method and the chain. However, it is not clear if such alternatives are actually more useful than chain.

## 4. Algorithmic regularization

By *algorithmic regularization* we mean here a numerical method in which the algorithm gives regular results at collision of two bodies without using coordinate transformation. The first such method was discovered simultaneously and independently by Mikkola and Tanikawa (1999ab) and Preto and Tremaine (1999). This particular method uses the *logarithmic Hamiltonian* (LogH)

$$\Lambda = \ln(T + B) - \ln(U), \tag{8}$$

which gives equations of motion

$$t' = \frac{\partial \Lambda}{\partial B} = 1/(T + B);\quad \mathbf{r}'_k = \mathbf{v}_k/(T + B);\quad \mathbf{v}'_k = \mathbf{A}_k/U \tag{9}$$

where $\mathbf{v}_k = \dot{\mathbf{r}}_k$ and $\mathbf{A}_k = \frac{\partial U}{\partial \mathbf{r}_k}/m_k$ are the velocity and acceleration correspondingly.

It is important to note that the derivatives of coordinates only depend on velocities and vice versa. This makes a simple leapfrog algorithm possible (see below). The most important feature is that the resulting leapfrog is *exact for two-body motion*, except for a phase error, and thus *regularizes close approaches.*

The derivation of the logarithmic Hamiltonian equations of motion can also be done in the way that one transforms the time in the Newtonian equations

of motion $\dot{\mathbf{v}} = \mathbf{A}$; $\dot{\mathbf{r}} = \mathbf{v}$ by dividing the first equation by $U$ and the second one by $T + B$, which gives correct equations (with the additional equation $t' = 1/(T + B)$) since $U = T + B$ along the orbit, due to energy conservation.

The *Time Transformed Leapfrog* (TTL) method is a generalization of this idea (Mikkola & Aarseth 2002). In the time transformation one takes some other function $\Omega(\mathbf{r})$ in place of the potential $U$ and defines an auxiliary quantity $W$ by the differential equation $\dot{W} = \dot{\Omega} = \frac{\partial \Omega}{\partial \mathbf{r}} \cdot \mathbf{v}$.

The resulting TTL equations read

$$t' = 1/W; \quad \mathbf{r}'_k = \frac{1}{W}\frac{\partial T}{\partial \mathbf{p}_k}; \quad \mathbf{v}'_k = \frac{1}{\Omega}\mathbf{A}_k; \quad W' = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \mathbf{v}_k/\Omega, \tag{10}$$

and these can also be used to construct a leapfrog-like mapping which, for suitable functions $\Omega$, are asymptotically exact for two-body motion near collision. It can be shown that TTL is mathematically equivalent to LogH if one takes $\Omega = U$.

### 4.1. LogH leapfrog

First one computes the constant $B = -T + U$ from initial values. The equations of motion can be used to define the basic subroutines $\mathbf{X}(s)$ and $\mathbf{V}(s)$ as

$$\mathbf{X}(s): \quad \delta t = s/(T + B); \quad \mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t\ \mathbf{v}_k; \quad t \rightarrow t + \delta t \tag{11}$$

and

$$\mathbf{V}(s): \quad \widetilde{\delta} t = s/U; \quad \mathbf{p} \rightarrow \mathbf{p}_k + \widetilde{\delta} t \mathbf{A}_k \tag{12}$$

which can be called in a sequence

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

using always the most recent results as input for the next operation.

### 4.2. TTL

Here one first evaluates the initial value of $W = \Omega$, then uses the leapfrog subroutines

$$\mathbf{X}(s): \quad \delta t = s/W; \quad \mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t\ \mathbf{v}_k; \quad t \rightarrow t + \delta t \tag{13}$$

$$\mathbf{V}(s): \quad \widetilde{\delta} t = s/\Omega; \quad \delta \mathbf{v}_k = \widetilde{\delta} t \mathbf{A}_k; \quad W \rightarrow W + \widetilde{\delta} t \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot (\mathbf{v}_k + \frac{1}{2}\delta \mathbf{v}_k);$$

$$\mathbf{v}_k \rightarrow \mathbf{v}_k + \delta \mathbf{v}_k, \tag{14}$$

advancing the coordinates and velocities using the operation sequence

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

repeatedly.

For $\Omega$ one may use any suitable function, but usually it is advantageous to take

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

where

$$\Omega_{ij} = 1, \qquad \text{or} \qquad \Omega_{ij} = m_i m_j,$$

the latter choice being recommendable if the masses are comparable.

The leapfrog alone is, however, in many cases not accurate enough. The accuracy can be improved e.g. by using the higher order leapfrog algorithms of Yoshida (1990). Alternatively one may use the extrapolation method (Press et al. 1986).

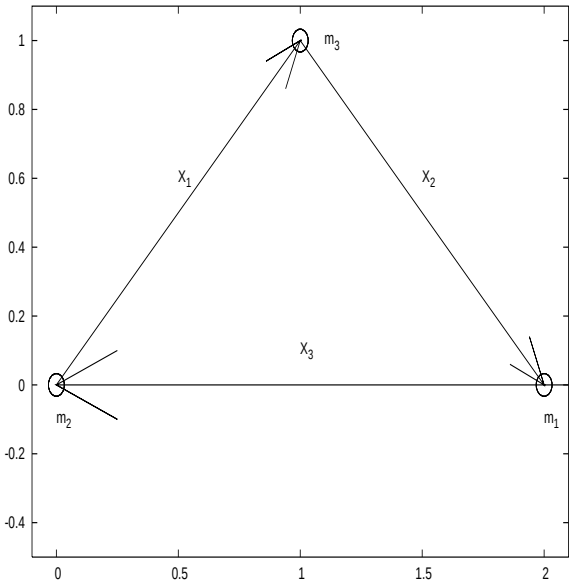


Figure 2. Labeling of vectors in the three-body regularization.

### 4.3. A simple LogH algorithm for the three-body problem

The three-body problem is still one of the most studied problems in few-body dynamics. Therefore it may be of interest to consider in more detail a simple regular three-body algorithm. This also serves as further illustration of the use of the algorithmic regularization.

Following Heggie (1974) we use the three interparticle vectors (see Figure 2)

$$\mathbf{X}_1 = \mathbf{r}_3 - \mathbf{r}_2; \quad \mathbf{X}_2 = \mathbf{r}_1 - \mathbf{r}_3; \quad \mathbf{X}_3 = \mathbf{r}_2 - \mathbf{r}_1. \tag{15}$$

as new coordinates. Let the corresponding velocities be $\mathbf{V}_k = \dot{\mathbf{X}}_k$, then the kinetic and potential energies can be written

$$T = \frac{1}{4M}\sum_{i<j} m_i m_j \mathbf{V}_{k_{ij}}^2; \quad U = \sum_{i<j} \frac{m_i m_j}{|X_{k_{ij}}|}, \tag{16}$$

where $M = \sum_k m_k$ is the total mass and $k_{ij} = 6 - i - j$.

The equations of motion are

$$\dot{\mathbf{X}}_k = \mathbf{V}_k; \quad \dot{\mathbf{V}}_k = -M\frac{\mathbf{X}_k}{|\mathbf{X}_k|^3} + m_k \sum_\nu \frac{\mathbf{X}_\nu}{|\mathbf{X}_\nu|^3}, \tag{17}$$

and after the application of the logarithmic Hamiltonian modification they read

$$t' = 1/(T+B); \quad \mathbf{X}'_k = \dot{\mathbf{X}}_k/(T+B); \quad \mathbf{V}'_k = \dot{\mathbf{V}}_k/U, \tag{18}$$

which are suitable for the leapfrog algorithm, given in equations (11) and (12), as well as for Yoshida's higher order leapfrogs.

The usage of the relative vectors, instead of some inertial coordinates, is advantageous in attempting to avoid large roundoff effects. One could also integrate only two of the triangle sides, obtaining the remaining one from the conditions

$$\sum_k \mathbf{X}_k = \mathbf{0}; \quad \sum_k \mathbf{V}_k = \mathbf{0}.$$

However this hardly reduces the computational effort required by the method. Instead one may, occasionally, compute the longest side, and the corresponding velocity, from the above triangle conditions. Note, however, that the sums of the sides are not only integrals of the exact solution, but they are also exactly conserved by the leapfrog mapping.

The transformation from the variables $\mathbf{X}$ to centre-of-mass coordinates $\mathbf{r}$ can be done as

$$\mathbf{r}_1 = \frac{(m_3\mathbf{X}_2 - m_2\mathbf{X}_3)}{M}; \quad \mathbf{r}_2 = \frac{(m_1\mathbf{X}_3 - m_3\mathbf{X}_1)}{M}; \quad \mathbf{r}_3 = \frac{(m_2\mathbf{X}_1 - m_1\mathbf{X}_2)}{M}, \tag{19}$$

and the velocities obey the same rule.

## 5. Final remarks

It is necessary to emphasize the importance of the chain structure, not only in the CHAIN-method, but also when one uses one or other of the algorithmic regularizations. The reason is roundoff errors. If one uses centre-of-mass coordinates, then the relative coordinates of a distant close pair are differences of large numbers and there is considerable cancellation leading to irrecoverable errors.

The author's impressions, based on experience, can be summarized as:

1. Heggie's method is the simplest to program (of the KS-based methods).
2. CHAIN is the most efficient KS-regularized code.

3. LogH is a good alternative for the lazy programmer.

4. TTL is, thus far, the only one that can handle large mass ratios although this particular problem is still partly unsolved.

5. For the chain algorithm, use of a high order numerical integrator, such as the extrapolation method (Press et al. 1986), is necessary. Similarly the TTL requires extrapolation to improve the leapfrog results. The LogH-method can also be improved with extrapolation, alternatively one may use a higher order leapfrog (Yoshida, 1990).

## References

Aarseth, S. J. and Zare, K., 1974, Cel. Mech., 10, 185–205.

Alexander, M. E., 1986, J. Comp. Phys., 64, 195–219.

Heggie, D. C., 1974, Cel. Mech., 10, 217–241.

Kustaanheimo, P. and Stiefel, E., 1965, J. Reine Angew. Math., 218, 204–219.

Mikkola, S., 1985, Mon. Not. R. Astr. Soc, 215, 171–177.

Mikkola, S. & Aarseth, S. J., 1993, Celest. Mech. Dyn. Astron. 57, 439

Mikkola, S. & Tanikawa, K, 1999a, Celestial Mechanics and Dynamical Astronomy, 74, 287-295.

Mikkola, S. & Tanikawa, K., 1999b, Mon. Not. R. Astr. Soc, 310, 745-749.

Mikkola, S. and Aarseth, S., 2002, Celestial Mechanics and Dynamical Astronomy, 84, 343-354.

Press, W. H., Flannery, B. P.,Teukolsky, S. A. and Wetterling, W. T., 1986, Numerical Recipes, Cambridge University Press.

Preto, M. and Tremaine, S., 1999, Astron. J., 118, 2532–2541.

Stiefel, E. L. and Scheifele, G., 1971, Linear and Regular Celestial Mechanics, Springer, Berlin.

Szebehely, V. and Zare, K, 1975, Cel. Mech., 11, 469.

Yoshida, H. , 1990, Phys. Lett. A., 150, 262–268.

Zare, K., 1974, Cel. Mech., 10, 207–215.