1 D Regularisation

1. Coordinate Transformation $r = u^2$

$$H = \frac{p^2}{2\mu} - \frac{GMm}{r} = E_0 = \text{const.} \quad p = \mu \dot{r} = 2u\dot{u}\mu$$

Canonical Trafo: $p\dot{r} = P\dot{u} \Rightarrow P = 4u^2\dot{u}\mu$

$$H = \frac{P^2}{8u^2\mu} - \frac{GMm}{u^2} = \text{const. } E_0$$

2. Time Transformation

$$dt = r\,ds = u^2 ds \; ; \quad \dot{u} = \frac{du}{dt} = \frac{1}{r}\frac{du}{ds} \Rightarrow$$

$$= g(p,r)$$
$$= g(P,u)$$

$$u^2\dot{u} = u' = \frac{1}{4}\frac{P}{\mu}$$

3. Poincare-Transform:

$$0 = \Gamma = g(P,u)\,H(P,u) = \frac{P^2}{8\mu} - GMm - E_0 u^2$$

4. Canonical Eq: $P' = \frac{\partial \Gamma}{\partial u} = -2E_0 u = 4u''\mu$

$$\Rightarrow \quad u'' + \frac{1}{2}\frac{E_0}{\mu} u = 0$$ harmonic oscillator (if $E_0 < 0$)

$$\omega^2 = \frac{E_0}{2\mu}$$ half frequency

2D

Need

$$r = L(u)u = u \circ u = u^2 = \left[\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}\right]^2$$

$$\mathring{r} = 2u\mathring{u}$$

Vector!

Complex numbers

Levi-Civita

Levi-Civita Transformation 2-D

$$H = \mu \frac{V^2}{2} - \frac{G m_1 m_2}{R} = H(V, R)$$

$$(u \cdot u) = R \Leftrightarrow L(u)\, u = R \Leftrightarrow L(u) = \begin{pmatrix} u_1 & -u_2 \\ u_2 & u_1 \end{pmatrix}$$

$$2 L(u) \dot{u} = \dot{R} = V$$

$$4 u^2 \dot{u}^2 = \dot{R}^2 = V^2 .$$

$$\Rightarrow \quad P = 4 u^2 \dot{u} = 2 L^T(u) \cdot V \quad \Rightarrow V^2 = \frac{P^2}{4u^2}$$

$$\Rightarrow \quad \frac{1}{2u^2} L(u) \cdot P = V$$

$$H = \mu \frac{P^2}{8u^2} - \frac{G m_1 m_2}{u^2} = H(P, u)$$

Time Transformation

$$\frac{dt}{ds} = R = u^2$$

Poincaré Transform

$$\Gamma = R(H - E) = \frac{\mu P^2}{8} - E u^2 - G m_1 m_2$$

harmonic oscillator $\omega^2 = \frac{\mu}{4E}$

half frequency

9.6.99 (1)

Are our transformations canonical?
(Levi-Civita plus time transformation)

$$S = \int_{t_1}^{t_2} L(q_i, \dot{q}_i, t)\, dt$$ S: action integral
L: Lagrange function

$\delta S = 0$ defines physical motion
(least action principle, Hamilton's variational princ.)

$$\delta S = \int_{t_1}^{t_2} \sum_i \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt$$ = (partial integr.) (variat. of boundary terms vanishes)

$$= \int_{t_1}^{t_2} \sum_i \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) \delta q \, dt$$

For all $\delta q_i \Rightarrow$ Euler-Lagrange Eq. of motion!

$$H = \sum_i p_i \dot{q}_i - L \iff L = \sum_i p_i \dot{q}_i - H$$

$$0 = \delta S = \int_{t_1}^{t_2} \sum_i \left(\dot{q}_i \delta p_i + p_i \delta \dot{q}_i - \frac{\partial H}{\partial p_i} \delta p_i - \frac{\partial H}{\partial q_i} \delta q_i \right) dt$$

$$= \int_{t_1}^{t_2} \left[\left(\dot{q}_i - \frac{\partial H}{\partial p_i} \right) \delta p_i - \left(\dot{p}_i + \frac{\partial H}{\partial q_i} \right) \delta q_i \right] dt$$

For all $\delta q_i, \delta p_i \Rightarrow$ Hamilton's Eq. of motion.

Canonical Transformation $\Longleftrightarrow$

$\Longleftrightarrow$ Hamilton's Eq. of motion invariant

$$\delta \int_{t_1}^{t_2} \left(\sum p_i \dot{q}_i - H \right) dt = 0$$

in both systems of variables.

1) Levi-Civita transformation

$$\sum_{i=1}^{3} p_i \dot{q}_i = \sum_{i=1}^{3} v_i \dot{r}_i = \sum_{i=1}^{3} v_i^2 \qquad \begin{array}{l} p_i = v_i \\ q_i = r_i \end{array}$$

Let $Q_i = u_i$, $P_i = 4u^2 \dot{u}_i$ Levi-Civita variables

$$\sum_{i=1}^{4} P_i \dot{Q}_i = 4u^2 \sum_{i=1}^{4} \dot{u}_i^2 = \sum_{i=1}^{3} v_i^2 = \sum_{i=1}^{3} p_i \dot{q}_i$$

since $\sum_{i=1}^{3} v_i^2 = \sum_{i=1}^{4} 4u^2 \dot{u}_i^2$ see before

---

2) Time transformation $dt = g(Q,t)\, ds$

$$\delta S = \delta \int_{t_1}^{t_2} \left(\sum_{i=1}^{3} p_i \dot{q}_i - H(p,q) \right) dt = \delta \int_{t_1}^{t_2} \left(\sum_{i=1}^{4} P_i \dot{Q}_i - H(P,Q) \right) dt$$

$$= \delta \int_{s_1}^{s_2} \left(\frac{1}{g} \sum_{i=1}^{4} P_i Q_i' - H(P,Q) \right) g \, ds$$

$$\boxed{Q_i' = \frac{\partial Q_i}{\partial s} = \frac{\partial Q_i}{\partial t} \cdot \frac{\partial t}{\partial s} = \dot{Q}_i \cdot g \, ; \; \Gamma = g\left(H - \frac{h}{\mu}\right)} = \delta \int_{s_1}^{s_2} \left(\sum_{i=1}^{4} P_i Q_i' - \Gamma \right) ds$$

$p_i, q_i, H(p_i, q_i, t), t$ satisfies Hamilton's eq.

$$\Longleftrightarrow$$

$P_i, Q_i, \Gamma(P_i, Q_i, s), s$ satisfies Hamilton's eq.

Γ: Poincaré transform of Hamiltonian

$$\Gamma = g\left(H - \frac{h}{\mu}\right) \qquad dt = g \cdot ds$$

Our example: $$g = r = u^2 = \sum_{i=1}^{4} u_i^2$$

$$H = \frac{1}{2}\sum_{i=1}^{3} v_i^2 - \frac{Gm_1m_2}{\mu r} = 2u^2 \sum_{i=1}^{4} \dot{u}_i^2 - \frac{Gm_1m_2}{\mu u^2}$$

$$\Gamma = r \cdot \left(H - \frac{h}{\mu}\right) = 2u^4 \sum_{i=1}^{4} \dot{u}_i^2 - \frac{Gm_1m_2}{\mu} - \frac{h}{\mu}u^2$$

$$= \sum_{i=1}^{4}\left(\frac{P_i^2}{8} - \frac{h}{\mu}Q_i^2\right) - \frac{Gm_1m_2}{\mu}$$

Canonical Equations:

$$Q_i' = \frac{\partial \Gamma}{\partial P_i} = \frac{P_i}{4} \Longleftrightarrow u_i' = u^2 \dot{u}_i = r\dot{u}_i \quad \text{O.K.}$$

$$P_i' = -\frac{\partial \Gamma}{\partial Q_i} = 2\frac{h}{\mu}Q_i \Longleftrightarrow \left(4u^2\dot{u}_i\right)' = 4u_i'' = 2\frac{h}{\mu}u_i$$

$u_i'' - \frac{h}{2\mu}u_i = 0$	$u_i'' + \frac{\lvert h \rvert}{2\mu}u_i = 0 \quad (h<0)$

Henceforth:

Time transformation $dt = R_1 R_2 \, ds$

Poincaré Hamiltonian $\Gamma = R_1 R_2 \left(H - \quad \right)$

e.g. $\vec{R}_1 = \vec{r}_1 - \vec{r}_3$
$\vec{R}_2 = \vec{r}_2 - \vec{r}_3$ for three bodies!

Hamiltonian equations $\Longrightarrow$

regularized equations of motion.

Next: Aarseth + Zare '74:
3-body regularization.

Three-Body Regularization

i) Three-Body Hamiltonian

$$H = \frac{1}{2}\left(m_1 v_1^2 + m_2 v_2^2 + m_3 v_3^2\right)$$

$$- \frac{G m_1 m_2}{r_1 - r_2} - \frac{G m_2 m_3}{r_2 - r_3} - \frac{G m_1 m_3}{r_1 - r_3}$$

$$\vec{v}_{CM} = \frac{m_1 \vec{v}_1 + m_2 \vec{v}_2 + m_3 \vec{v}_3}{m_1 + m_2 + m_3} = \frac{\sum m_i \vec{v}_i}{M}$$

$$\vec{V}_1 = \vec{v}_1 - \vec{v}_3 \qquad \vec{R}_1 = \vec{r}_1 - \vec{r}_3$$

$$\vec{V}_2 = \vec{v}_2 - \vec{v}_3 \qquad \vec{R}_2 = \vec{r}_2 - \vec{r}_3$$

$$\vec{r}_{CM} = \frac{\sum m_i \vec{r}_i}{M} \qquad r_i = |\vec{r}_i| \quad v_i = |\vec{v}_i|$$

$$R_i = |\vec{R}_i| \quad V_i = |\vec{V}_i|$$

$$M\vec{v}_{cm} = m_1\vec{v}_1 + m_2\vec{v}_2 + m_3\vec{v}_3 \quad (1)$$

$$m_3\vec{V}_1 = m_3\vec{v}_1 - m_3\vec{v}_3 \quad (2)$$

$$m_3\vec{V}_2 = m_3\vec{v}_2 - m_3\vec{v}_3 \quad (3)$$

$$M\vec{v}_{cm} + m_3\vec{V}_1 = (m_1+m_3)\vec{v}_1 + m_2\vec{v}_2 \quad (4)=(1)+(2)$$

$$M\vec{v}_{cm} + m_3\vec{V}_2 = (m_2+m_3)\vec{v}_2 + m_1\vec{v}_1 \quad (5)=(1)+(3)$$

$$(m_2+m_3)M\vec{v}_{cm} + (m_2+m_3)m_3\vec{V}_1 = (m_2+m_3)(m_1+m_3)\vec{v}_1 + (m_2+m_3)m_2\vec{v}_2$$

(4) * (m_2+m_3) = (6)

$$-m_2M\vec{v}_{cm} - m_2m_3\vec{V}_2 = -m_1m_2\vec{v}_1 - m_2(m_2+m_3)\vec{v}_2$$

(5) * $(-m_2)$ = (7)

$$m_3M\vec{v}_{cm} + (m_2+m_3)m_3\vec{V}_1 - m_2m_3\vec{V}_2 =$$

(8)
$$= (m_2+m_3)(m_1+m_3)\vec{v}_1 - m_1m_2\vec{v}_1$$
$$= (m_2m_3 + m_3(m_1+m_3))\vec{v}_1 = m_3M\vec{v}_1$$

$$\vec{v}_1 = \vec{v}_{cm} + \frac{m_2+m_3}{M}\vec{V}_1 - \frac{m_2}{M}\vec{V}_2 \quad (8)/(m_3M)$$

$$\vec{v}_2 = \vec{v}_{cm} + \frac{m_1+m_3}{M}\vec{V}_2 - \frac{m_1}{M}\vec{V}_1 \quad (9)$$ analogous 1 and 2 exchanged

$$\vec{V}_3 = \left(M\vec{V}_{cm} - m_1\vec{v}_1 - m_2\vec{v}_2 \right) / m_3 =$$

$$= \Big(M\vec{V}_{cm} - m_1\vec{v}_{cm} - \frac{m_1}{M}(m_2+m_3)\vec{V}_1 + \frac{m_1 m_2}{M}\vec{V}_2$$

$$- m_2\vec{v}_{cm} - \frac{m_2}{M}(m_1+m_3)\vec{V}_2 + \frac{m_1 m_2}{M}\vec{V}_1 \Big) / m_3$$

$$= \vec{V}_{cm} - \frac{m_1}{M}\vec{V}_1 - \frac{m_2}{M}\vec{V}_2 \qquad (10)$$

Transformation:

$$T = \frac{1}{2}\left(m_1 v_1^2 + m_2 v_2^2 + m_3 v_3^2 \right) = \frac{1}{2}\Big(M V_{cm}^2 + m_1 \frac{(m_2+m_3)^2}{M^2} V_1^2 + \frac{m_1 m_2^2}{M^2} V_2^2$$

$$+ m_2 \frac{(m_1+m_3)^2}{M^2} V_2^2 + \frac{m_2 m_1^2}{M^2} V_1^2 + \frac{m_3 m_1^2}{M^2} V_1^2 + \frac{m_3 m_2^2}{M^2} V_2^2$$

$$- 2\frac{m_1 m_2 (m_2+m_3)}{M^2} V_1 V_2 - 2\frac{m_1 m_2 (m_1+m_3)}{M^2} V_1 V_2 + 2\frac{m_1 m_2 m_3}{M^2} V_1 V_2$$

$$+ 2\frac{m_1(m_2+m_3)}{M} V_1 v_{cm} + 2\frac{m_2(m_1+m_3)}{M} V_2 v_{cm} - 2\frac{m_1 m_3}{M} V_1 v_{cm}$$

$$- 2\frac{m_1 m_2}{M} V_2 v_{cm} - 2\frac{m_1 m_2}{M} V_1 v_{cm} - 2\frac{m_2 m_3}{M} V_2 v_{cm} \Big)$$

Collect Terms: $V_1 V_2$:

$$\frac{2}{M^2}\left(m_1 m_2 m_3 - m_1 m_2 (m_1+m_3) - m_1 m_2 (m_2+m_3) \right) =$$

$$= \frac{2}{M^2}\left(-m_1 m_2 m_3 - m_1^2 m_2 - m_1 m_2^2 \right) = -\frac{2 m_1 m_2}{M}$$

Collect Terms V_1^2:

$$\frac{1}{M^2}\left(m_1(m_2+m_3)^2 + m_2 m_1^2 + m_3 m_1^2\right) =$$

$$= \frac{1}{M^2}\left(m_1 m_2^2 + 2 m_1 m_2 m_3 + m_1 m_3^2 + m_2 m_1^2 + m_3 m_1^2\right)$$

$$= \frac{1}{M^2} m_1 \left(m_2^2 + m_3^2 + 2 m_2 m_3 + m_1(m_2+m_3)\right)$$

$$= \frac{1}{M^2} m_1 (m_2+m_3)(m_1+m_2+m_3) = \frac{m_1(m_2+m_3)}{M}$$

Analogous V_2^2: $\dfrac{m_2(m_1+m_3)}{M}$

All Terms $V_1 v_{cm}$, $V_2 v_{cm}$ vanish! $\Longrightarrow$

$$H = \frac{1}{2} M v_{cm}^2 + \frac{1}{2}\frac{m_1(m_2+m_3)}{M} V_1^2 + \frac{1}{2}\frac{m_2(m_1+m_3)}{M} V_2^2$$

$$- \frac{m_1 m_2}{M} V_1 V_2 - \frac{G m_1 m_2}{R_1 - R_2} - \frac{G m_2 m_3}{R_2} - \frac{G m_1 m_3}{R_1}$$

Note: $r_1 - r_2 = R_1 - R_2$ has been used.

Note: $V_1 - V_2 = V_{rel}$, $R_1 - R_2 = R_{rel}$ $\Longrightarrow$

$$H = \frac{1}{2} M v_{cm}^2 + \frac{1}{2}\frac{m_1 m_2}{M} V_{rel}^2 - \frac{G m_1 m_2}{R_{rel}}$$

$$+ \frac{1}{2}\frac{m_1 m_3}{M}\left(V_1^2 + V_2^2\right) - \frac{G m_2 m_3}{R_2} - \frac{G m_1 m_3}{R_1}$$

$$\frac{m_2 m_3}{M}$$

Regularization:

$\vec{Q}_i, \vec{P}_i \in \mathbb{R}^4$

$$dt = R_1 R_2\, ds$$

$$R_1 = Q_1^2 \qquad \vec{R}_1 = L(\vec{Q}_1)\vec{Q}_1$$

$$R_2 = Q_2^2 \qquad \vec{R}_2 = L(\vec{Q}_2)\vec{Q}_2$$

$$\vec{P}_1 = 4Q_1^2 \dot{\vec{Q}}_1 = 4\vec{Q}_1' / Q_2^2 \qquad \vec{V}_1 = 2 \cdot L(\vec{Q}_1) \frac{\vec{P}_1}{4Q_1^2}$$

$$\vec{P}_2 = 4Q_2^2 \dot{\vec{Q}}_2 = 4\vec{Q}_2' / Q_1^2 \qquad \vec{V}_2 = 2 \cdot L(\vec{Q}_2) \frac{\vec{P}_2}{4Q_2^2}$$

$$V_1^2 = \dot{R}_1^2 = 4Q_1^2 \dot{Q}_1^2 = \frac{P_1^2}{4Q_1^2}; \quad V_2^2 = \frac{P_2^2}{4Q_2^2}$$

$$\Gamma = R_1 R_2 (H - E) = Q_1^2 Q_2^2 (H - E)$$

$$= \frac{m_1(m_2+m_3)}{M} \frac{P_1^2 Q_2^2}{8} + \frac{m_2(m_1+m_3)}{M} \frac{P_2^2 Q_1^2}{8}$$

$$- \frac{m_1 m_2}{M} \cdot \frac{L(\vec{Q}_1)\vec{P}_1\, L(\vec{Q}_2)\vec{P}_2}{4}$$

$$- \frac{G m_1 m_2 Q_1^2 Q_2^2}{|\vec{R}_1 - \vec{R}_2|} - G m_2 m_3 Q_1^2 - G m_1 m_3 Q_2^2 - E Q_1^2 Q_2^2$$

Next: Equations of Motion!

30.6.99 (1)

3-6 Equations of Motion

$$\vec{Q}_1' = \frac{\partial \Gamma}{\partial \vec{P}_1} = \frac{m_1(m_2+m_3)}{M}\frac{P_1 Q_2^2}{4} - \frac{m_1 m_2}{M}\frac{L^T(\vec{Q}_1)L(\vec{Q}_2)\vec{P}_2}{4}$$

$$\vec{P}_1' = -\frac{\partial \Gamma}{\partial \vec{Q}_1} = \frac{m_2(m_1+m_3)}{M}\frac{\vec{Q}_1 P_2^2}{4} - \frac{m_1 m_2}{M}\frac{L^T(\vec{P}_1)L(\vec{Q}_2)\vec{P}_2}{4}$$

$$- \frac{2Gm_1m_2}{|R_1-R_2|}\vec{Q}_1 Q_2^2 - 2Gm_2m_3\vec{Q}_1 - 2E\vec{Q}_1 Q_2^2$$

$$\vec{Q}_1' = \frac{\vec{P}_1}{4} \qquad \vec{P}_1' = 4\vec{Q}_1'' = \vec{Q}_1\vec{X} + \vec{Y}$$

$$\vec{X} = \frac{m_2(m_1+m_3)}{M}\frac{P_2^2}{4} - \frac{2Gm_1m_2}{|\vec{R}_1-\vec{R}_2|}Q_2^2 - 2EQ_2^2$$

$$\vec{Y} = -\frac{m_1m_2}{M}\frac{L^T(\vec{P}_1)L(\vec{Q}_2)\vec{P}_2}{4}$$

$$\boxed{\vec{Q}_1'' - \omega^2\vec{Q}_1 = \frac{1}{4}\vec{Y}} \qquad \omega^2 = \frac{1}{4}\vec{X}$$

Harmonic Oscillator with external quasi-periodic triggering $\Rightarrow$ resonances, strongly chaotic system (double pendulum!)

More Regularizations?

- 3-body

Aarseth 85: $$dt = \frac{R_1 R_2}{\sqrt{R_1 + R_2}}\, ds$$

- N-body

Heggie 74: $$dt = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} R_{ij}\, ds$$

$$R_{ij} = |\vec{r}_i - \vec{r}_j| = L(\vec{Q}_{ij})\, \vec{Q}_{ij}$$

$$\vec{V}_{ij} = \dot{\vec{R}}_{ij} = 2L(\vec{Q}_{ij})\, \vec{P}_{ij} / 4Q_{ij}^2$$

$$H(\vec{V}_{ij}, \vec{R}_{ij}) = \sum_{i<j} \frac{V_{ij}^2}{2\mu_{ij}} + \overset{\text{Mikkola 97}}{\overset{\downarrow}{\ldots}} + \sum_{i<j} \frac{G m_i m_j}{R_{ij}}$$

$$\Gamma = \left(H(V_{ij}, R_{ij}) - E \right) \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} R_{ij}$$

$$= \left[H\left(\frac{L(Q_{ij}) P_{ij}}{2 Q_{ij}^2}, L(Q_{ij}) Q_{ij} \right) - E \right] \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} Q_{ij}^2$$

Heggie's Global Regularization

Very Complicated Equations of Motion!

High Dimensionality: $N(N-1)/2$

- Chain Method (Mikkola)

$\vec{R}_K = \vec{r}_{K+1} - \vec{r}_K \qquad \vec{R}_K = L(\vec{Q}_K)\vec{Q}_K$

$$dt = \frac{1}{T+U} ds$$

- Slow-Down Treatment Mikkola, Aarseth 96

$$\ddot{\vec{r}} = -\frac{G m_1 m_2}{K^2 |\vec{r}_1 - \vec{r}_2|^3} (\vec{r}_1 - \vec{r}_2) + \vec{F}_{ext}$$

Slow-Down coefficient K

$$\dot{\vec{r}} = \frac{1}{K} \cdot \vec{v}$$

Period is K-fold longer!

- Stumpff-Functions

$$\vec{u}'' + \frac{|h|}{2\mu} \vec{u} = \frac{u^2}{2} L^T(\vec{u}) \vec{F}_{ext} = \frac{u^2}{2}$$

Use Stumpff functions for series evaluation of solution. See later....

Examples of the regularizing transformations for special orbits

i) circular motion $\dot{r} = 0$

$$c^2 = r^2 \dot{\varphi}^2 = r^2 v_\varphi^2 = \frac{G m_1 m_2}{\mu r^2} \cdot r^3 \implies$$

$$v_\varphi^2 = \frac{G m_1 m_2}{\mu r} = \frac{G(m_1 + m_2)}{r}$$

$$h = \frac{1}{2}\mu v_\varphi^2 - \mu v_\varphi^2 = -\frac{1}{2}\mu v_\varphi^2 = -\frac{G(m_1 \cdot m_2)}{2r}$$

Reg. Transformation 1:

$$d\varphi = \frac{c}{r^2}\,dt = \frac{v_\varphi}{r}\,dt = \sqrt{G(m_1 + m_2)} \cdot r^{-3/2}\,dt$$

φ: true anomaly, „time dilatation" for $r \to 0$

Reg. Transformation 2:

$$du = \sqrt{\frac{2|h|}{\mu}}\,\frac{dt}{r} = \frac{v_\varphi}{r}\,dt = \sqrt{G(m_1 + m_2)} \cdot r^{-3/2}\,dt$$

For circular orbit same!

u: eccentric anomaly, fictitious time

ii) radial motion $c = 0$, $v_r = \mathring{r}$

$$\mathring{v}_r = -\frac{Gm_1m_2}{\mu r^2}$$

$$h = \frac{1}{2}\mu v_r^2 - \frac{Gm_1m_2}{r} = \text{const.} = h_0$$

Reg. Transformation 1:

not defined for $c = 0$

Reg. Transformation 2:

$$du = \sqrt{\frac{2Gm_1m_2}{\mu r} - v_r^2}\,\frac{dt}{r} = \sqrt{\frac{2|h_0|}{\mu}}\,\frac{dt}{r}$$

still well defined, but time dilatation $\to \infty$ for $r \to 0$

well suited for eccentric motion!

Levi-Civita – Transformation

Why? Regularization interesting for perturbed 2-body motion (by 3rd body). $\Rightarrow$ Motion not confined in plane!

- Can we find regularization of full 2D-vector equation of motion? Levi-Civita
- Can we find regularization of full 3D-vector equation of motion?

Kustaanheimo + Stiefel, generalized Levi-Civita

2D – Vector Equations of Motion

$$\ddot{\vec{r}} = - \frac{G m_1 m_2}{\mu r^3} \vec{r}$$

$$\mu \frac{\dot{\vec{r}}^2}{2} - \frac{G m_1 m_2}{r} = h = \text{const.}$$

4.

Start with regularization of eccentric anomaly type:

$$ds = \frac{dt}{r} \qquad \frac{1}{r}\frac{d}{ds} = \frac{d}{dt}$$

from (*):

$$\frac{1}{r}\frac{d}{ds}\frac{1}{r}\frac{d}{ds}\vec{r} + \frac{Gm_1m_2}{\mu r^3}\vec{r} = 0$$

$$\frac{1}{r}\left(\frac{\vec{r}''}{r} - \frac{r'}{r^2}\vec{r}'\right) + \frac{Gm_1m_2}{\mu r^3}\vec{r} = 0$$

$$\boxed{\vec{r}'' - \frac{r'}{r}\vec{r}' + \frac{Gm_1m_2}{\mu r}\vec{r} = 0} \qquad (3)$$

from (**):

$$\frac{1}{r^2}\left(\frac{d}{ds}\vec{r}\right)^2 - \frac{2Gm_1m_2}{\mu r} = 2h/\mu$$

$$\boxed{\frac{\vec{r}'^2}{r^2} - \frac{2Gm_1m_2}{\mu r} = 2h/\mu} \qquad (4)$$

We are not done since $r' \cdot \vec{r}' \neq \vec{r}'^2$

Search for Transformation 2D

5

- $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \longrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ real space

mapping from 2D vector space into 2D v.sp.

- should be conformal

$$du^2 = \alpha \, dx^2 \qquad \alpha \neq 0$$

(different orbits should not be mapped onto each other!)

$$\Rightarrow \quad L(\vec{u})\vec{u} = \vec{x}$$

Matrix $L(u)$ orthogonal!

- further condition:

$$L(\vec{u})\vec{u} \;\hat{=}\; \vec{u}^2 = \vec{u} \cdot \vec{u}$$

with $\cdot$ = product in field („Körper")

Note 2D: $\cdot \Rightarrow$ commutative product in field $\mathbb{C}$ (complex numbers)

4D: $\cdot \Rightarrow$ non-commutative product $\mathbb{Q}$ Quaternions (Schiefkörper)

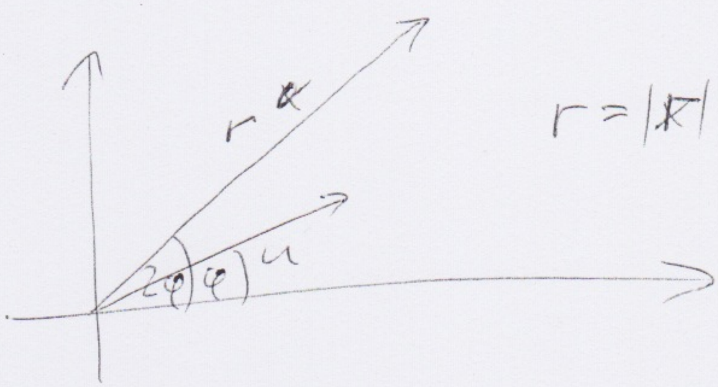
8D: $\cdot \Rightarrow$ „nullteilerfreies" product Cayley's numbers

Ansatz:

$L(\vec{u})\cdot\vec{u} = \vec{u}^2$ product of complex numbers

$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = u_1 + i u_2$ $\qquad i^2 = -1$

$\vec{u}^2 = \vec{r} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} u_1^2 - u_2^2 \\ 2u_1u_2 \end{pmatrix}$ $\qquad \begin{matrix} \mathrm{Re}\, u^2 \\ \mathrm{Im}\, u^2 \end{matrix}$



$r = |\vec{r}| = |u|^2$

$L(\vec{u}) = \begin{pmatrix} u_1 & -u_2 \\ u_2 & u_1 \end{pmatrix}$

$$x_1 = u_1^2 - u_2^2$$
$$x_2 = 2u_1u_2$$

Levi-Civita transformation

$$x_1' = 2u_1u_1' - 2u_2u_2'$$
$$x_2' = 2u_1u_2' + 2u_1'u_2$$

$\vec{r} = L(\vec{u})\vec{u}$ $\qquad \vec{r}\,' = 2L(\vec{u})\vec{u}\,'$

$$ds^2 = dx_1^2 + dx_2^2 = (2u_1\,du_1 - 2u_2\,du_2)^2 + (2u_1\,du_2 + 2u_2\,du_1)^2$$

$$= 4\,(u_1^2 + u_2^2)\,(du_1^2 + du_2^2)$$

conformal mapping!

$u_1^2 + u_2^2 = 0$ if and only if

$x_1^2 + x_2^2 = 0$!

$\Rightarrow$ therefore no divisors of zero may exist in algebra!

$$\boxed{\Rightarrow\ r'^2 = 4\,u^2\,u'^2}$$

$$\vec{r} = L(\vec{u})\,\vec{u} \quad = (\vec{u}\cdot\vec{u})$$

$$\vec{r}\,' = 2L(\vec{u})\,\vec{u}\,'$$

$$\vec{r}\,'' = 2L(\vec{u}\,')\,\vec{u}\,' + 2L(\vec{u})\,\vec{u}\,''$$

since $L(\vec{u})$ linear, homogeneous in $\vec{u}$

$$L^T(\vec{u})L(\vec{u}) = |\vec{u}|^2\cdot E = |\vec{u}|^2\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

$$L^T(\vec{u}) = \begin{pmatrix}u_1 & u_2\\ -u_2 & u_1\end{pmatrix} \quad \Longrightarrow$$

$$L^{-1}(\vec{u}) = \frac{1}{|\vec{u}|^2}\,L^T(\vec{u})$$

$$r'^2 = |\vec{r}\,'|^2 = 4|\vec{u}|^2|\vec{u}\,'|^2 \quad ; \text{ from (4):}$$

$$\frac{4|\vec{u}|^2\,(\vec{u}\,'\cdot\vec{u}\,')}{|\vec{u}|^4} - \frac{2Gm_1m_2}{\mu|\vec{u}|^2} = \frac{2h}{\mu}$$

$$(\vec{u}\,'\cdot\vec{u}\,') = \frac{Gm_1m_2}{2\mu} + \frac{h}{2\mu}|\vec{u}|^2$$

from (3):

$$2L(\vec{u})\vec{u}'' + 2L(\vec{u}')\vec{u}' - \frac{r'}{|\vec{u}|^2}\cdot 2L(\vec{u})\vec{u}'$$

$$+ \frac{Gm_1m_2}{\mu|\vec{u}|^2} L(\vec{u})\vec{u} = 0 \quad \Big| \cdot \frac{1}{2}L^{-1}(\vec{u})$$

$$\vec{u}'' + \frac{Gm_1m_2}{2\mu|\vec{u}|^2}\vec{u} - \frac{r'}{|\vec{u}|^2}\vec{u}' + \frac{L^T(\vec{u})L(\vec{u}')}{|\vec{u}|^2}\vec{u}' = 0$$

$$\boxed{\vec{u}'' + \frac{Gm_1m_2}{2\mu|\vec{u}|^2}\vec{u} - \frac{\vec{u}'\cdot\vec{u}'}{|\vec{u}|^2}\vec{u} = 0}$$

to be demonstrated: $(\vec{u}'\cdot\vec{u}') = L(\vec{u}')\vec{u}'$

$$\boxed{\vec{u}'' - \frac{h}{2\mu}\vec{u} = 0}$$

2D periodic motion with period

$$\omega^2 = -\frac{h}{2\mu}$$

N-Body Lecture

Rainer Spurzem

December 2008

Video files from previous lecture:

http://www.ari.uni-heidelberg.de/mitarbeiter/spurzem/nbody-lecture/

Lectures I – VIII completed until Dec 15, 2008

NBODYn n=?	individual steps: ITS steady	ABS scheme	HITS blocked	Hermite scheme	Regularisation	AC neighbour scheme ACS/HACS	Remarks
0,1 (1h)	×		(×)				
2	×					×	
3	×				×		
5	×				×	×	1985
4			×		×		GRAPE Version exists
ICHIRO			×			T-sym.	Planetesimals
6			×		×	×	GRAPE version not good
6++			×		×	×	MPI mass par. Vers.
7			×		×	×	GR treatm of mass. binaries
KIRA			×		only 2B no KS		GRAPE Version exists
Systolic non-blocking			×				Hemsend. Dorband Merritt
φ-GRAPE			×				GRAPE Berczik Harfst

HYBRIDS:

NBODYx : with TREE, NBODY5, Aarseth+McMillan 93

Eurostar : with SCF, NBODY6++, Hemsendorf et al 02

? : : with TREE, NBODY6++, Miocchi+Sp, 02

WINE : : with SPH/TREE, Wetzstein, Burkert, Naab 03

φGRAPE : like NBODY1, parallel GRAPE, Harfst, .., Berczik et al NewA 2007

Spurz. GRK

Sources of Code/Information:

Sverre Aarseth
http://www.sverre.com
http://www.ast.cam.ac.uk/~sverre
http://www.nbodylab.org
→ anon. ftp - sample codes ...

Rainer Spurzem/Thomas Brüsemeister

ftp://ftp.ari.uni-heidelberg.de/pub/staff/Spurzem/nb6mpi/

xxx http://svn.ari.uni-heidelberg.de/repos/nbody xxx
http://www. " " " /mitarbeiter/bruesemeister/agdguide/

Shigeru Ida's Home Page

http://www.geotech.ac.jp/lab/ida/ida/top.htm

(Japanese mostly)

Eiichiro Kokubo's Home Page:

http://yso.mtk.nao.ac.jp/~kokubo

---

xxx Under Construction!

Chapter 2. NBODY6++ Deployment

Table of Contents

Overview

For NBODY6++ there is a deployment package available which contains scripts to submit, monitor and kill NBODY6++ jobs. It also provides automatic data-staging, that is, the NBODY6++ input and output files are automatically transferred back and forth. The NBODY6++ source code which is not part of the submission package must be separately downloaded (usually via SVN).

Note: The deployment package requires a NBODY6++ version which uses the GNU autotools (autoconf, automake) for the build process. Currently only the development version in the trunk of the NBODY Subversion repository supports this.

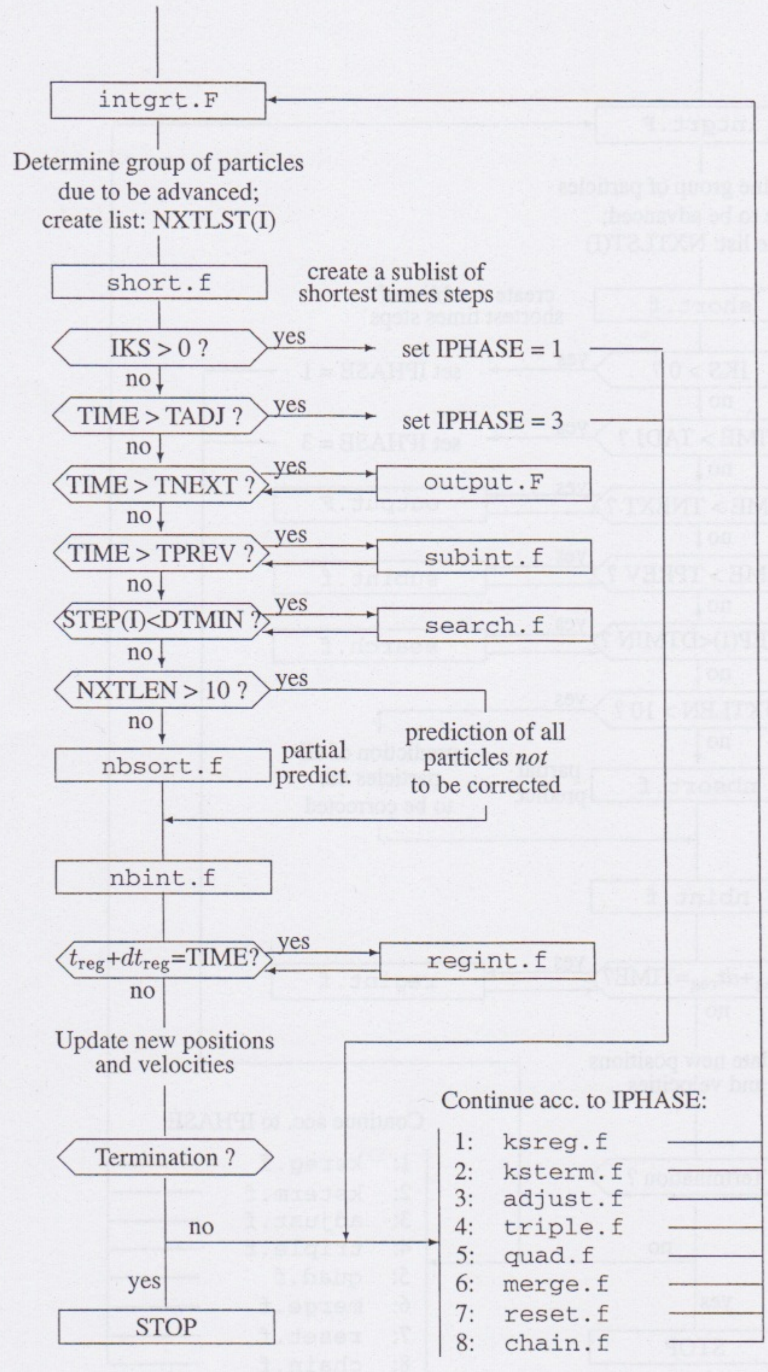
The AstroGrid-D deployment scripts for NBODY6++ can be downloaded from the AstroGrid-D software repository. The current (stable) branch is 0.2.x.

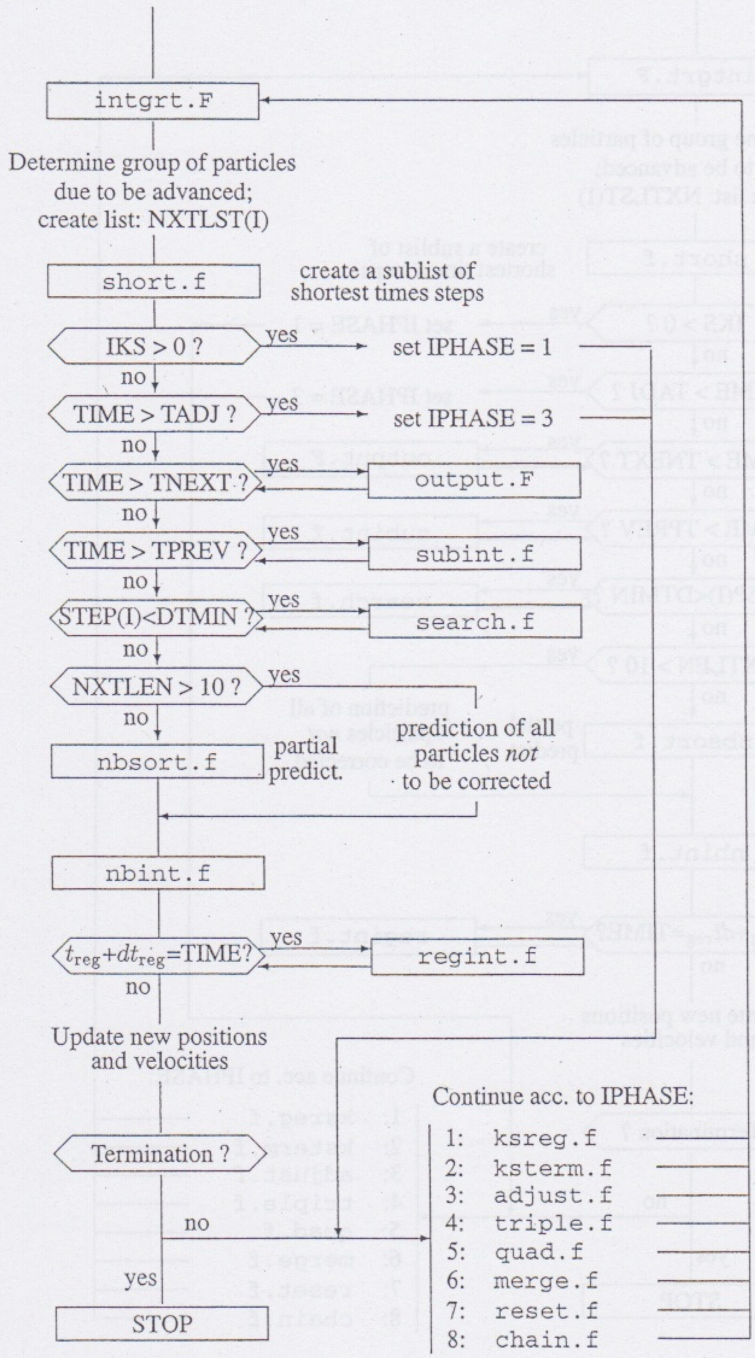
```
svn co http://svn.ari.uni-heidelberg.de/repos/nbody/deployment/branches/0.2.x nb6deployment
```

The scripts expect the Nbody6++ source code in the 'nbody6src' subdirectory of the deployment package. If the directory does not exist and the user tries to submit a job the script will automatically ask if it should fetch the sources from the SVN server. This step can also be done manually:

Manual download:

```
svn co http://svn.ari.uni-heidelberg.de/repos/nbody/nbody6/trunk nbody6src
```


intgrt.F
Determine group of particles
due to be advanced;
create list: NXTLST(I)
short.f
create a sublist of
shortest times steps
IKS > 0 ?
yes
set IPHASE = 1
no
TIME > TADJ ?
yes
set IPHASE = 3
no
TIME > TNEXT ?
yes
output.F
no
TIME > TPREV ?
yes
subint.f
no
STEP(I)<DTMIN ?
yes
search.f
no
NXTLEN > 10 ?
yes
no
nbsort.f
partial
predict.
prediction of all
particles not
to be corrected
nbint.f
$t_{reg}+dt_{reg}$=TIME?
yes
regint.f
no
Update new positions
and velocities
Continue acc. to IPHASE:
1: ksreg.f
2: ksterm.f
3: adjust.f
4: triple.f
5: quad.f
6: merge.f
7: reset.f
8: chain.f
Termination ?
no
yes
STOP


intgrt.F
Determine group of particles due to be advanced; create list: NXTLST(I)
short.f
create a sublist of shortest times steps
IKS > 0 ?
yes
set IPHASE = 1
no
TIME > TADJ ?
yes
set IPHASE = 3
no
TIME > TNEXT ?
yes
output.F
no
TIME > TPREV ?
yes
subint.f
no
STEP(I)<DTMIN ?
yes
search.f
no
NXTLEN > 10 ?
yes
no
nbsort.f
partial predict.
prediction of all particles *not* to be corrected
nbint.f
$t_{reg}+dt_{reg}$=TIME?
yes
regint.f
no
Update new positions and velocities
Termination ?
no
yes
STOP
Continue acc. to IPHASE:
1: ksreg.f
2: ksterm.f
3: adjust.f
4: triple.f
5: quad.f
6: merge.f
7: reset.f
8: chain.f

2 Das Hermite-Schema

Ein Verfahren höherer Genauigkeit erhält man mit dem sogenannten Hermite-Schema. Man berechne zunächst zu einer Zeit t_0 Beschleunigung und deren Zeitableitung auf ein Teilchen an der Position $\mathbf{r}$, Geschwindigkeit $\mathbf{v}$ durch

$$\mathbf{a}_0 = \sum_j Gm_j \frac{\mathbf{r}_{ij}}{r_{ij}^3} \; ; \; \dot{\mathbf{a}}_0 = \sum_j Gm_j \left[\frac{\mathbf{v}_{ij}}{r_{ij}^3} - \frac{3(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})\mathbf{r}_{ij}}{r_{ij}^5} \right] \tag{4}$$

mit $\mathbf{r}_{ij} := \mathbf{r}_j - \mathbf{r}_i$, $\mathbf{v}_{ij} := \mathbf{v}_j - \mathbf{v}_i$, $r_{ij} := |\mathbf{r}_{ij}|$, $v_{ij} := |\mathbf{v}_{ij}|$; Durch

$$\begin{aligned}
\mathbf{x}_p(t) &= \frac{1}{6}(t-t_0)^3\dot{\mathbf{a}}_0 + \frac{1}{2}(t-t_0)^2\mathbf{a}_0 + (t-t_0)\mathbf{v}_0 + \mathbf{x}_0 \\
\mathbf{v}_p(t) &= \frac{1}{2}(t-t_0)^2\dot{\mathbf{a}}_0 + (t-t_0)\mathbf{a}_0 + \mathbf{v}_0
\end{aligned} \tag{5}$$

extrapoliere man die Teilchenpositionen und Geschwindigkeiten für einen späteren Zeitpunkt $t = t_0 + dt$ ("Prediction"). Hier wurde der Index i für das betrachtete Teilchen weggelassen, und der Index 0 steht für die Werte zum Zeitpunkt t_0, p für "predicted". Mit Hilfe der "predicted" Positionen und Geschwindigkeiten für alle Teilchen j können nun neue Werte $\mathbf{a}_1$ und $\dot{\mathbf{a}}_1$ wie oben, aber für den neuen Zeitpunkt $t > t_0$ berechnet werden.

Ebenso könnte man $\mathbf{a}_1$ und $\dot{\mathbf{a}}_1$ aus einer Taylorreihe berechnen:

$$\begin{aligned}
\mathbf{a}_1 &= \frac{1}{6}(t-t_0)^3\mathbf{a}^{(3)} + \frac{1}{2}(t-t_0)^2\mathbf{a}^{(2)} + (t-t_0)\dot{\mathbf{a}}_0 + \mathbf{a}_0 \\
\dot{\mathbf{a}}_1 &= \frac{1}{2}(t-t_0)^2\mathbf{a}^{(3)} + (t-t_0)\mathbf{a}^{(2)} + \dot{\mathbf{a}}_0 .
\end{aligned} \tag{6}$$

Durch Gleichsetzen der beiden Ansätze erhalten wir die sogenannte Hermite-Interpolation für die zweite und dritte Ableitung der Beschleunigung:

$$\begin{aligned}
\frac{1}{2}\mathbf{a}^{(2)} &= -3\frac{\mathbf{a}_0 - \mathbf{a}_1}{(t-t_0)^2} - \frac{2\dot{\mathbf{a}}_0 + \dot{\mathbf{a}}_1}{(t-t_0)} \\
\frac{1}{6}\mathbf{a}^{(3)} &= 2\frac{\mathbf{a}_0 - \mathbf{a}_1}{(t-t_0)^3} + \frac{\dot{\mathbf{a}}_0 + \dot{\mathbf{a}}_1}{(t-t_0)^2} ,
\end{aligned} \tag{7}$$

Mit deren Hilfe kann nun die "prediction" für $\mathbf{x}$ und $\mathbf{v}$ korrigiert werden auf höhere Ordnung:

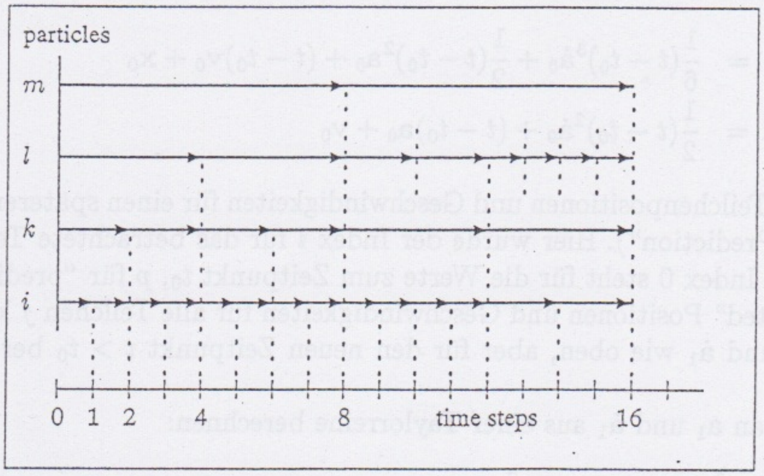
$$\begin{aligned}
\mathbf{x}_c(t) &= \mathbf{x}_p(t) + \frac{1}{24}(t-t_0)^4\mathbf{a}^{(2)} + \frac{1}{120}(t-t_0)^5\mathbf{a}^{(3)} \\
\mathbf{v}_c(t) &= \mathbf{v}_p(t) + \frac{1}{6}(t-t_0)^3\mathbf{a}^{(2)} + \frac{1}{24}(t-t_0)^4\mathbf{a}^{(3)} .
\end{aligned} \tag{8}$$

Zur Vermeidung von Asymmetrien der paarweisen Kräfte sollte hier darauf geachtet werden, zunächst $\mathbf{x}_c$ und $\mathbf{v}_c$ für alle Teilchen zu berechnen, bevor die Positionen und Geschwindigkeiten aktualisiert werden für den nächsten Zeitschritt. Nachdem der Vorgang für alle Teilchen durchgeführt wurde, kann ein neuer Zeitschritt wieder wie oben

Individual Timesteps

Emil
Khalisi

would determine the time–step of force calculation for the whole rest of the system. However, bodies in regions where the variation in force is relatively small, a permanent recomputing of the force terms is time consuming, so in order to economize the calculation, these objects shall be allowed to move a longer distance before a recomputation becomes necessary. This is the idea of a vital method for assigning different time–steps, $\Delta t = t_1 - t_0$, between the integrations, the so–called "individual time–step scheme" (Aarseth 1963 [1]).



Each particle is assigned its own Δt_i which is illustrated in Figure *. The particle named i has the smallest time step at the beginning, so its phase space cooditates are determined at each time step. The time step of k is twice as large as i's, and its coordinates are just extrapolated ("predicted") at the odd time steps, while a full force calculation is due at the dotted times. The step width may or not be altered after the end of the integration cycle for the special particle, as demonstrated for k and l beyond the label "8". The time steps have to stay commensurable with each other.

As a first estimate, the rate of change of the accelaration seems to be a reasonable quantity for the choice of the time step: $\Delta t_i \propto \sqrt{a_i/\dot{a}_i}$. But it turns out that for special situations in a many-body system, it provides some undesired numerical errors. After some experimentation, the following formula was adopted (Aarseth 1985 [2]):

$$\Delta t_i = \sqrt{\eta \frac{|a_{1,i}||a_{1,i}^{(2)}| + |\dot{a}_{1,i}|^2}{|\dot{a}_{1,i}||a_{1,i}^{(3)}| + |a_{1,i}^{(2)}|^2}}, \qquad (11)$$

where η is a dimensionless accuracy parameter which controls the error. In most applications it is taken to be $\eta \approx 0.02$ to 0.04, see also next paragraph. In the running code, the time–steps are adjusted to their appropriate values fairly quickly. Although successive steps normally change smoothly, it is prudent to restrict the growth by a stability factor of 1.2.

1.3 The Ahmad–Cohen scheme

The computation of the full force for each particle in the system makes simulations very time–consuming for large memberships. It is therefore desirable to design a method in order to speed up the calculations while retaining the collisional approach. One way to

Neighbour Scheme

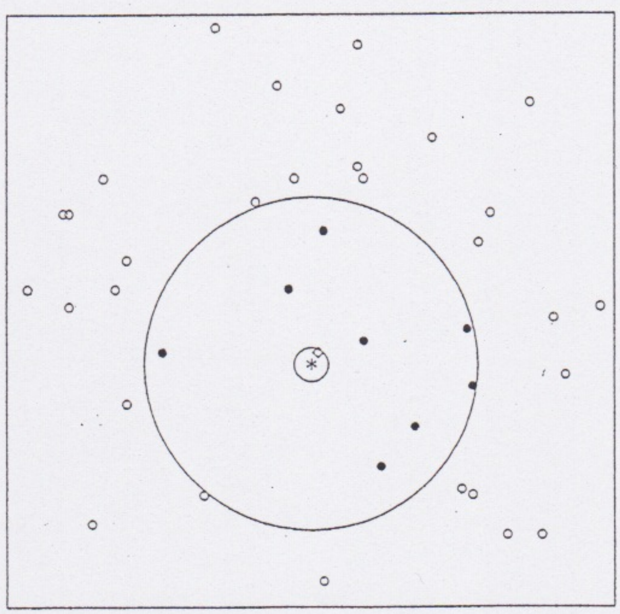
Emil Khalisi

achieve this is to employ a "neighbour scheme", suggested by Ahmad & Cohen (1973, [6]).

The basic idea is to split the force polynomial on a given particle i into two parts, an irregular and a regular component:

$$a_i = a_{i,\mathrm{irr}} + a_{i,\mathrm{reg}}. \tag{12}$$

The irregular acceleration $a_{i,\mathrm{irr}}$ results from particles in a certain neighbourhood of i. They give rise to a stronger fluctuating gravitational force, so it is determined more frequently than the regular one of the more distant particles that do not change their relative distance to i so quickly. We can replace the full summation in eq. (1) by a sum over the n nearest particles for $a_{i,\mathrm{irr}}$ and add a distant contribution from all the other. Whether a particle is a neighbour or not is determined by its distance; all members inside a specified sphere ("neighbour sphere") are held in a list which is modified at the end of each "regular time–step" when a total force summation is carried out. In addition, approaching particles within a surrounding shell satisfying $\mathrm{R} \cdot \mathrm{V} < 0$ are included. This "buffer zone" serves to identify fast approaching particles before they penetrate too far inside the neighbour sphere.



Figures 1.3 and 1.3 show how the Ahmad–Cohen scheme works for one particle (Makino & Aarseth 1992 [28]). At the beginning of the force calculation, a list of neighbour objects around the particle i is created first (filled dots). From this neighbour list the irregular component $a_{i,\mathrm{irr}}$ is calculated, and then the summation is continued to the distant particles obtaining $a_{i,\mathrm{reg}}$. At the same time we also calculate the first time derivative. From the equations (5) and (6) the position and velocity of the particle i are predicted. At time $t_{1,\mathrm{irr}}$ we apply the "corrector" only for $a_{i,\mathrm{irr}}$ from the neighbours; the regular component we do not correct but obtain by extrapolating $a_{i,\mathrm{reg}}$. At the next step, $t_{2,\mathrm{irr}}$, the same predictor–corrector method proceeds for the neighbour particles, while the correction of the distant acceleration term is still neglected. When t_1 is reached, the total force is calculated on the basis of the full application of the

References

Aarseth, S.J. [1972], 'Direct integration methods for the N-body problem', in *Gravitational N-Body Problem* ed. M. Lecar (D. Reidel), 373–87.

Aarseth, S.J. [1985], 'Direct methods for N-body simulations', in *Multiple Time Scales* ed. J.U. Brackbill & B.I. Cohen (Academic Press), 377–418.

Aarseth, S.J. [1999a], 'From NBODY1 to NBODY6: the growth of an industry', *Publ. Astron. Soc. Pac.*, **111**, 1333–46.

Aarseth, S.J. [1999b], 'Star Cluster Simulations: the State of the Art', *Cel. Mech. Dyn. Ast.* **73**, 127.

Aarseth, S.J. [2001a], 'NBODY2: a direct N-body integration code', *New Astron.* **6**, 277–91.

Aarseth, S.J. [2001b], 'Regularization methods for the N-body problem', in *The Restless Universe*, ed. B.A. Steves & A.J. Maciejewski (Inst. Phys. Publ.), 93–108.

Aarseth, S.J. [2003], *Gravitational N-Body Simulations* (Cambridge University Press).

Aarseth, S.J. & Zare, K. [1974], 'A regularization of the three-body problem', *Celes. Mech.* **10**, 185–205.

Ahmad, A. & Cohen, L. [1973], 'A numerical integration scheme for the N-body gravitational problem', *J. Comput. Phys.* **12**, 389–402.

Bulirsch, R. & Stoer, J. [1966], 'Numerical treatment of ordinary differential equations by extrapolation methods', *Num. Math.* **8**, 1–13.

Dorband, E.N., Hemsendorf, M., Merritt, D. [2003], 'Systolic and hyper-systolic algorithms for the gravitational N-body problem, with an application to Brownian motion', *J. Comput. Phys.* **185**, 484–511.

Eggleton, P.P., Fitchett, M.J. & Tout, C.A. [1989], 'The distribution of visual binaries with two bright components', *Astrophys. J.* **347**, 998-1012. (Also see Errata in *Astrophys. J.* **354**, 387.)

Glaschke, P. [2006], 'Studying the Formation of Protoplanets : A new Hybrid Code for Planetesimal Dynamics', Promotion Univ. Heidelberg, `http://www.ub.uni-heidelberg.de/archiv/6553`.

Hansen, B.M.S. & Phinney, E.S. [1997], 'The pulsar kick velocity distribution', *Mon. Not. R. Astron. Soc.* **291**, 569–77.

Harfst, S., Gualandris, A., Merritt, D., Spurzem, R., Portegies Zwart, S., Berczik, P. [2006], ' Performance Analysis of Direct N-body Algorithms on Special-Purpose Supercomputers', *New Astron.* , in press.

Heggie, D.C. [1974], 'A global regularisation of the gravitational N-body problem', *Celes. Mech.* **10**, 217–41.

Heggie, D.C. & Ramamani, N. [1995], 'Approximate self-consistent models for tidally truncated star clusters', *Mon. Not. R. Astron. Soc.* **272**, 317–22.

von Hoerner, S. [1960], 'Die numerische Integration des n-Körper-Problemes für Sternhaufen', *Zs. f. Astroph.*, **50**, 184–214.

von Hoerner, S. [1963], 'Die numerische Integration des n-Körper-Problemes für Sternhaufeni II', *Zs. f. Astroph.*, **57**, 47–82.

von Hoerner, S. [2001], 'How it all started'. In: Dynamics of Star Clusters and the Milky Way, ASP Conference Series, Vol. 228. Edited by S. Deiters, B. Fuchs, R. Spurzem, A. Just, and R. Wielen. San Francisco: Astronomical Society of the Pacific. p.11.

Hurley, J.R., Pols, O.R. & Tout, C.A. [2000], 'Comprehensive analytical formulae for stellar evolution as a function of mass and metallicity', *Mon. Not. R. Astron. Soc.* **315**, 543–69.

Khalisi, E., Spurzem, R.. [2006] 'NBODY6++ - Features of the computer code', `ftp://ftp.ari.uni-heidelberg.de/pub/staff/spurzem/nb6mpi/nb6++manual-new.pdf`

Khalisi, E., Omarov, C., Spurzem, R., Giersz, M., Lin, D., [2003] 'Collisional dynamics of black holes, star clusters and galactic nuclei'. In:Krause, E., Jaeger, W., Resch, M. (Eds.), High Performance Computing in Science and Engineering'03. Springer Verlag, pp. 71-87.

Kokubo, E., Makino, J. [2004], 'A Modified Hermite Integrator for Planetary Dynamics', *Publ. Astron. Soc. Japan* **56**, 861.

Kustaanheimo, P. & Stiefel, E. [1965], 'Perturbation theory of Kepler motion based on spinor regularization', *J. Reine Angew. Math.* **218**, 204–19.

Kroupa, P., Tout, C.A. & Gilmore, G. [1993], 'The distribution of low-mass stars in the Galactic disc', *Mon. Not. R. Astron. Soc.* **262**, 545–87.

Kroupa, P., Aarseth, S.J. & Hurley, J. [2001], 'The formation of a bound star cluster: from the Orion Cluster to the Pleiades', *Mon. Not. R. Astron. Soc.* **321**, 699–712.

Makino, J. [1991], 'Optimal order and time-step criterion for Aarseth-type N-body integrators', *Astrophys. J.* **369**, 200–12.

Makino, J. [1991], 'A Modified Aarseth Code for GRAPE and Vector Processors', *Publ. Astron. Soc. Japan* , **43**, 859.

Makino, J. & Aarseth, S.J. [1992], 'On a Hermite integrator with Ahmad–Cohen scheme for gravitational many-body problems', *Publ. Astron. Soc. Japan* **44**, 141–51.

Makino, J. & Hut, P. [1988], 'Performance Analysis of direct N-body calculations', *Astrophys. J. Suppl.* , **68**, 833.

Makino, J. & Hut, P., Kaplan, M., Saygın, H. [2006], 'A time-symmetric block time-step algorithm for N-body simulations', *New Astron.* **12**, 124–133.

Makino, J. & Taiji, M. [1998], 'Scientific Simulations with Special Purpose Computers', Wiley, Chichester.

Mardling, R.A. [2003], 'A new three-body formalism', in preparation.

Mardling, R.A. & Aarseth, S.J. [1999], 'Dynamics and stability of three-body systems', in *The Dynamics of Small Bodies in the Solar System*, ed. B.A. Steves & A. Roy (Kluwer), 385–92.

Mardling, R.A. & Aarseth, S.J. [2001], 'Tidal interactions in star cluster simulations', *Mon. Not. R. Astron. Soc.* **321**, 398–420.

Mikkola, S. [1985], 'A practical and regular formulation of the N-body equations', *Mon. Not. R. Astron. Soc.* **215**, 171–7.

Mikkola, S. and Aarseth, S.J. [1990], 'A chain regularization method for the few-body problem', *Celes. Mech. Dyn. Ast.* **47**, 375–90.

Mikkola, S. & Aarseth, S.J. [1993], 'An implementation of N-body chain regularization', *Celes. Mech. Dyn. Ast.* **57**, 439–59.

Mikkola, S. & Aarseth, S.J. [1996], 'A slow-down treatment for close binaries', *Celes. Mech. Dyn. Ast.* **64**, 197–208.

Mikkola, S. & Aarseth, S.J. [1998], 'An efficient integration method for binaries in N-body simulations', *New Astron.* **3**, 309–20.

Mikkola, S. & Aarseth, S.J. [2001], 'A time transformed leap-frog scheme', *Cel. Mech. Dyn. Ast.* **84**, 343–354.

Miyamoto, M. & Nagai, R. [1975], 'Three-dimensional models for the distribution of mass in galaxies', *Publ. Astron. Soc. Japan* **27**, 533–43.

Nitadori, K., Makino, J., Hut, P. [2006], ' Performance Tuning of N-Body Codes on Modern Microprocessors: I. Direct Integration with a Hermite Scheme on x86_64 architecture', *New Astron.* , **12**, 169.

Spurzem, R. [1999], 'Direct N-body simulations', *Journ. Comp. Appl. Math.*, **109**, 407–432.

Spurzem, R., Baumgardt, H. & Ibold, N. [2001], 'A parallel implementation of an N-body integrator on general and special-purpose computers', unfinished preprint at `ftp.ari.uni-heidelberg.de/pub/staff/spurzem/edinpaper.ps.gz`

Tout, C.A., Aarseth, S.J., Pols, O. & Eggleton, P. [1997], 'Rapid binary star evolution for N-body simulations and population synthesis', *Mon. Not. R. Astron. Soc.* **291**, 732–48.

beginnen. Es ist außerdem leicht möglich, individuelle Zeitschritte einzuführen, d.h. die Korrektur an einem gegebenen Zeitpunkt nur für eine Teilmenge der Teilchen durchzuführen. Das Verfahren ist jedoch nicht symplektisch, d.h. seine Lösungen sind nicht darstellbar als Lösungen der kanonischen Gleichungen einer Hamilton-Funktion, die sich nur leicht von der des gegebenen Problems unterscheidet. Dadurch weist dieses Verfahren im allgemeinen immer eine leichte Drift von Erhaltungsgrößen wie der Energie auf. Gegenüber dem "Leap Frog" muß die Summe über alle Teilchen hier zweimal durchgeführt werden ($\mathbf{a}$ und $\dot{\mathbf{a}}$), was besonders bei größeren Teilchenzahlen schnell wachsenden Rechenzeitbedarf bedeutet.

3 Das iterierte, zeitumkehrbare Hermite-Verfahren

Makino (1997) hat erkannt, daß das Hermite-Verfahren verbessert werden kann durch den Verzicht auf den höchsten Term der Korrektur von $\mathbf{x}$ und eine iterierbare, zeitumkehrbare Form der Korrektur. Nach Weglassen des höchsten Terms für $\mathbf{x}_c$ in Gleichung (8) bekommen wir durch zunächst identische Umformung:

$$
\begin{aligned}
\mathbf{v}_c(t) &= \mathbf{v}_0(t) + \frac{1}{2}(t-t_0)\Big(\mathbf{a}_1 + \mathbf{a}_0\Big) - \frac{1}{12}(t-t_0)^2\Big(\dot{\mathbf{a}}_1 - \dot{\mathbf{a}}_0\Big) \\
\mathbf{x}_c(t) &= \mathbf{x}_0(t) + \frac{1}{2}(t-t_0)\Big(\mathbf{v}_c + \mathbf{v}_0\Big) - \frac{1}{12}(t-t_0)^2\Big(\mathbf{a}_1 - \mathbf{a}_0\Big)\,. \qquad (9)
\end{aligned}
$$

Algorithmisch verhält sich dieses Verfahren aber völlig anders als das normale Hermite-Verfahren (selbst wenn man dieses iterieren würde), da zuerst $\mathbf{v}_c$ berechnet wird und schon im gleichen Schritt für $\mathbf{x}_c$ eingesetzt wird. Dadurch wird die Konvergenz beschleunigt, aber auch eine Zeitsymmetrie des Algorithmus erreicht (diese ist auch beim Leap-Frog vorhanden, nicht aber beim normalen Hermite-Verfahren). Mit den neuen Werten für $\mathbf{x}_c$ und $\mathbf{v}_c$ können $\mathbf{a}_1$ und $\dot{\mathbf{a}}_1$ erneut berechnet und so neue Werte für $\mathbf{x}_c$ und $\mathbf{v}_c$ berechnet werden. Schon nach zwei Iterationen zeigt sich eine erhebliche Verbesserung der Eigenschaften des Integrators, vor allem für kleine Teilchenzahlen.

Der numerische Aufwand steigt durch jede Iteration, die zu einer erneuten Berechnung von $\mathbf{a}$ und $\dot{\mathbf{a}}$ für alle anderen Teilchen führt. Die Berechnung von $\mathbf{a}^{(2)}$ und $\mathbf{a}^{(3)}$ nach dem Hermite-Verfahren Gleichung (7) ist strenggenommen nicht mehr nötig. Da bei hohen Teilchenzahlen jedoch der Aufwand dafür gering ist, werden diese gerne zusätzlich berechnet, um die bessere Anpassungsformel für den Zeitschritt Gleichung (10) (siehe unten) verwenden zu können.

Literatur

C.L. Siegel, Himmelsmechanik

K. Stumpff, Himmelsmechanik I-III

Makino, J. et al., Astrophysical Journal Vol. 480, p. 432 (1997)

Makino, J., Aarseth, S.J., Proc. Astron. Soc. Japan Vol. 44, p. 141 (1992)

Aarseth, S.J., Celest. Mechanics and Dyn. Astron. Vol. 73, p. 127 (1999)

```
vvec:=array(1..4);
levi:=array(1..4,1..4);
with(linalg,multiply);
multiply(levi,vvec);

levi:=[[ 1, i, j,-k],
       [ i,-1, k, j],
       [ j,-k,-1,-i],
       [ k, j,-i, 1]];

with(linalg,transpose);lt:=transpose(levi);
with(linalg,multiply);multiply(lt,levi);
leviinv:=array(1..4,1..4);
with(linalg,inverse);
leviinv:=inverse(levi);

with(linalg,multiply);
multiply(levi,leviinv);

a:=a1*el+a2*il+a3*jl+a4*kl;
b:=b1*er+b2*ir+b3*jr+b4*kr;
x0:=expand(a*b);

x1:=expand(algsubs( el*er=levi[1,1], x0));
x1:=expand(algsubs( el*ir=levi[1,2], x1));
x1:=expand(algsubs( el*jr=levi[1,3], x1));
x1:=expand(algsubs( el*kr=levi[1,4], x1));

x1:=expand(algsubs( il*er=levi[2,1], x1));
x1:=expand(algsubs( il*ir=levi[2,2], x1));
x1:=expand(algsubs( il*jr=levi[2,3], x1));
x1:=expand(algsubs( il*kr=levi[2,4], x1));

x1:=expand(algsubs( jl*er=levi[3,1], x1));
x1:=expand(algsubs( jl*ir=levi[3,2], x1));
x1:=expand(algsubs( jl*jr=levi[3,3], x1));
x1:=expand(algsubs( jl*kr=levi[3,4], x1));

x1:=expand(algsubs( kl*er=levi[4,1], x1));
x1:=expand(algsubs( kl*ir=levi[4,2], x1));
x1:=expand(algsubs( kl*jr=levi[4,3], x1));
x1:=expand(algsubs( kl*kr=levi[4,4], x1));

x3:=algsubs(b1=a1,x1);
x3:=algsubs(b2=a2,x3);
x3:=algsubs(b3=a3,x3);
x3:=algsubs(b4=a4,x3);

yy:=sort(x3,[k,j,i]);

cc1:=coeff(yy,i);
cc2:=coeff(yy,j);
cc3:=coeff(yy,k);
cc0:=expand(yy-cc1*i-cc2*j-cc3*k);

ccc:=expand(cc0^2+cc1^2+cc2^2+cc3^2);

expand((a1^2+a2^2+a3^2+a4^2)^2-ccc);
```

```
vvec:=array(1..4);
levi:=array(1..4,1..4);
with(linalg,multiply);
multiply(levi,vvec);

levi:=[[ 1, i, j, k],
       [ i,-1, k,-j],
       [ j,-k,-1, i],
       [ k, j,-i,-1]];

leviinv:=array(1..4,1..4);
with(linalg,inverse);
leviinv:=inverse(levi);

with(linalg,multiply);
multiply(levi,leviinv);

a:=a1*el+a2*il+a3*jl+a4*kl;
b:=b1*er+b2*ir+b3*jr+b4*kr;
x0:=expand(a*b);

x1:=expand(algsubs( el*er=levi[1,1], x0));
x1:=expand(algsubs( el*ir=levi[1,2], x1));
x1:=expand(algsubs( el*jr=levi[1,3], x1));
x1:=expand(algsubs( el*kr=levi[1,4], x1));

x1:=expand(algsubs( il*er=levi[2,1], x1));
x1:=expand(algsubs( il*ir=levi[2,2], x1));
x1:=expand(algsubs( il*jr=levi[2,3], x1));
x1:=expand(algsubs( il*kr=levi[2,4], x1));

x1:=expand(algsubs( jl*er=levi[3,1], x1));
x1:=expand(algsubs( jl*ir=levi[3,2], x1));
x1:=expand(algsubs( jl*jr=levi[3,3], x1));
x1:=expand(algsubs( jl*kr=levi[3,4], x1));

x1:=expand(algsubs( kl*er=levi[4,1], x1));
x1:=expand(algsubs( kl*ir=levi[4,2], x1));
x1:=expand(algsubs( kl*jr=levi[4,3], x1));
x1:=expand(algsubs( kl*kr=levi[4,4], x1));

x3:=algsubs(b1=a1,x1);
x3:=algsubs(b2=a2,x3);
x3:=algsubs(b3=a3,x3);
x3:=algsubs(b4=a4,x3);

yy:=sort(x3,[k,j,i]);

cc1:=coeff(yy,i);
cc2:=coeff(yy,j);
cc3:=coeff(yy,k);
cc0:=expand(yy-cc1*i-cc2*j-cc3*k);

ccc:=expand(cc0^2+cc1^2+cc2^2+cc3^2);

expand((a1^2+a2^2+a3^2+a4^2)^2-ccc);
```