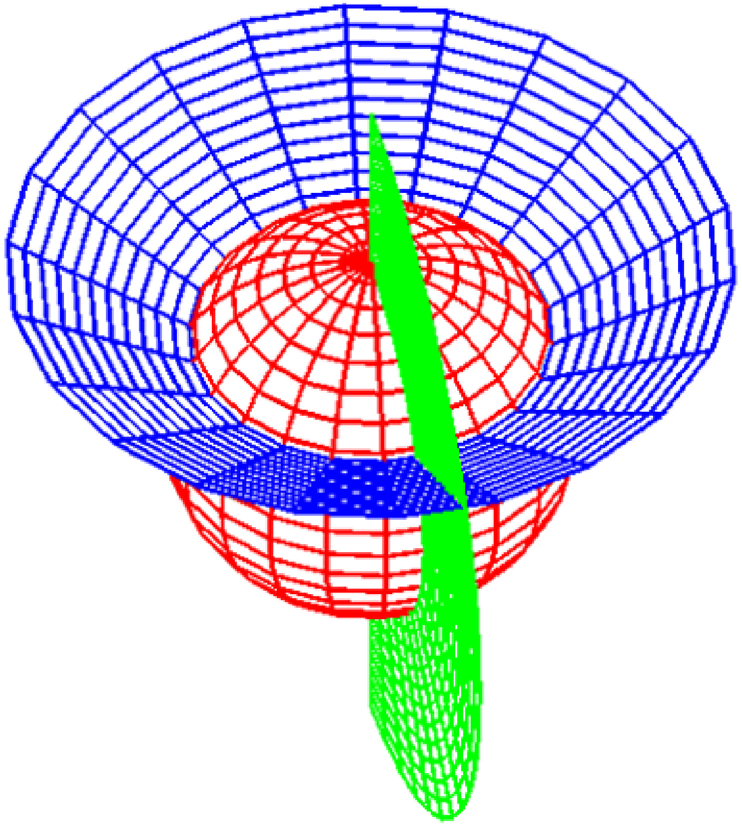
Computing separation variables on a computer

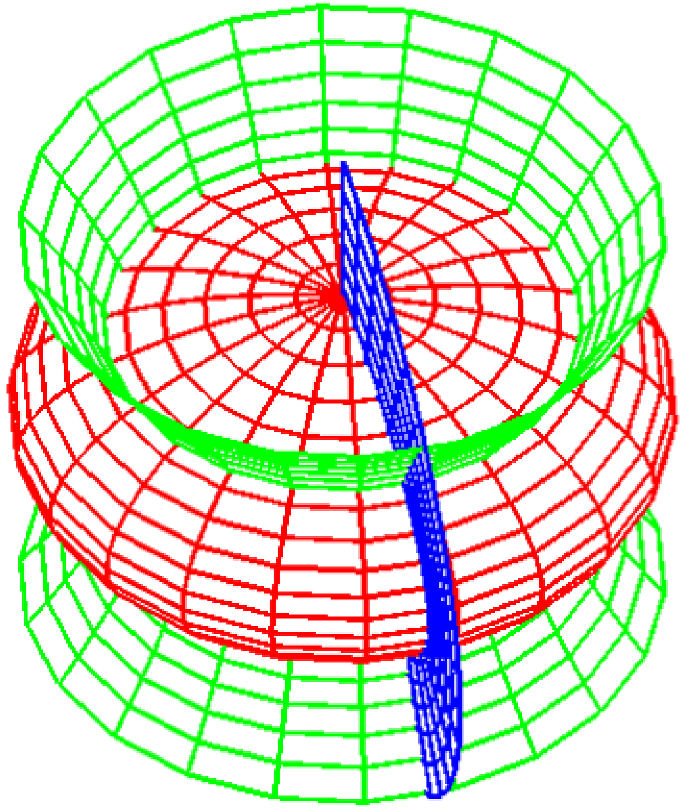
Yury Grigoryev

Orthogonal coordinate systems

Different coordinate systems are convenient for different problems.
For example, spherical coordinate system is convenient for a problem of a particle in a central force field, while oblate spheroidal system can be convenient for the Jacobi-Calogero inverse square model.



spherical



oblate spheroidal

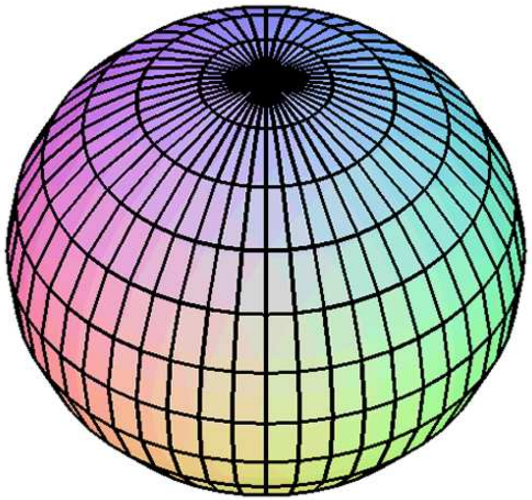
Using separation variables in numerical simulations

- Reduces computation time
- Easier parallelizing
- Easier control of the correctness through integrals of motion

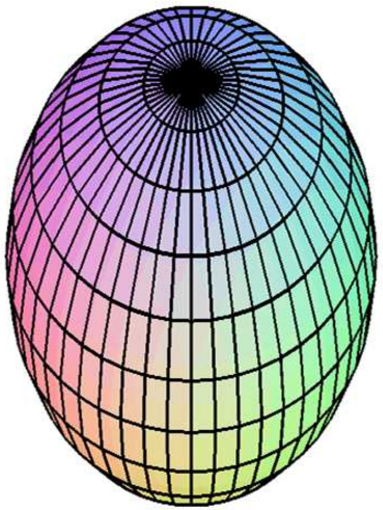
Theoretical overview

$\mathcal{Q}$ - riemannian manifold with local coordinates $q = (q^1, q^2, \ldots, q^n)$ and positive definite metric tensor $\mathbf{G}$.

Examples:



sphere



ellipsoid

Natural Hamilton function

$$H = T(p,q) + V(q) = \sum_{i,j=1}^{n} \mathsf{g}^{ij}(q) p_i p_j + V(q)$$

Separation of variables for the Hamilton-Jacobi equation

Hamilton-Jacobi equation	$H(p,q) = E$
Additive Action	$\mathcal{S}(Q,\alpha_1,\ldots,\alpha_n) \equiv \sum_{i=1}^{n} \mathcal{S}_i(Q_i,\alpha_1,\ldots,\alpha_n)$
Conjugated momenta	$P_i = \dfrac{\partial \mathcal{S}_i(Q_i,\alpha_1,\ldots,\alpha_n)}{\partial Q_i}$
Separated equations	$\Phi_i(Q_i, P_i,\, \alpha_1,\ldots,\alpha_n) = 0$

Levi-Civita criteria

The Hamilton-Jacobi equation admits separation of variables if the Hamilton function $H(P,Q)$ satisfies the following $n(n-1)/2$ equations with $i \neq j$

$$\frac{\partial H}{\partial P_j}\frac{\partial H}{\partial P_i}\frac{\partial^2 H}{\partial Q_j \partial Q_i} - \frac{\partial H}{\partial P_j}\frac{\partial H}{\partial Q_i}\frac{\partial^2 H}{\partial Q_j \partial P_i}$$

$$-\frac{\partial H}{\partial Q_j}\frac{\partial H}{\partial P_i}\frac{\partial^2 H}{\partial P_j \partial Q_i} + \frac{\partial H}{\partial Q_j}\frac{\partial H}{\partial Q_i}\frac{\partial^2 H}{\partial P_j \partial P_i} = 0.$$

Separating variables with coordinate transformations

Point transformations:

$$Q = f(q), \qquad P = f'(q)\, p\,, \qquad f = (f_1, \ldots, f_n)$$

The Hamiltonian is covariant w.r.t point transformations

$$H = T(P, Q) + V(Q) = \sum_{i,j=1}^{n} \mathsf{g}^{ij}(Q) P_i P_j + V(Q).$$

Killing Tensors

$\mathcal{Q}$ - riemannian manifold with a metric tensor

$$\mathbf{G} = \mathsf{g}^{ij}\partial_i \otimes \partial_j, \qquad \partial_k \equiv \frac{\partial}{\partial q^k}$$

The correspondence between tensor $\mathbf{K}$ and polynomial P_K:

$$\mathbf{K} = (K^{i\ldots j}) \qquad \longleftrightarrow \qquad P_K = K^{i\ldots j} p_i \cdots p_j \,.$$

Definition 1 *Killing Tensor* $\mathbf{K}$ *of rank* ℓ *is a symmetric* $(\ell, 0)$ *tensor in the space* $\mathcal{Q}$ *satisfying tensor Killing equation*

$$[\mathbf{K}, \mathbf{G}] = 0 \qquad \Longleftrightarrow \qquad \{P_K, P_G\} = 0 \,.$$

L-tensors

Definition 2 *Conformal Killing Tensor* $\mathbf{L}$ *of rank* ℓ *is a symmetric* $(\ell, 0)$ *tensor in the space* $\mathcal{Q}$*, satisfying tensor equation*

$$\{P_L, P_G\} = cP_G\,.$$

Definition 3 *Conformal tensor* $\mathbf{L}$ *with simple eigenvalues and zero Nijenhuis torsion*

$$T^m_{ij} \equiv 2L^\alpha_i \partial_\alpha L^m_j - 2L^m_\alpha \partial_i L^\alpha_j = 0\,.$$

is called L*-tensor, or Benenti tensor.*

Separating variables for L-systems

Theorem 1 [Benenti] *Eigenvalues* Q_i *of tensor* $\mathbf{L}$ *are separation variables for the integrable system.*

Integrable systems admitting separation in such variables are called L-systems, or Benenti systems.

For given $\mathbf{L}$ we can define a recursion operator $\mathbf{N}$, i.e. canonical lifting of tensor $\mathbf{L}$ on a cotangent bundle $T^*\mathcal{Q}$ according to the rule

$$\mathbf{N}\frac{\partial}{\partial q^k} = \sum_{i=1}^{n} L^i_k \frac{\partial}{\partial q^i} + \sum_{ij} p_j \left(\frac{\partial L^j_i}{\partial q^k} - \frac{\partial L^j_k}{\partial q^i} \right) \frac{\partial}{\partial p_i},$$

$$\mathbf{N}\frac{\partial}{\partial p_k} = \sum_{i=1}^{n} L^k_i \frac{\partial}{\partial p_i}.$$

Recurrent equations for the integrals of motion H_m:

$$dH_{m+1} = \mathbf{N}^* dH_m + \sigma_{m+1} dH, \qquad m = 1, \ldots, n-1, \qquad H_n \equiv 0\,.$$

Here σ_m are the coefficients of the characteristic polynomial of $\mathbf{L}$: $\det(\lambda \mathrm{I} - \mathbf{L}) = \sum_{m=0}^{n} \sigma_m \lambda^{n-m}$

Algorithm of calculating separation variables

1. Build and solve equations for L

$$d(\mathtt{i}_{X_T} d\,\theta - T d\sigma_1) = 0$$

$$d(\mathtt{i}_{X_V} d\,\theta - V d\sigma_1) = 0$$

   Here $\mathtt{i}$ is a hook operator, $\sigma_1 = \mathrm{tr}\,\mathbf{L}$ and $\theta = \sum_{i,j=1}^{n} L^i_j\, p_i dq^j$ is an L-deformation of standard Liouville 1-form $\theta_0 = \sum p_j dq^j$.

2. Calculate eigenvalues of L - separation variables

3. Solve a chain of equations $dH_{m+1} = \mathbf{N}^* dH_m + \sigma_{m+1} dH$ for the integrals of motion.

Implementation (Maple worksheet)

Applying the programme to some integrable systems

1. Neumann system The Neumann system describes the motion of a particle on a sphere under the influence of a quadratic potential $V(x) = \frac{1}{2}\sum_{i=1}^{N+1} \alpha_i x_i^2$, where $\alpha_i \neq \alpha_j$

Spheroconical coordinates: $\sum_{i=1}^{N+1} \frac{x_i^2}{\lambda - \alpha_i} = 0$

Manual transformation of variables

$$X_i = x_i^2, i = 1, .., N$$

$$T = 2\sum_a X_a(1 - X_a)Y_a^2 - 4\sum_{a<b} X_a X_b Y_a Y_b$$

$$V = \frac{1}{2}\sum_a (\alpha_a - \alpha_{N+1})X_a$$

2. Holt system

$$H(x) = \frac{1}{2}({p_X}^2 + {p_Y}^2) + aX^{-\frac{2}{3}}(\frac{3b}{4}X^2 + Y^2 + c)$$

Manual transformation of variables

$$a \to 4(\frac{3}{2})^{1/3}a, \quad c \to \frac{c}{3a}$$

$$X = \frac{2}{3}x^{3/2}, \quad p_X = p_x/\sqrt{x}$$

$$Y = -\frac{1}{2\sqrt{3a}}p_y \quad p_Y = 2\sqrt{3a}y$$

$$H = \frac{p_x^2 + p_y^2}{2x} + 2a(bx^2 + 3y^2) + \frac{2c}{x}$$

Ways of improving the performance of the programme

- Watching after and modifying the intermediate results of the programme

- Using special PDE solvers for overdetermined systems

Ways of broadening the class of systems separable by the programme

- Performing manual transformations of variables
- Solving the systems for $\mathbf{N}$ instead of those for $\mathbf{L}$
- Adding typical non-point transformations of variables with arbitrary parameters to the equations for L