Geometric Integration and Lie group methods, THREAD
Network-wide training event 6

February 5, 2021

FOURTH ASSIGNMENT. This is the first assignment of the second part of the course. We consider here the Euler free rigid body equations

$$\dot{m} = -\mathbb{I}^{-1} m \times m. \tag{1}$$

Here m is the angular momentum in body coordinates of the free rigid body, and is modelled as a vector with three components $m \in \mathbb{R}^3$. The *inertia tensor* $\mathbb{I}$ is thought of as 3×3 diagonal matrix

$$\mathbb{I} = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

The "$\times$" is the standard cross product between vectors in $\mathbb{R}^3$ so

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_2 - a_3 b_2 \\ -a_1 b_3 + a_3 b_1 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

This means that (1) can be written in matrix-vector form as

$$\begin{pmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \end{pmatrix} = \begin{pmatrix} 0 & \frac{m_3}{I_3} & -\frac{m_2}{I_2} \\ -\frac{m_3}{I_3} & 0 & \frac{m_1}{I_1} \\ \frac{m_2}{I_2} & -\frac{m_1}{I_1} & 0 \end{pmatrix} \cdot \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = A(m) \cdot m$$

The exercise today is to implement two fourth order Lie group integrators for this problem. An ingredient in the Lie group integrators for this problem is the matrix exponential function that needs to be computed for skew-symmetric 3×3-matrices. so our first goal is to develop our own bespoke method for exponentiating such matrices.

Problem 1. The matrix exponential can be defined as

$$\mathrm{e}^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} \tag{2}$$

There is a formula for this when A is skew-symmetric and 3×3. Because of the skew-symmetry, only 3 of the 9 elements in the matrix can be chosen freely, a very standard parametrisation is, using a vector $a = [a_1, a_2, a_3]^T$

$$A = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

Define $\alpha = \|a\| = \sqrt{a_1^2 + a_2^2 + a_3^2}$. Rodrigues formula reads

$$\mathrm{e}^A = I + \frac{\sin\alpha}{\alpha} + \frac{1-\cos\alpha}{\alpha^2}A^2, \quad \text{A skew-symmetric } 3 \times 3$$

(a) Verify this formula on paper. First show that $A^3 = -\alpha^2 A$ (use a CAS if you wish). Then split the infinite sum (2) into even and odd terms and reduce each of these terms (except the first) to be proportional to either A or A^2 and sum up the two series.

(b) Implement the formula in a function `expso3`. Verify the code for some arbitrary 3×3 skew-symmetric matrices by comparing to general built-in functions. In Matlab there is a function `expm` and in Python you can import a function of the same name from `scipy.linalg`.

Remark. Note that both of the functions of α appearing in the formula have removable singularities at $\alpha = 0$, you should account for this, e.g. by including a test and replacing the functions by some truncated Taylor expansion for small α.

Problem 2. We now want to implement the two fourth order methods discussed in the lectures, and we shall apply them to the Euler's free rigid body equations (1). We first remind ourselves how the methods look like.

1. The Runge–Kutta–Munthe–Kaas method of order 4 can be presented as follows. Given an input value m_n take one step with time step size h by computing

$$\begin{aligned} A_1 &= hA(m_n), \\ A_2 &= hA(\exp(\tfrac{1}{2}A_1) \cdot m_n), \\ A_3 &= hA(\exp(\tfrac{1}{2}A_2 - \tfrac{1}{8}[A_1, A_2]) \cdot m_n), \\ A_4 &= hA(\exp(A_3) \cdot m_n), \\ m_{n+1} &= \exp(\tfrac{1}{6}(A_1 + 2A_2 + 2A_3 + A_4 - \tfrac{1}{2}[A_1, A_4])) \cdot m_n. \end{aligned}$$

   Here the bracket is the matrix commutator $[A, B] = AB - BA$.

2. The second method is the commutator-free method of order 4, first presented in [1].

$$
\begin{aligned}
M_1 &= m_n \\
M_2 &= \exp(\tfrac{1}{2}hA_1)\cdot m_n \\
M_3 &= \exp(\tfrac{1}{2}hA_2)\cdot m_n \\
M_4 &= \exp(hA_3 - \tfrac{1}{2}hA_1)\cdot M_2 \\
m_{n+\frac{1}{2}} &= \exp(\tfrac{1}{12}h(3A_1 + 2A_2 + 2A_3 - A_4))\cdot m_n \\
m_{n+1} &= \exp(\tfrac{1}{12}h(-A_1 + 2A_2 + 2A_3 + 3A_4))\cdot m_{n+\frac{1}{2}}
\end{aligned}
$$

Here it is understood that $A_i = A(M_i)$ for every i.

(a) Implement both the two methods presented above. Include in particular calls to the function you made in **(a)**. Try the method out, both for the symmetric case ($I_1 = I_2 \neq I_3$) and all I_i distinct. Check in both cases that the numerical solution m_n preserves the norm $\|m_n\|$ for all n (to machine accuracy).

(b) Try now to solve the problem with a classical integrator, e.g. use some library routine in Matlab or Python with strict tolerance, to calculate a reference solution at your chosen end point. Verify then that the convergence order of both the Lie group integrators is $p = 4$.

References

[1] E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *Future Generation Computer Systems*, 19:341–352, 2003.