BIT
2001, Vol. 41, No. 5, pp. 996–1007

0006-3835/01/4105-0996 $16.00

GEOMETRIC INTEGRATION OF ORDINARY DIFFERENTIAL EQUATIONS ON MANIFOLDS *

E. HAIRER

Section de mathématiques, Université de Genève, 2-4 rue du Lièvre,
CH-1211 Genève 24, Switzerland. email: Ernst.Hairer@math.unige.ch

Abstract.

This article illustrates how classical integration methods for differential equations on manifolds can be modified in order to preserve certain geometric properties of the exact flow. Projection methods as well as integrators based on local coordinates are considered. The central ideas of this article have been presented at the 40th anniversary meeting of the journal BIT.

AMS subject classification: 65L80, 65P10, 34C40.

Key words: Geometric integration, differential equations on manifolds, symmetric methods, projection methods, methods based on local coordinates.

1 Idea of geometric integration.

In this introductory section we consider systems of ordinary differential equations (initial value problems)

$$\dot{y} = f(y), \qquad y(0) = y_0, \tag{1.1}$$

and we study the question to what extent geometric properties of the exact flow $\varphi_t(y_0)$ can be preserved by a numerical approximation.

Example 1.1. Consider the equations for the mathematical pendulum

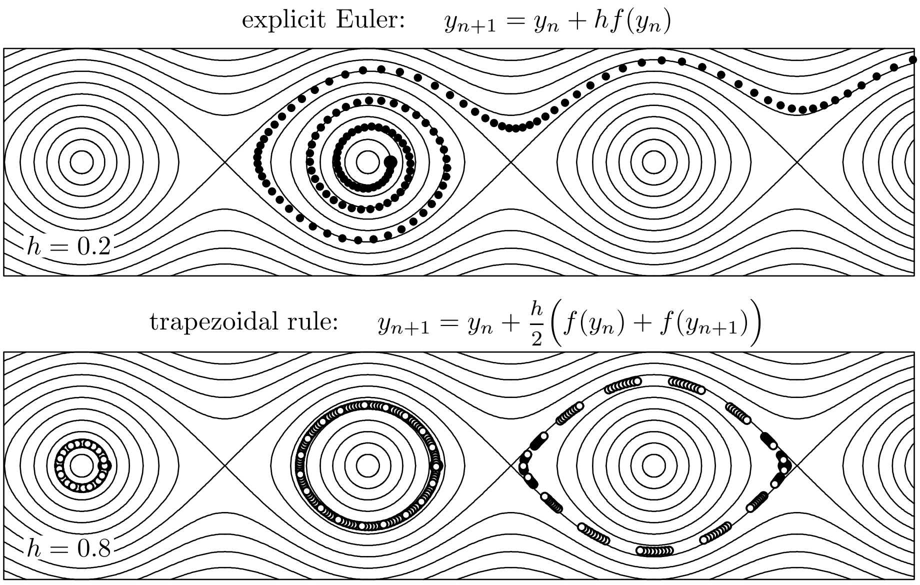
$$\dot{p} = -\sin q, \qquad \dot{q} = p. \tag{1.2}$$

One can check by differentiation that the expression (total energy)

$$H(p,q) = \frac{1}{2}p^2 - \cos q$$

is constant along solutions of (1.2). This means that the solutions remain on the level curves of $H(p,q)$, which are drawn as solid lines in Fig. 1.1. We now apply two numerical methods: the explicit Euler method gives qualitatively wrong approximations (the energy $H(p,q)$ increases), whereas the trapezoidal rule shows a qualitatively correct periodic motion. We therefore call the trapezoidal rule a geometric integrator for this problem.

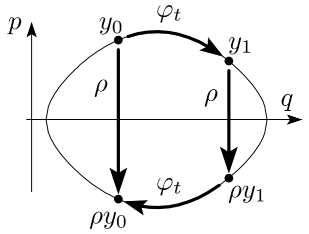
*Received October 2000. Communicated by Gustaf Söderlind.



explicit Euler: $y_{n+1} = y_n + hf(y_n)$

trapezoidal rule: $y_{n+1} = y_n + \frac{h}{2}\Big(f(y_n) + f(y_{n+1})\Big)$

Figure 1.1: Numerical solution (150 steps) for problem (1.2) obtained by Euler's method and by the trapezoidal rule with $h = 0.2$ and $h = 0.8$, respectively. Initial values are $p_0 = 0$, $q_0 = 0.5$ for Euler's method, and $p_0 = 0$, $q_0 \in \{0.5 - 2\pi, 1.5, 2.873 + 2\pi\}$ for the trapezoidal rule. The solid lines represent the exact flow.



Let us give a simple geometric explanation of the numerical phenomenon encountered in Example 1.1. With the reflection

$$\rho(p, q) = (-p, q),$$

the exact flow φ_t of the pendulum equation satisfies

$$\rho \circ \varphi_t = \varphi_{-t} \circ \rho = \varphi_t^{-1} \circ \rho \tag{1.3}$$

(see the small picture to the right, where $y_i = (p_i, q_i)$). This can be checked analytically by differentiation with respect to time, and implies that any solution that crosses twice the horizontal axis has to be periodic. Consider then a numerical method, denoted by $y_{n+1} = \Phi_h(y_n)$. Using the fact that for the problem (1.2) the righthand side $f(y)$ satisfies $f \circ \rho = -\rho \circ f$, one obtains that the Euler method as well as the trapezoidal rule (in fact all Runge–Kutta methods) satisfy $\rho \circ \Phi_h = \Phi_{-h} \circ \rho$, which is the first identity in (1.3). The second relation of (1.3) holds only for *symmetric* or *time-reversible* methods, i.e., methods that satisfy

$$\Phi_h = \Phi_{-h}^{-1}.$$

This means that exchanging $h \leftrightarrow -h$ and $y_n \leftrightarrow y_{n+1}$ leaves the formula unchanged. This is the case for the trapezoidal rule, but not for the explicit Euler method.

Hence, the same argumentation as for the analytic solution makes it plausible that for symmetric methods the numerical solution lies near a closed curve.

Example 1.2 (Toda lattice). The movement of particles interacting pairwise with exponential forces is described by a Hamiltonian system

$$\dot{p} = -\frac{\partial H}{\partial q}(p,q), \qquad \dot{q} = \frac{\partial H}{\partial p}(p,q)$$

with

$$H(p,q) = \sum_{k=1}^{n} \Big(\frac{1}{2}\, p_k^2 + \exp(q_k - q_{k-1}) \Big).$$

We consider periodic boundary conditions $q_0 = q_n$ and $n = 3$. This system has the remarkable property that the eigenvalues of the matrix

$$L = \begin{pmatrix} a_1 & b_1 & b_3 \\ b_1 & a_2 & b_2 \\ b_3 & b_2 & a_3 \end{pmatrix}$$

(where $a_k = -p_k/2$ and $b_k = \frac{1}{2}\exp\big(\frac{1}{2}(q_k - q_{k+1})\big)$) are constant along the solutions (isospectral flow). Moreover, the time-one flow of the system is related to one iteration of the QR method for computing the eigenvalues of a matrix.

In our experiment of Fig. 1.2 we fix initial values as $p_1 = 0$, $p_2 = 1$, $p_3 = 0.5$, $q_1 = 1$, $q_2 = 2$, $q_3 = 4$, and we apply two different methods. Similarly as in the first experiment we observe that the non-symmetric method (here the classical explicit Runge–Kutta method of order 4) cannot conserve the eigenvalues along the numerical solution, but the symmetric method (trapezoidal rule) preserves

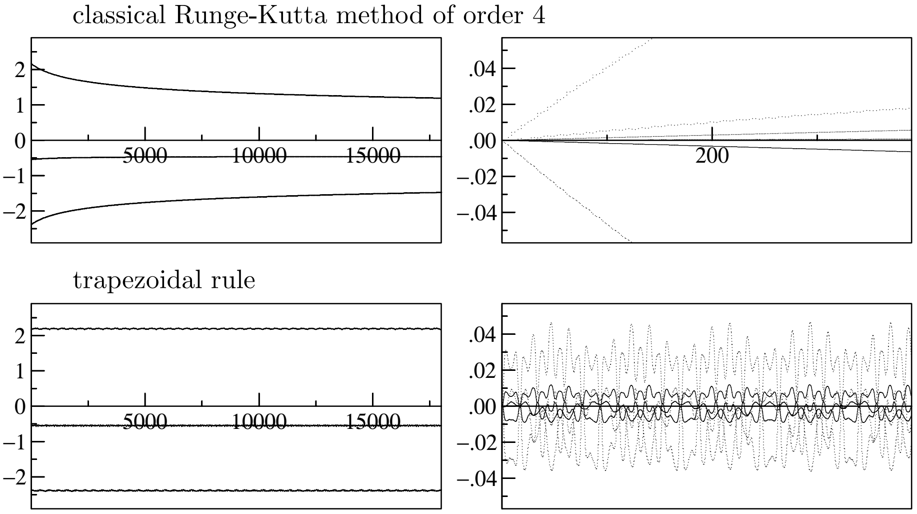


Figure 1.2: Numerically obtained eigenvalues (left) and errors in the eigenvalues (right) for the step sizes $h = 0.1$ (dotted) and $h = 0.05$ (solid line).

them very well with an error that remains bounded as $\mathcal{O}(h^2)$ over very long time intervals. Here, a simple explanation is no longer possible. One needs to combine the techniques of backward error analysis with those of Hamiltonian perturbation theory (see Hairer, Lubich, and Wanner [14]).

The examples above have shown that symmetric methods perform qualitatively better for integrations over long time intervals. Most of the commonly used techniques for solving differential equations on manifolds destroy the symmetry of the underlying method. The aim of this article is to show how the symmetry can be restored.

2 Differential equations on manifolds.

Let $\mathcal{M} \subset \mathbb{R}^n$ be a given manifold. A system of ordinary differential equations

$$\dot{y} = f(y) \quad (y \in \mathbb{R}^n) \tag{2.1}$$

is a *differential equation on the manifold* $\mathcal{M}$, if

$$y_0 \in \mathcal{M} \quad \text{implies} \quad y(t) \in \mathcal{M} \quad \text{for all } t.$$

This is equivalent to the requirement on the vector field that

$$f(y) \in T_y\mathcal{M} \quad \text{for} \quad y \in \mathcal{M},$$

where $T_y\mathcal{M}$ is the tangent space of $\mathcal{M}$ at the point $y \in \mathcal{M}$. Recall that for manifolds given by $\mathcal{M} = \{y \in \mathbb{R}^n \,|\, g(y) = 0\}$, the tangent space takes the form $T_y\mathcal{M} = \{v \in \mathbb{R}^n \,|\, g'(y)v = 0\}$.

Differential equations on manifolds arise in a variety of applications, and their numerical treatment has been the subject of many publications. Let us mention some important situations with a selection of typical references:

- *Differential equations with invariants:* Baumgarte [4], Shampine [24], Ascher, Chin, and Reich [2].
- *Problems on Lie groups:* Crouch and Grossman [6], Munthe-Kaas [19], Diele, Lopez and Peluso [7], Iserles, Munthe-Kaas, Nørsett, and Zanna [16].
- *Differential-algebraic systems:* Gear [9], Rheinboldt [22], Griepentrog and März [10], Hairer, Lubich, and Roche [13], Brenan, Campbell, and Petzold [5], Hairer and Wanner [15], Eich-Soellner and Führer [8].

A naïve approach for the numerical solution of a differential equation on a manifold $\mathcal{M}$ would be to apply a method to the problem (2.1) without taking care of the manifold $\mathcal{M}$, and to hope that the solution stays close to the manifold. This is of course illusory as demonstrated in Figure 2.1 (left picture), where we applied the explicit Euler method (370 steps) with step size $h = 0.1$ to the equation (2.2).

A foremost requirement on a numerical integrator is that the numerical approximation lies exactly on the manifold. But, if the exact flow on the manifold has certain geometric properties, it is natural to ask for numerical methods that preserve them. Let us demonstrate this with the following two examples:

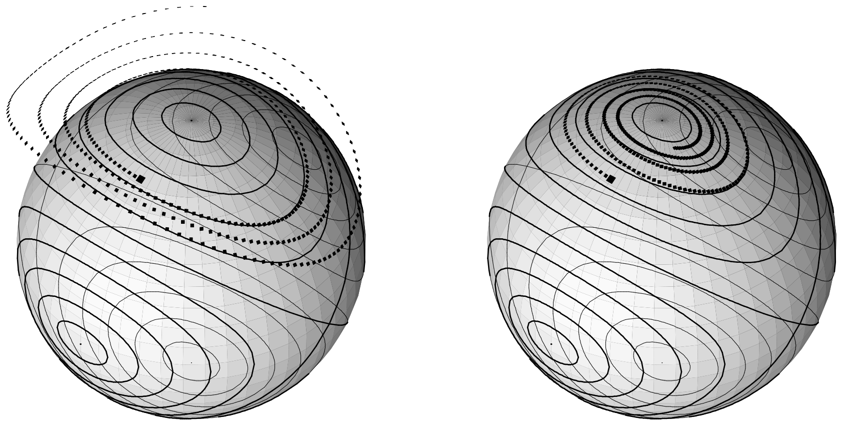


Figure 2.1: Rigid body simulation ($I_1 = 2$, $I_2 = 1$, $I_3 = 2/3$); explicit Euler (left), and explicit Lie–Euler method (right); initial value $y_0 = (\cos(1.1), 0, \sin(1.1))^T$ and step size $h = 0.1$.

Example 2.1 (Rigid body). The movement of a rigid body with mass centered at the origin is described by the Euler equations

$$\begin{pmatrix} \dot y_1 \\ \dot y_2 \\ \dot y_3 \end{pmatrix} = \begin{pmatrix} 0 & y_3/I_3 & -y_2/I_2 \\ -y_3/I_3 & 0 & y_1/I_1 \\ y_2/I_2 & -y_1/I_1 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \tag{2.2}$$

where the vector $y = (y_1, y_2, y_3)^T$ is the angular momentum in the body frame, and I_1, I_2, I_3 are the principal moments of inertia.

This problem is of the form

$$\dot Y = B(Y)Y \quad \text{with skew-symmetric } B(Y). \tag{2.3}$$

One checks by differentiation that (2.3) is a differential equation on the manifold

$$\mathcal{M} = \{Y \mid Y^T Y = \mathit{const}\}$$

(orthogonal Lie group, if Y is matrix-valued).

The most simple Lie group method for the problem (2.3) is the so-called Lie–Euler method

$$y_{n+1} = \exp\bigl(hB(y_n)\bigr)y_n.$$

Since the exponential of a skew-symmetric matrix is orthogonal, the numerical solution stays exactly on the sphere (see the right picture of Figure 2.1).

However, the flow of (2.2) has a further interesting property: it has

$$H(y) = \frac{1}{2}\Bigl(\frac{y_1^2}{I_1} + \frac{y_2^2}{I_2} + \frac{y_3^2}{I_3}\Bigr) \tag{2.4}$$

as conserved quantity. Hence, the solutions of (2.2) lie on the intersection of the sphere and the ellipsoid given by $H(y) = \mathit{const}$ (closed curves in Figure 2.1). The numerical solution of the Lie–Euler method does not mimic this property. From a geometric integrator of (2.2) we would expect that the numerical solution lies on a closed curve.

Example 2.2 (Outer solar system). Consider the N-body problem, which is a Hamiltonian system

$$\dot{p} = -\frac{\partial H}{\partial q}(p,q), \qquad \dot{q} = \frac{\partial H}{\partial p}(p,q),$$

$$H(p,q) = \frac{1}{2}\sum_{i=1}^{N} m_i^{-1}\|p_i\|^2 + \sum_{i<j} V_{ij}\big(\|q_i - q_j\|\big),$$

where $V_{ij}(r) = -Gm_im_j/r$, and the constants G, m_i are those used in [11]. This system has the total energy $H(p,q)$ and also the angular momentum $L(p,q) = \sum_{i=1}^{N} p_i \times q_i$ as first integrals. Therefore, it is a differential equation on the manifold

$$\mathcal{M} = \{(p,q) \,|\, H(p,q) = \mathit{const}, L(p,q) = \mathit{const}\}. \tag{2.5}$$

In our experiment of Figure 2.2 we apply the explicit Euler method with step size $h = 10$, and we project after every integration step orthogonally onto the manifold $\mathcal{M}$ (see Section 3). Hence the numerical solution exactly conserves the energy $H(p,q)$ and also the angular momentum $L(p,q)$, but nevertheless it does not show the expected quasiperiodic motion of the planets. From a geometric integrator of this problem we expect more that just producing an approximation that lies on the manifold (2.5).

At this point it is interesting to mention that for *constrained Hamiltonian systems* much attention has been paid for qualitatively correct simulations (especially in molecular dynamics). Let us mention the symmetric methods SHAKE and

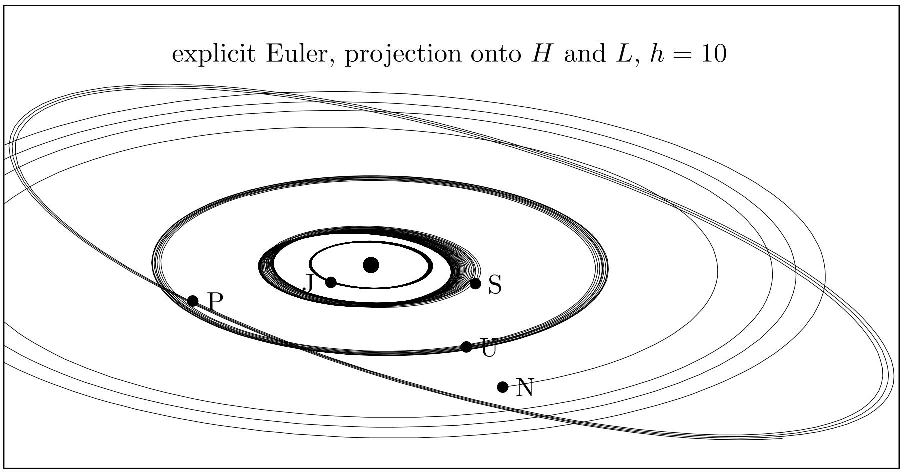


Figure 2.2: Numerical simulation of the outer solar system.

RATTLE of Ryckaert, Ciccotti and Berendsen [23] and Andersen [1]. Their symplecticity has been studied by Leimkuhler and Skeel [18], and further geometric integrators are investigated by Reich [21] and Jay [17].

For differential equations on general manifolds much less work has been investigated for a qualitatively correct simulation. The next two sections will show how classical approaches for differential equations on manifolds can be modified in order to make them to symmetric integrators.

3 Projection methods.

Consider a differential equation $\dot{y} = f(y)$ on a manifold $\mathcal{M}$. Projection methods are a standard approach for the numerical solution of differential equations on manifolds. One step $y_n \mapsto y_{n+1}$ proceeds as follows:

Algorithm 3.1 (Standard projection method).

- *Compute $\widehat{y}_{n+1} = \Phi_h(y_n)$, where Φ_h represents any numerical integrator applied to $\dot{y} = f(y)$, e.g., a Runge–Kutta method;*
- *project $\widehat{y}_{n+1}$ orthogonally onto the manifold $\mathcal{M}$ to obtain $y_{n+1} \in \mathcal{M}$.*

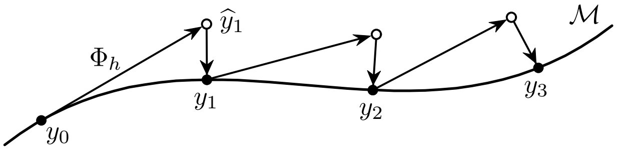


Figure 3.1: Use of standard projection.

The standard projection method is illustrated in Figure 3.1. Consider again the rigid body equations of Example 2.1. This time we apply the (symmetric) trapezoidal rule (5000 steps with $h = 1$) and, in order to obtain an approximation that lies exactly on the sphere, we apply the projection of Algorithm 3.1. The rather disappointing result is presented in Figure 3.2 (left picture). This is explained by the fact that the projection destroys the symmetry of the underlying method.

How can we restore the symmetry? A symmetric projection step has been proposed by Ascher and Reich [3] in order to enforce energy conservation in Hamiltonian systems. The following algorithm is introduced and discussed in Hairer [12].

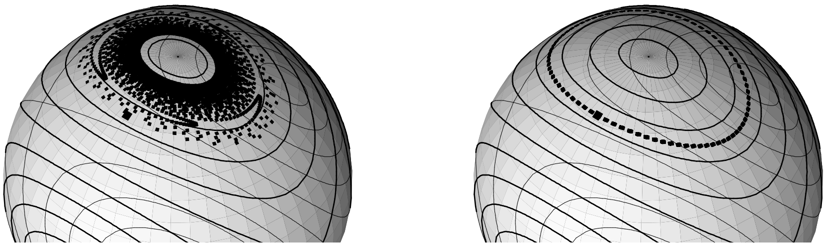


Figure 3.2: Rigid body simulation with projection methods; standard projection (left), and symmetric projection (right).

The idea is to perturb the vector y_n before applying a symmetric one-step method, such that the final projection is of the same size as the perturbation.

Algorithm 3.2 (Symmetric projection method).

- $\widehat{y}_n = y_n + G^T(y_n)\mu$ *where* $g(y_n) = 0$*;*
- $\widehat{y}_{n+1} = \Phi_h(\widehat{y}_n)$ *(symmetric one-step method for* $\dot{y} = f(y)$*);*
- $y_{n+1} = \widehat{y}_{n+1} + G^T(y_{n+1})\mu$ *with* μ *such that* $g(y_{n+1}) = 0$.

Here, $G(y) = g'(y)$ *denotes the Jacobian of* $g(y)$*, if the manifold is given by* $\mathcal{M} = \{y \mid g(y) = 0\}$*. It is important to take the same vector* μ *in the perturbation and in the projection.*

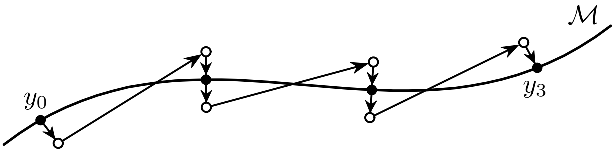


Figure 3.3: Use of symmetric projection.

In the numerical experiment of Figure 3.2 we replace the standard projection with the symmetric projection of Algorithm 3.2. The numerical solution apparently lies on a closed curve close to the exact solution (right picture of Figure 3.2). This new algorithm is therefore a geometric integrator for the rigid body equations, considered as a differential equation on the sphere. Further numerical experiments (not presented here) demonstrate that the symmetric projection recovers in many situations the correct long-time behaviour.

4 Methods based on local coordinates.

Besides projection methods, the use of local coordinate transformations is a further well-established approach for solving differential equations on manifolds. Let us shortly outline the main idea. Assume that

$$\psi : U \to \mathbb{R}^n, \qquad \psi(U) \subset \mathcal{M}$$

is a local parametrization of the manifold $\mathcal{M}$ (close to $y_n = \psi(z_n)$). The change of coordinates $y = \psi(z)$ then transforms the differential equation $\dot{y} = f(y)$ into

$$\psi'(z)\dot{z} = f\big(\psi(z)\big). \tag{4.1}$$

This looks like an overdetermined system of differential equations, because the dimension of z is equal to that of $\mathcal{M}$, which is usually smaller than that of y and $f(y)$. However, $f(y) \in T_y\mathcal{M}$ by assumption, so that (4.1) actually becomes equivalent to a system

$$\dot{z} = F(z), \qquad z(t_n) = z_n. \tag{4.2}$$

The idea is to compute one step of a numerical method applied to (4.2), and to map the result via the transformation ψ back to the manifold. One step $y_n \mapsto y_{n+1}$ of the resulting algorithm is defined as follows:

Algorithm 4.1 (Local coordinates approach).

- *Choose a local parametrization and compute* z_n *from* $y_n = \psi(z_n)$*;*
- *compute* $\widehat{z}_{n+1} = \Phi_h(z_n)$*, the result of the method* Φ_h *applied to* (4.2)*;*
- *define the numerical solution by* $y_{n+1} = \psi(\widehat{z}_{n+1})$*.*

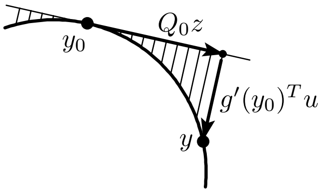
It is important to remark that the parametrization $y = \psi(z)$ *can be changed in every step (see Figure* 4.1*).*

There are many possible choices for local parametrizations. We just mention a few of them, which are rather general. For the first two choices we assume that the manifold is given by $\mathcal{M} = \{y \mid g(y) = 0\}$.

- *Generalized coordinate partitioning* (Wehage and Haug [25]): based on a QR decomposition of the matrix $g'(y)$, one selects for z a suitable subset of the components of y.
- *Tangent space parametrization* (Potra and Rheinboldt [20]): we split

$$y - y_0 = \underbrace{Q_0 z}_{T_{y_0}\mathcal{M}} + \underbrace{g'(y_0)^T u}_{(T_{y_0}\mathcal{M})^\perp},$$

  where $u = u(z)$ is such that $g(y) = 0$. This defines $y = \psi(z) = \psi_{y_0}(z)$.



- *Exponential map* $\exp : \mathfrak{g} \to G$ for differential equations on Lie groups G (Munthe-Kaas [19]). The parameter space is the corresponding Lie algebra $\mathfrak{g}$.
- *Cayley transform* for quadratic Lie groups (Diele, Lopez and Peluso [7])

$$Y = \psi(Z) = (I - Z)^{-1}(I + Z).$$

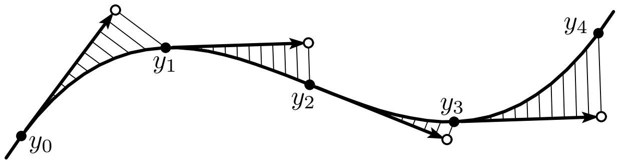


Figure 4.1: Use of standard tangent space parametrization.

For an illustration we use again the rigid body equations of Example 2.1. But, in order to avoid symmetries in the manifold, we consider this time the differential equation (2.2) as a problem on the manifold

$$\mathcal{M} = \left\{ y \,\Big|\, \frac{y_1^2}{I_1} + \frac{y_2^2}{I_2} + \frac{y_3^2}{I_3} = const \right\},$$

and we do not explicitly use the fact that the solution lies on the sphere. Using the symmetric trapezoidal rule and the above described tangent space parametrization

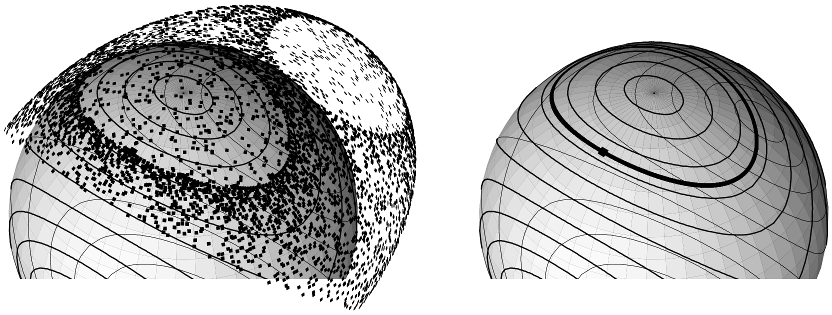


Figure 4.2: Rigid body simulation with methods based on local coordinates; tangent space at initial value (left), and tangent space at the midpoint (right).

(see Figure 4.1), the local coordinates approach of Algorithm 4.1 produces the result shown in the left picture of Figure 4.2. As expected, the solution lies on the ellipsoid, but there it does not show the correct qualitative behaviour.

We now explain how this approach can be symmetrized. We consider the tangent space parametrization only (see Figure 4.3), but an extension to other parametrizations such as the exponential map or the Cayley transform for Lie group methods, is straight-forward.

Algorithm 4.2 (Symmetric use of local coordinates).

- Consider a local parametrization depending on ζ and compute z_n from $y_n = \psi_\zeta(z_n)$;
- compute $\widehat{z}_{n+1} = \Phi_h(z_n)$, the result of the method Φ_h applied to (4.2);
- define the numerical solution by $y_{n+1} = \psi_\zeta(\widehat{z}_{n+1})$;
- determine implicitly $\zeta \in \mathcal{M}$ such that $\zeta = (z_n + \widehat{z}_{n+1})/2$.

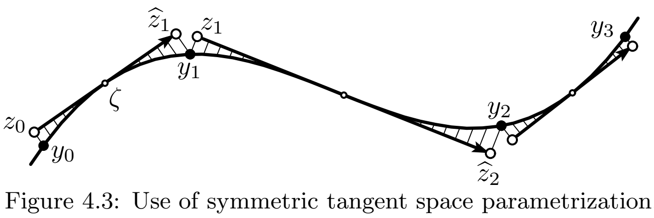


Figure 4.3: Use of symmetric tangent space parametrization.

Using Algorithm 4.2 we get a qualitatively correct result as shown in the right picture of Figure 4.2. There is a close connection of Algorithm 4.2 with the one proposed by Zanna, Engø and Munthe-Kaas [26] for Lie group methods. In that article, the exponential map is centered at the geodesic midpoint of y_n and y_{n+1}. We suggest to consider the midpoint in the parameter space.

Acknowledgement.

We are grateful to Åke Björck, to Gustaf Söderlind, and to his secretary Aniko Wolf for making the 40th anniversary meeting of BIT to a success and to an unforgettable event.

REFERENCES

1. H. C. Andersen, *Rattle: A "velocity" version of the Shake algorithm for molecular dynamics calculations*, J. Comput. Phys., 52 (1983), pp. 24–34.
2. U. M. Ascher, H. Chin, and S. Reich, *Stabilization of DAEs and invariant manifolds*, Numer. Math., 67 (1994) pp. 131–149.
3. U. M. Ascher and S. Reich, *On some difficulties in integrating highly oscillatory Hamiltonian systems*, Lecture Notes in Computational Science and Engineering, 4 (1998), pp. 281–296.
4. J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Comput. Methods Appl. Mech. Engrg., 1 (1972), pp. 11–16.
5. K. E. Brenan, S. L. Campbell and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North Holland, New York, 1989.
6. P. E. Crouch and R. Grossman, *Numerical integration of ordinary differential equations on manifolds*, J. Nonlinear Sci., 3 (1993), pp. 1–33.
7. F. Diele, L. Lopez, and R. Peluso, *The Cayley transform in the numerical solution of unitary differential systems*, Adv. Comput. Math., 8 (1998), pp. 317–334.
8. E. Eich-Soellner and C. Führer, *Numerical Methods in Multibody Dynamics*, Teubner, Stuttgart, 1998.
9. C. W. Gear, *Simultaneous numerical solution of differential-algebraic equations*, IEEE Trans. Circuit Theory, CT-18 (1971), pp. 89–95.
10. E. Griepentrog and R. März: *Differential-algebraic equations and their numerical treatment*, Teubner Texte zur Math., Vol. 88, Teubner, Stuttgart, 1986.
11. E. Hairer, *Numerical Geometric Integration*, Unpublished Lecture Notes, March 1999, available on http://www.unige.ch/math/folks/hairer/.
12. E. Hairer, *Symmetric projection methods for differential equations on manifolds*, BIT, 40:4 (2000), pp. 726–734.
13. E. Hairer, Ch. Lubich, and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods*, Lecture Notes in Mathematics, Vol. 1409, Springer-Verlag, Berlin, 1989.
14. E. Hairer, Ch. Lubich, and G. Wanner, *Geometric Numerical Integration*, monograph in preparation.
15. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, 2nd ed., Springer Series in Comput. Math., Vol. 14, Springer-Verlag, Berlin, 1996.
16. A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, *Lie-group methods*, Acta Numerica, 9 (2000), pp. 215–365.
17. L. Jay, *Symplectic partitioned Runge–Kutta methods for constrained Hamiltonian systems*, SIAM J. Numer. Anal., 33 (1996), pp. 368–387.
18. B. J. Leimkuhler and R. D. Skeel, *Symplectic numerical integrators in constrained Hamiltonian systems*, J. Comput. Phys., 112 (1994), pp. 117–125.
19. H. Munthe-Kaas, *High order Runge–Kutta methods on manifolds*, Appl. Numer. Math., 29 (1999), pp. 115–127.
20. F. A. Potra and W. C. Rheinboldt, *On the numerical solution of Euler–Lagrange equations*, Mech. Struct. Mech., 19 (1991), pp. 1–18.
21. S. Reich, *Symplectic integration of constrained Hamiltonian systems by composition methods*, SIAM J. Numer. Anal., 33 (1996), pp. 475–491.

22. W. C. Rheinboldt, *Differential-algebraic systems as differential equations on manifolds*, Math. Comp., 43 (1984), pp. 473–482.

23. J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, *Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes*, J. Comput. Phys., 23 (1977), pp. 327–341.

24. L. F. Shampine, *Conservation laws and the numerical solution of ODEs*, Comput. Maths. Appls., 12B (1986) pp. 1287–1296.

25. R. A. Wehage and E. J. Haug, *Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems*, J. Mechanical Design, 104 (1982), pp. 247–255.

26. A. Zanna, K. Engø, and H. Munthe-Kaas, *Adjoint and selfadjoint Lie-group methods*, BIT, 41:2 (2001), pp. 395–421.