The Newton Iteration on Lie Groups *

Brynjulf Owren† and Bruno Welfert‡

†Department of Mathematical Sciences, NTNU, N-7034 Trondheim, Norway

‡Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804, U.S.A.

September 19, 1996

Abstract

We define the Newton iteration for solving the equation $f(y) = 0$, where f is a map from a Lie group to its corresponding Lie algebra. Two versions are presented, which are formulated independently of any metric on the Lie group. Both formulations reduce to the standard method in the Euclidean case, and are related to existing algorithms on certain Riemannian manifolds. In particular, we show that, under classical assumptions on f, the proposed method converges quadratically. We illustrate the techniques by solving a fixed-point problem arising from the numerical integration of a Lie-type initial value problem via implicit Euler.

*This work was in part sponsored by The Norwegian Research Council under contract no. 111038/410, through the SYNODE project. WWW: `http://www.imf.unit.no/num/synode`

†Email: `Brynjulf.Owren@imf.unit.no`, WWW: `http://www.imf.unit.no/~bryn`

‡Email: `bdw@math.la.asu.edu`, WWW: `http://math.la.asu.edu/~bdw`

1 Motivation

Recently, there has been an increased interest in studying numerical algorithms on manifolds. Traditionally, most numerical algorithms we know from the literature are formulated in vector spaces, and the convergence properties are usually investigated by imposing the additional structure of a Banach space. This approach can be justified through the fact that all manifolds can, by a proper choice of coordinates, be interpreted locally as Euclidian $\mathbb{R}^N$. This may however lead to difficulties if the algorithm operates on different charts. Another frequently used approach is to embed the manifold globally into a higher-dimensional Euclidian space. In this case there is the difficulty that the approximation will typically drift off the manifold if no information about the manifold is encoded in the algorithm. Finally, to accomodate the use of object-oriented implementations of numerical algorithms, it is of importance to phrase the algorithms independently of any particular choice of coordinates. Edelmann et al. [5] introduce conjugate gradient methods for Stiefel and Grassman manifolds by using a Riemannian metric and by following the corresponding geodesics.

The motivation for this paper comes from recent approaches for solving ordinary differential equations on manifolds, see Munthe-Kaas [9, 10], Crouch and Grossman [4], and Marthinsen and Owren [8]. So far, the efforts have been focusing on explicit methods, but we believe that there is also a need for implicit methods for manifolds. Consider for instance the implicit Euler method applied to the problem $y' = g(y)$, which has the form

$$y^{(k+1)} = y^{(k)} + hg(y^{(k+1)})$$

Now, if instead we consider a differential equation on a Lie matrix group G with corresponding Lie algebra $\mathfrak{g}$, i.e.

$$y' = y \cdot g(y), \quad y \in G \subset \mathrm{GL}(n, \mathbb{R}),\ g : G \to \mathfrak{g} \subset \mathfrak{gl}(n, \mathbb{R}),$$

the Crouch-Grossman and Munthe-Kaas versions of the implicit Euler method would be

$$y^{(k+1)} = y^{(k)} \cdot \exp(hg(y^{(k+1)})). \tag{1}$$

For general g it is not possible to solve this equation explicitely. Instead we try to compute a sequence $\left(y_n^{(k+1)}\right)_{n\geq 0}$ that converges to the solution of (1) in some sense. An obvious choice for such a sequence is the one generated by the fixed point iteration

$$y_{n+1}^{(k+1)} = y^{(k)} \cdot \exp(hg(y_n^{(k+1)})),\ n = 0, 1, \ldots \tag{2}$$

In the classical version of the implicit Euler method, a fixed point iteration of the above type is frequently not feasible, usually, some version of the Newton iteration is preferred. The generalization of Newton's method to (1) can be obtained by observing that for y "near" $y^{(k)}$, we have the representation $y = y^{(k)} \cdot \exp(v(y))$, with $v(y) \in \mathfrak{g}$, thus we need to solve the equation

$$f(y) = 0, \tag{3}$$

where

$$\begin{array}{rccl} f: & G & \to & \mathfrak{g} \\ & y & \mapsto & v(y) - hg(y). \end{array}$$

In this paper we propose two algorithms for solving problems of the form $f(y) = 0$ where f is a map from a Lie group to its corresponding Lie algebra.

We start by reviewing some of the main ideas of interest to us regarding Lie groups and Lie algebras and their relationship. We also define the notion of convergence in the Lie group. This is done in Section 2. Section 3 presents two versions of the Newton algorithm, which are compared and analyzed in Section 4. Section 5 illustrates the numerical behavior of the proposed algorithms with two simple examples arising from an implicit Euler time discretization of a Lie-type ODE. Finally, we conclude with some important remarks on the applicability and possible generalizations to homogeneous manifolds.

2 Background and Notations

2.1 Lie group, tangent space and Lie algebra

A *Lie group* $(G, \cdot)$ is a differential manifold G equipped with a binary operation $\cdot$ on $G \times G$ satisfying the axioms of a group, such that the map

$$\begin{array}{ccc} G \times G & \to & G \\ (y, z) & \mapsto & y \cdot z^{-1} \end{array}$$

is smooth. By abuse of notation, the Lie group is simply refered to as G. The symbol e designates the identity element of G. We refer to [13, 14] for the definition of a differentiable manifold.

The *tangent space* to G at a point $y \in G$, denoted $TG|_y$, is the set of all functional operators Δ_y (also called linear derivations) such that

$$\Delta_y(\lambda f + \mu g) = \lambda \Delta_y(f) + \mu \Delta_y(g) \quad \text{(linearity)}$$

and

$$\Delta_y(fg) = f(y)\Delta_y(g) + g(y)\Delta_y(f) \quad \text{(derivation, Leibnitz rule)}$$

for all functions f, g defined in a neighborhood of y, and $\lambda, \mu \in \mathbb{R}$. The union of all tangent spaces as y describes G is called the *tangent bundle*, denoted TG, i.e.,

$$TG = \bigcup_{y \in G} TG|_y.$$

A *vector field* on G is a map

$$\begin{array}{rccc} X: & G & \to & TG \\ & y & \mapsto & \Delta_y \end{array}$$

associating a tangent vector $\Delta_y \in TG|_y$ to each $y \in G$. The *differential* of a map $\varphi : G \to G$ is a map $\varphi' : TG \to TG$. Composition of maps (denoted $\circ$) works the usual way under differentiation, i.e.,

$$(\varphi \circ \psi)' = \varphi' \circ \psi' \quad \text{(chain rule)} \tag{4}$$

if φ and ψ are two differentiable maps.

For a given $y \in G$, the mapping

$$\begin{array}{rccc} L_y: & G & \to & G \\ & z & \mapsto & y \cdot z \end{array}$$

is called *left multiplication* in the group. The set of *left-invariant* vector fields X (which satisfy

$$L'_y \circ X = X \circ L_y$$

for all $y \in G$, i.e., such that the diagram

$$\begin{tikzcd} TG \arrow[r, "L'_y"] & TG \\ G \arrow[u, "X"] \arrow[r, "L_y"] & G \arrow[u, "X"'] \end{tikzcd}$$

is commutative) has the structure of an algebra, and is called the *Lie algebra* of the Lie group G, denoted $\mathfrak{g}$. The multiplication operator in the Lie algebra is a skew-symmetric bilinear operator called the *Lie bracket*, denoted $[-,-]$, which satisfies the Jacobi identity

$$[[u, v], w] + [[v, w], u] + [[w, u], v] = 0 \tag{5}$$

for all $u, v, w \in \mathfrak{g}$.. In particular, $[u, u] = 0$ for all $u \in \mathfrak{g}$. If G is abelian[1] then also $[u, v] = 0$ for any $u, v \in \mathfrak{g}$. The Lie bracket also defines a linear operator ad on $\mathfrak{g}$ such that

$$\mathrm{ad}_u(v) = [u, v]$$

for all $u, v \in \mathfrak{g}$. The identity (5) is then equivalent to

$$\mathrm{ad}_{[u,v]} = [\mathrm{ad}_u, \mathrm{ad}_v]$$

for all $u, v \in \mathfrak{g}$, i.e., to the fact that ad is an algebra homomorphism on $\mathfrak{g}$ [3, p.30].

Remark 2.1 *The tangent bundle $T\mathfrak{g}$ of $\mathfrak{g}$ can be identified with $\mathfrak{g}$ itself since $\mathfrak{g}$ is a vector space.*

Remark 2.2 *If G is finite dimensional then $\mathfrak{g}$ is a finite dimensional vector space in which the standard Newton algorithm can be applied to solve nonlinear equations, provided a coordinate system for $\mathfrak{g}$ is available.*

The map L'_y determines an isomorphism of $\mathfrak{g}$ with the tangent space $TG|_y$ to G at $y \in G$ via the relation

$$L'_y(\mathfrak{g}) = TG|_y \tag{6}$$

or, equivalently,

$$\mathfrak{g} = \left(L'_y\right)^{-1}(TG|_y) = L'_{y^{-1}}(TG|_y) \tag{7}$$

(the map $L'_{y^{-1}}$ is generally refered to as the canonical left invariant 1-form on G, see [3, p.35]). In particular, since L_e is the identity operator on G, its differential L'_e is also the identity and we have the identification

$$\mathfrak{g} = TG|_e$$

and an element $u \in \mathfrak{g}$ defines a left-invariant vector field X_u by the relation

$$X_u(y) = L'_y(u)$$

(see Figure 1).

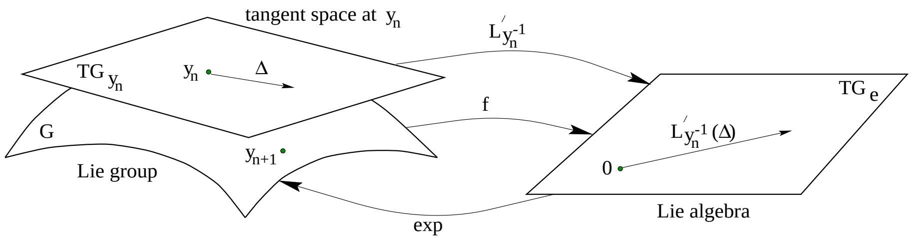

Figure 1: Lie group, tangent space and Lie algebra.

2.2 Some examples

Varadarajan [13] and Abraham et al. [1, supplementary chapter 9] give a list of a few of the most important Lie groups. We include two of them here: the first example is the standard Euclidean case, which provides a basis for comparing the proposed Newton algorithm described in Section 3.3 with its classical counterpart. The second example will provide the setting for solving a fixed-point problem arising from a Lie-type differential equation in Section 5.

[1] i.e, $L_y(z) = L_z(y)$ for all $y, z \in G$.

2.2.1 Euclidean space $\mathbb{R}^n$

The Euclidean space $\mathbb{R}^n$ is a particular example of an abelian Lie group. In this case we have

$$\begin{aligned}(G,\cdot) &= (\mathbb{R}^n,+)\\ TG &= \mathbb{R}^n\\ (\mathfrak{g},+,[-,-]) &= (\mathbb{R}^n,+,[u,v]=0)\\ L_y(z) &= y+z\\ L'_y(u) &= u\end{aligned}$$

2.2.2 Lie group $\mathbf{O}(n,\mathbb{R})$

The set $\mathrm{GL}(n,\mathbb{R})$ of all invertible $n\times n$ matrices with real coefficients forms a Lie group under standard matrix multiplication. The subset $\mathrm{O}(n,\mathbb{R})$ of $\mathrm{GL}(n,\mathbb{R})$ consisting of orthogonal matrices y (such that $y^T y = I_n$, where I_n denotes the $n\times n$ identity matrix) forms a subgroup of $\mathrm{GL}(n,\mathbb{R})$ called the orthogonal group. The corresponding Lie algebra is the set of all $n\times n$ skew-symmetric matrices

$$\mathfrak{so}(n,\mathbb{R}) = \{v\in\mathfrak{gl}(n,\mathbb{R}), v^T+v=0\}$$

where $\mathfrak{gl}(n,\mathbb{R})$ denotes the set of all real $n\times n$ matrices ($\mathfrak{gl}(n,\mathbb{R})$ is also the Lie algebra associated with $\mathrm{GL}(n,\mathbb{R})$). We thus have

$$\begin{aligned}(G,\cdot) &= (\mathrm{O}(n,\mathbb{R}), y\cdot z = yz)\\ (\mathfrak{g},+,[-,-]) &= (\mathfrak{so}(n,\mathbb{R}),+,[u,v]=uv-vu)\\ TG &= \{yv, y\in G, v\in\mathfrak{g}\}\\ L_y(z) &= yz\\ L'_y(u) &= yu\end{aligned}$$

The orthogonal group $\mathrm{SO}(3,\mathbb{R})$ is the configuration space of a rigid body plays an important role in computational mechanics (e.g., numerical integration of orthogonal flows, see [2]). The importance of matrix Lie groups comes also from the fact that any Lie algebra over $\mathbb{R}$ is isomorphic to the Lie algebra associated to a subgroup of $\mathrm{GL}(n,\mathbb{R})$, according to Ado's Theorem [13, p.237].

2.3 The exponential map and its differential

The exponential map is a map

$$\begin{aligned}\exp:\ \mathfrak{g} &\to G\\ u &\mapsto \exp(u)\end{aligned}$$

which is certainly the single most important construct associated to G and $\mathfrak{g}$. Given $u\in\mathfrak{g}$, a vector field X_u determines an integral curve $\gamma_u(t)$ solution of the initial value problem

$$\begin{cases}\gamma'_u(t) = X_u(\gamma_u(t))\\ \gamma_u(0)=e.\end{cases}$$

The exponential map is then defined by the relation

$$\exp(u) = \gamma_u(1).$$

In the Euclidean case this map reduces to the identity on $\mathbb{R}^n$, while it corresponds to the classical exponentiation of a matrix

$$\exp(u) = \sum_{k\geq 0}\frac{u^k}{k!} \tag{8}$$

if u belongs to $\mathfrak{gl}(n,\mathbb{R})$. Note that in both cases exp maps the origin 0 of $\mathfrak{g}$ to the identity element e of G. This is always the case. In fact, the exponential mapping can be shown [13, p.86] to be an analytic diffeomorphism on an open neighborhood $\mathcal{N}(0)$ of $0 \in \mathfrak{g}$. If

$$N(e) = \exp(\mathcal{N}(0)) \tag{9}$$

denotes the image of $\mathcal{N}(0)$ by exp, this means in particular that any $y \in N(e)$ can be written in the form

$$y = \exp(v)$$

for some $v \in \mathcal{N}(0)$, and that if

$$\exp(u) = \exp(v) \in N(e)$$

for some $u, v \in \mathcal{N}(0)$, then $u = v$.

If G is abelian, exp is also a homomorphism from $\mathfrak{g}$ to G, i.e.,

$$\exp(u+v) = \exp(u)\cdot\exp(v) = \exp(v)\cdot\exp(u) \tag{10}$$

for all $u, v \in \mathfrak{g}$. In the non-abelian case (e.g., $G = \mathrm{O}(n,\mathbb{R})$), exp is not a homomorphism and (10) must be replaced by

$$\exp(w) = \exp(u)\cdot\exp(v) \tag{11}$$

where w is given by the Baker-Campbell-Hausdorff (BCH) formula [13, pp.114]

$$w = u + v + \frac{1}{2}[u,v] + \frac{1}{12}([u,[u,v]] + [v,[v,u]]) + \dots \tag{12}$$

valid[2] for $u, v \in \mathcal{N}(0)$.

The differential of exp at $u \in \mathfrak{g}$ is a map (recall $T\mathfrak{g} = \mathfrak{g}$)

$$\begin{array}{rccl} \exp'_u : & \mathfrak{g} & \to & TG|_{\exp(u)} \\ & v & \mapsto & \left.\dfrac{d}{dt}\right|_{t=0} \exp(u+tv) \end{array}$$

which is usually expressed by a map $\mathrm{dexp}_u : \mathfrak{g} \to \mathfrak{g}$ using the identification (7) of $TG|_{\exp(v)}$ with $\mathfrak{g}$, i.e.,

$$\mathrm{dexp}_u = L'_{(\exp(u))^{-1}} \circ \exp'_u = L'_{\exp(-u)} \circ \exp'_u\,. \tag{13}$$

For $v \in \mathfrak{g}$, $\mathrm{dexp}_u(v)$ can be obtained using the sum form [13, p.108] [7]

$$\begin{aligned} \mathrm{dexp}_u(v) &= \sum_{n\geq 0} \frac{(-1)^n}{(n+1)!}\mathrm{ad}_u^n(v) && (14) \\ &= \sum_{n\geq 0} \frac{(-1)^n}{(n+1)!}\underbrace{[u,[u,\dots[u}_{n\text{ terms}},v]\dots]] \\ &= v - \frac{1}{2}[u,v] + \frac{1}{6}[u,[u,v]] - \dots && (15) \end{aligned}$$

Note that if u and v commute ($[u,v]=0$) then $\mathrm{dexp}_u(v) = v$, so that dexp_0 reduces to the identity map $\mathrm{Id}_{\mathfrak{g}}$ on $\mathfrak{g}$.

[2] $\mathcal{N}(0)$ may be taken "small" enough such that the series (12) converges if $\mathfrak{g}$ is also a Banach space, see Section 2.4 and [13, (2.15.21)].

2.4 Convergence in the Lie group

How "large" $\mathcal{N}(0)$ (resp. $N(e)$) is depends on the existence of a metric defined on the Lie algebra (identified with the vector space $TG|_e$). In $\mathbb{R}^n$, we have $\mathcal{N}(0) = \mathbb{R}^n = \mathfrak{g} = G = N(e)$ $(e = 0)$. However, Varadarajan [13, p.90] gives the example of the Lie group $G = \mathrm{GL}(2, \mathbb{R})$ and a $y \in G$ such that there is no $u \in \mathfrak{g} = \mathfrak{gl}(2, \mathbb{R})$ verifying $y = \exp(u)$, i.e., $\exp(\mathfrak{g})$ is only a subset on G in general.

For our purpose of showing convergence of the Newton iteration in G (see Section 4), we shall assume that the associated Lie algebra $\mathfrak{g}$ is a Banach space, i.e., there exist a norm $\|-\|$ on $\mathfrak{g}$ inducing a complete metric. This requirement is more general than in [5], where a Newton iteration based on Hilbert spaces[3] is proposed.

We recall that a linear operator $\varphi : \mathfrak{g} \to \mathfrak{g}$ is bounded if

$$\|\varphi\| = \sup_{u \neq 0} \frac{\|\varphi(u)\|}{\|u\|} = \sup_{\|u\|=1} \|\varphi(u)\| < \infty.$$

Definition 2.1 *Let $N(e)$ be as in (9). A sequence $\{y_n\}_{n\geq 0}$ of points of $N(e)$ is said to converge to e if and only if the associated sequence $\{u_n\}_{n\geq 0}$ defined by*

$$y_n = \exp(u_n)$$

for all $n \geq 0$ converges to 0, in the sense that

$$\lim_{n\to\infty} \|u_n\| = 0. \tag{16}$$

Definition 2.2 *If there exists $y \in G$ such that*

$$y^{-1} y_n = \exp(u_n) \in N(e) \tag{17}$$

and (16) holds, then the sequence $\{y_n\}_{n\geq 0}$ is said to converge to y. Furthermore, the convergence is of order p if $\|u_{n+1}\| \leq \mathcal{O}(\|u_n\|^p)$.

As usual, linear convergence corresponds to $p = 1$, and quadratic convergence to $p = 2$.

Remark 2.3 *In the non-abelian case, the map $\mathrm{dexp} : u \mapsto \mathrm{dexp}_u$ is only "almost" linear in the vicinity of $u = 0$, the origin of $\mathfrak{g}$, see (15). The non-linearity of the exponential map may then result in a deterioration or the absence of convergence if $y^{-1}y_0$ lies outside a neighborhood of e contained in $N(e)$.*

The continuity of the Lie bracket $[-,-]$ guarantees the existence of a constant $\mu \geq 0$ such that

$$\|[u, v]\| \leq \mu \|u\| \, \|v\| \tag{18}$$

for all $u, v \in \mathfrak{g}$. This is equivalent to the boundedness of the operator ad_u on compact sets of $\mathfrak{g}$:

$$\|\mathrm{ad}_u\| \leq \mu \|u\|. \tag{19}$$

Let Φ denote the function

$$\Phi(t) = \frac{e^t - 1}{t}$$

The following lemmas will be useful in the convergence analysis of the Newton algorithm:

[3] a Hilbert space is an inner product space with a complete metric [1]. In particular, a Hilbert space is also a Banach space.

Lemma 2.1

$$\max(||u||,||v||) \le \beta \quad \Rightarrow \quad ||\mathrm{dexp}_u - \mathrm{dexp}_v|| \le \mu ||u-v|| \Phi'(\mu\beta). \tag{20}$$

Proof: Write

$$\mathrm{ad}_u^n - \mathrm{ad}_v^n = \sum_{i=0}^{n-1} \mathrm{ad}_u^i \circ (\mathrm{ad}_u - \mathrm{ad}_v) \circ \mathrm{ad}_v^{n-1-i},$$

so that

$$\begin{aligned} ||\mathrm{ad}_u^n - \mathrm{ad}_v^n|| &\le \sum_{i=0}^{n-1} ||\mathrm{ad}_u||^i ||\mathrm{ad}_u - \mathrm{ad}_v|| ||\mathrm{ad}_v||^{n-1-i} \\ &\le ||\mathrm{ad}_u - \mathrm{ad}_v|| \mu^{n-1} \sum_{i=0}^{n-1} ||u||^i ||v||^{n-1-i} \\ &\le \mu ||u-v|| n (\mu\beta)^{n-1}. \end{aligned}$$

Finally, it comes, using (14)

$$||\mathrm{dexp}_u - \mathrm{dexp}_v|| \le \sum_{n\ge 0} \frac{||\mathrm{ad}_u^n - \mathrm{ad}_v^n||}{(n+1)!} \le \mu ||u-v|| \sum_{n \ge 1} \frac{n(\mu\beta)^{n-1}}{(n+1)!} = \mu||u-v||\Phi'(\mu\beta).$$

Q.E.D.

Because $\mathrm{dexp}_0 = \mathrm{Id}_{\mathfrak{g}}$, remark that for small enough $||u|| \le \beta$ the map dexp_u is well-defined and we have

$$||\mathrm{dexp}_u|| \simeq ||\mathrm{dexp}_u^{-1}|| \simeq \Theta(\mu\beta) \tag{21}$$

for some generic function $\Theta(t)$ analytic around $t=0$ satisfying $\Theta(0)=1$. A more precise statement for (21) can be made using the functions Φ and h (defined below), but this would only obscure the exposition.

Lemma 2.2 *There exists a constant $\delta > 0$ and a real function $h = h(t)$, analytic in the disk $|t| \le \delta$, non-decreasing for $t \ge 0$, with $h(0) = 1$, such that for all $u, v \in \mathfrak{g}$*

$$||u|| + ||v|| \le \beta \quad \Rightarrow \quad ||w|| \le ||u+v||\ h(\mu\beta) \tag{22}$$

and

$$||u|| \le \frac{\beta}{2}, \quad ||u|| + ||v|| \le \beta \quad \Rightarrow \quad ||w-u|| \le ||v||\ h(\mu\beta) \tag{23}$$

for any $\beta \le \delta/\mu$, where w is given by (12).

Lemma 2.2 shows that (10) "almost" holds in terms of the norm $||-||$ for small $||u||$ and $||v||$. Note that in the abelian case, $\mu = 0$ and $h(\mu\beta) = 1$ for any $\beta \ge 0$, so that (22) and (23) are tight.

The proof of the lemma is based on a variation of the proof of [13, Theorem 2.15.4]. It is rather technical and is omitted here. The interested reader will find it in the Appendix of this paper.

3 Two versions of Newton iteration

In this section we present a Newton's algorithm for solving (3), defined directly on the Lie group G, and a variation of it, in which the standard technique is applied to a an equation posed on g obtained by transforming (3) via the exponential map. We first introduce the differential of f. The two versions of Newton's method are also compared in the last part of the section, where it is shown that the second version is merely a linearization (in a sense which is precised below) of the first one.

3.1 The differential of the map f

The differential of f at a point $y \in G$ is a map $f'_y : TG|_y \to \mathfrak{g}$ defined as

$$f'_y(\Delta_y) = \left.\frac{d}{dt}\right|_{t=0} f\left(y \cdot \exp\left(tL'_{y^{-1}}(\Delta_y)\right)\right) \tag{24}$$

for any tangent vector $\Delta_y \in TG|_y$ to the manifold G at y. The image by f'_y of a tangent vector Δ_y is obtained by first identifying Δ_y with a element $v \in \mathfrak{g}$ via left multiplication. The exponential mapping transforms the scaled (by t) v to an element z of the Lie group (see Figure 1). Then $f'_y(\Delta_y)$ is obtained as

$$f'_y(\Delta_y) = \lim_{t \to 0} \frac{f(y \cdot z) - f(y)}{t}. \tag{25}$$

Analogously to (13), the differential f'_y can be expressed via a function $\mathrm{d}f_y : \mathfrak{g} \to \mathfrak{g}$ given by

$$\mathrm{d}f_y = (f \circ L_y)' = f'_y \circ L'_y. \tag{26}$$

Thus

$$\mathrm{d}f_y(u) = f'_y\left(L'_y u\right) = \left.\frac{d}{dt}\right|_{t=0} f(y \cdot \exp(tu)) \tag{27}$$

(compare (27) with [14, (2) p.107], with $\varphi = f \circ L_y$).

Lemma 3.1 *For $y \in G$, $u \in \mathfrak{g}$ and $t \in \mathbb{R}$ the following identity holds:*

$$\frac{d}{dt} f(y \cdot \exp(tu)) = \mathrm{d}f_{y \cdot \exp(tu)}(u).$$

Proof: The result simply follows from the fact that $[tu, su] = 0$ for any $t, s \in \mathbb{R}$:

$$\begin{aligned} \frac{d}{dt} f(y \cdot \exp(tu)) &= \left.\frac{d}{ds}\right|_{s=t} f(y \cdot \exp(su)) = \left.\frac{d}{ds}\right|_{s=0} f(y \cdot \exp((s+t)u)) \\ &= \left.\frac{d}{ds}\right|_{s=0} f(y \cdot \exp(tu) \cdot \exp(su)) = \mathrm{d}f_{y \cdot \exp(tu)}(u). \end{aligned}$$

Q.E.D.

By changing u into $-u$, we obtain

Corollary 3.1 *For $y \in G$, $u \in \mathfrak{g}$ and $t \in \mathbb{R}$ we have*

$$\frac{d}{dt} f(y \cdot \exp(-tu)) = -\mathrm{d}f_{y \cdot \exp(-tu)}(u).$$

3.2 The Newton iteration

Using the formula (24), Newton's method on the Lie group G may proceed as follows: given $y_0 \in G$, we first determine the differential f'_{y_0} according to (24). Then find $\Delta_{y_0} \in TG|_{y_0}$ satisfying the equation

$$f'_{y_0}(\Delta_{y_0}) + f(y_0) = 0,$$

and finally update y_0 by $y_1 = y_0 \cdot \exp\left(L'_{y_0^{-1}}(\Delta_{y_0})\right)$. In view of (26), the algorithm can be written as

Version 1: NEWTON'S METHOD ON THE LIE GROUP G

- Given $y_n \in G$, determine $\mathrm{d}f_{y_n}$ according to (27).
- Find $u_n \in \mathfrak{g}$ such that
$$\mathrm{d}f_{y_n}(u_n) + f(y_n) = 0. \tag{28}$$
- Compute $y_{n+1} = y_n \cdot \exp(u_n)$.

Since $y_{n+1} = y_n \cdot \exp\left(-\mathrm{d}f_{y_n}^{-1} \circ f(y_n)\right) = L_{y_n} \circ \exp \circ\, \mathrm{d}f_{y_n}^{-1} \circ (-f)(y_n)$, the method is a fixed-point iteration on G, with iteration function

$$\varphi(y) = L_y \circ \exp \circ\, \mathrm{d}f_y^{-1} \circ (-f)(y).$$

Remark 3.1 *If the map f is an homomorphism from G to $\mathfrak{g}$ (i.e., if $f(y \cdot z) = f(y) + f(z)$ for all $y, z \in G$), then*

$$\begin{aligned}
\mathrm{d}f_y(u) &= \lim_{t\to 0} \frac{f(y \cdot \exp(tu)) - f(y)}{t} = \lim_{t \to 0} \frac{f(\exp(tu)))}{t} \quad (= \mathrm{d}f_e(u)) \\
&= \lim_{n\to\infty} nf(\exp(\frac{1}{n}u)) = \lim_{n\to\infty} f(\exp(u)) = f(\exp(u)),
\end{aligned}$$

so that

$$f(y_1) = f(y_0 \cdot \exp(u_0)) = f(y_0) + \mathrm{d}f_{y_0}(u_0) = 0$$

and this version of the algorithm converges in one iteration independently of the initial choice y_0.

Practical implementations of the above algorithm requires the determination of the differential $\mathrm{d}f_y$ for a given y, as well as the solution to the equation (28).

Example 3.1 Given $z \in G$ and $h > 0$, the equation

$$y = z \cdot \exp(hg(y))$$

is equivalent, for y such that $z^{-1}y \in N(e)$, to the equation

$$f(y) \stackrel{\text{def}}{=} v(y) - hg(y) = 0, \tag{29}$$

with

$$L_{z^{-1}}(y) = z^{-1}y = \exp(v(y)). \tag{30}$$

The quantity $f(y_n)$ is thus evaluated as

$$f(y_n) = \exp^{-1}(z^{-1} \cdot y_n) - hg(y_n).$$

The differential $\mathrm{d}f_{y_n}$ is obtained by differentiating v and g:

$$\mathrm{d}f_{y_n} = \mathrm{d}v_{y_n} - h\mathrm{d}g_{y_n}.$$

From the relation (30) we obtain

$$L'_{z^{-1}} = \exp'_{v(y)} \circ v',$$

so that

$$\mathrm{dexp}_{v(y_n)}(\mathrm{d}v_{y_n}) = L'_{\exp(-v(y_n))} \circ \exp'_{v(y_n)} \circ v' \circ L'_{y_n} = L'_{\exp(-v(y_n))\cdot z^{-1}\cdot y_n} = L'_e = \mathrm{Id}_{\mathfrak{g}}.$$

Consequently, equation (28) becomes[4]

$$\mathrm{dexp}^{-1}_{v(y_n)}(u_n) - h\mathrm{d}g_{y_n}(u_n) + f(y_n) = 0. \tag{31}$$

[4] The notation $\mathrm{dexp}^{-1}_{v(y_n)}$ is understood as $\left(\mathrm{dexp}_{v(y_n)}\right)^{-1}$. Since $z^{-1} \cdot y_n = \exp(v(n))$, it is also true that

$$\mathrm{dexp}^{-1}_{v(y_n)} = \mathrm{d}\left(\exp^{-1}\right)_{z^{-1}\cdot y_n}.$$

Because of the identity

$$\operatorname{dexp}_v^{-1} = \mathrm{Id}_{\mathfrak{g}} + \frac{1}{2}\mathrm{ad}_v + \sum_{p\geq 1} F_{2p}\ \mathrm{ad}_v^{2p} \tag{32}$$

resulting from (14), (53) and the linearity of ad_v, the equation (31) is a problem of Sylvester type. It can be solved for u_n via the solution of a linear system, provided a coordinate system for $\mathfrak{g}$ is known.

From a practical point of view, the quantity $\operatorname{dexp}_{v(y_n)}^{-1}(u)$ can be evaluated by:

- truncating the series (32) to a few terms, since one expects from (29) that $v(y_n)$ be "small" compared to 1 for small h, so that $\|\mathrm{ad}_v\| \ll \|\mathrm{Id}_{\mathfrak{g}}\|$. For example,

$$\operatorname{dexp}_{v(y_n)}^{-1}(u) \simeq u \tag{33}$$

  (as in the commutative case) or

$$\operatorname{dexp}_{v(y_n)}^{-1}(u) \simeq u + \frac{1}{2}[v(y_n), u]. \tag{34}$$

- using a finite difference approximation

$$\begin{aligned}
\operatorname{dexp}_{v(y_n)}^{-1}(u) &= \mathrm{d}\left(\exp^{-1}\right)_{z^{-1}\cdot y_n}(u) \\
&= \left.\frac{d}{dt}\right|_{t=0} \exp^{-1}(z^{-1}\cdot y_n \cdot \exp(tu)) \\
&\simeq \frac{\exp^{-1}(z^{-1}\cdot y_n\cdot \exp(tu)) - \exp^{-1}(z^{-1}\cdot y_n)}{t} \\
&= \frac{\exp^{-1}(z^{-1}\cdot y_n\cdot\exp(tu)) - v(y_n)}{t}
\end{aligned} \tag{35}$$

  for some small value $t > 0$ (e.g., $t = \mathcal{O}(\sqrt{\mathbf{u}})$, where $\mathbf{u}$ is the round-off unit of the machine).

Alternately, the change of variable $u = \operatorname{dexp}_{v(y_n)}(w)$ (resp. $u_n = \operatorname{dexp}_{v(y_n)}(w_n)$) transforms (31) into

$$w_n - h\mathrm{d}g_{y_n} \circ \operatorname{dexp}_{v(y_n)}(w_n) + f(y_n) = 0. \tag{36}$$

A simple calculation shows that

$$\mathrm{d}g_{y_n} \circ \operatorname{dexp}_{v(y_n)} = \mathrm{d}(g\circ L_z \circ \exp)_{v(y_n)}, \tag{37}$$

so that u_n in (31) may also be obtained by solving (37) for w_n using an approximation of $\mathrm{d}g_{y_n} \circ \operatorname{dexp}_{v(y_n)}(w_n)$ similar to (35), i.e.,

$$\mathrm{d}g_{y_n} \circ \operatorname{dexp}_{v(y_n)}(w_n) \simeq \frac{g(z\cdot\exp(v_n + tw_n)) - g(y_n)}{t}, \tag{38}$$

followed by an approximation of $\operatorname{dexp}_{v(y_n)}(w_n)$ similar to (33) or (34),

$$u_n \simeq w_n, \tag{39}$$

or

$$u_n \simeq w_n - \frac{1}{2}[v(y_n), w_n]. \tag{40}$$

Note however that approximations like (33), (34), (39) or (40) may not be computationally satisfying in cases where the initial guess y_0 is "too far" from the solution y (for example when solving (29) with a large value of h and $y_0 = z$), due to the non-linearity of the exponential map (see Remark 2.3). In non-abelian groups, the approximation (35) should be preferred.

For more general functions f, the differential $\mathrm{d}f_{y_n}$ can be evaluated at u using (27) in a form similar to (35), namely,

$$\mathrm{d}f_{y_n}(u) \simeq \frac{f(y_n\cdot\exp(tu)) - f(y_n)}{t} \tag{41}$$

for some $t = \mathcal{O}(\sqrt{\mathbf{u}})$.

Without loss of generality, suppose now that there exists a *known* $z \in G$ and some $v_n \in \mathfrak{g}$ such that
$$y_n = z \cdot \exp(v_n) = L_z \circ \exp(v_n).$$
For example, z may be an approximation to the solution y to $f(y) = 0$, such that $z^{-1} y_n \in N(e)$. Then
$$f(y_n) = f \circ L_z \circ \exp(v_n) \stackrel{\text{def}}{=} \tilde{f}(v_n).$$
The function $\tilde{f} = f \circ L_z \circ \exp$ is a map from $\mathfrak{g}$ to $\mathfrak{g}$. In particular, the differential $\mathrm{d}\tilde{f}_{u_n}$ of $\tilde{f}$ at v_n is a linear map defined by
$$\mathrm{d}\tilde{f}_{v_n}(u) = \left.\frac{d}{dt}\right|_{t=0} \tilde{f}(v_n + tu) \tag{42}$$
If $\mathfrak{g}$ is finite dimensional, then the standard Newton procedure can be applied to the problem $\tilde{f}(v) = 0$. This leads to the following algorithm:

Version 2: LIE ALGEBRA-BASED NEWTON'S METHOD

- Given $y_n \in G$ such that $y_n = z \cdot \exp(v_n)$, determine $\mathrm{d}\tilde{f}_{v_n}$ according to (42).
- Find $u_n \in \mathfrak{g}$ such that
$$\mathrm{d}\tilde{f}_{v_n}(u_n) + \tilde{f}(v_n) = 0. \tag{43}$$
- Compute $v_{n+1} = v_n + u_n$ and $y_{n+1} = z \cdot \exp(v_{n+1})$.

In practice, one may start Version 2 with an initial guess v_0 in $\mathfrak{g}$ rather than a y_0 in G and avoid computing the updates y_n until convergence of v_n (in $\mathfrak{g}$). If the solution y to $f(y) = 0$ is expected to be "close" to z, then $v_0 = 0$ is appropriate as a starting guess.

Remark 3.2 *If the map $\tilde{f}$ is linear, then*
$$f(z) = \tilde{f}(0) = \tilde{f}(0 + 0) = f(z) + f(z),$$
so that $f(z) = 0$, so that z is already a solution of $f(y) = 0$. Note also that $\tilde{f}$ linear is not equivalent to f being an homomorphism from G to $\mathfrak{g}$ (see Remark (3.1)), since
$$\tilde{f}(u) + \tilde{f}(v) = f(z \cdot \exp(u)) + f(z \cdot \exp(v)) = f(z \cdot \exp(u) \cdot \exp(v)) \neq f(z \cdot \exp(u + v)) = \tilde{f}(u + v)$$
in general if $f(z) = 0$ and f is an homomorphism.

Example 3.2 Using (30) the equation (29) in Example 3.1 can be written
$$\tilde{f}(v) \stackrel{\text{def}}{=} f(z \cdot \exp(v)) = v - hg(z \cdot \exp(v)). \tag{44}$$
The differential of $\tilde{f}$ at v_n is
$$\begin{aligned} \mathrm{d}\tilde{f}_{v_n} &= \mathrm{Id}_{\mathfrak{g}} - h\mathrm{d}(g \circ L_z \circ \exp)_{v_n} \\ &= \mathrm{Id}_{\mathfrak{g}} - h\mathrm{d}g_{y_n} \circ L'_{y_n^{-1}} \circ L'_z \circ \exp'_{v_n} \\ &= \mathrm{Id}_{\mathfrak{g}} - h\mathrm{d}g_{y_n} \circ \mathrm{dexp}_{v_n}, \end{aligned}$$
so that (43) becomes
$$u_n - h\mathrm{d}g_{y_n} \circ \mathrm{dexp}_{v_n}(u_n) + \tilde{f}(v_n) = 0. \tag{45}$$
Since
$$\tilde{f}(v_n) = f(z \cdot \exp(v_n)) = f(y_n) \tag{46}$$
if $v_n = v(y_n)$ with y_n defined in (30) $(y = y_n)$, u_n in (45) is equal to w_n in (36). In particular, $\mathrm{d}g_{y_n} \circ \mathrm{dexp}_{v_n}(u_n)$ nay be evaluated as in (38).

Remark 3.3 *More generally, it is easy to see that*
$$\mathrm{d}\tilde{f}_{v_n}(u_n) = \mathrm{d}f_{y_n} \circ \mathrm{dexp}_{v_n}(u_n) \tag{47}$$
holds.

3.3 Comparison of the two versions

In view of (36) and (45), the two algorithms are related. Indeed, we show here that the second version is a "linearized" version of the first algorithm. To see this, we use the following lemma:

Lemma 3.2 *Let $u, v \in \mathfrak{g}$, and consider $w = w(v)$ given by the BCH formula (12). Then*

$$w = u + \mathrm{dexp}_u^{-1}(v) + \mathcal{O}(v^2),$$

where $\mathcal{O}(v^2)$ is a smooth $\mathfrak{g}$-valued function of v such that $\mathcal{O}(v^2)/\|v^2\|$ is bounded at $v = 0$.

Proof: Differentiating the relation

$$\exp(u) \cdot \exp(v) = \exp(w)$$

defining w in terms of u and v, with respect to v yields

$$L'_{\exp(u)} \circ \exp'_v = \exp'_{w(v)} \circ w'_v$$

so that

$$w'_v = \mathrm{dexp}_{w(v)}^{-1} \circ L'_{\exp(-w(v))} \circ L'_{\exp(u)} \circ L'_{\exp(v)} \circ \mathrm{dexp}_v = \mathrm{dexp}_{w(v)}^{-1} \circ \mathrm{dexp}_v .$$

Taking $v = 0$ and noting that $w(0) = u$ (recall that $\mathrm{dexp}_0 = \mathrm{Id}_{\mathfrak{g}}$) leads to the result. **Q.E.D.**

Using Lemma 3.2 one obtains

Theorem 3.1 *Version 2 of the Newton's algorithm is a linearization of Version 1, in the sense that if $y_n^{(1)} = z \cdot \exp(v_n^{(2)})$ then*

$$y_{n+1}^{(1)} = z \cdot \exp\left(v_{n+1}^{(1)}\right)$$

with

$$v_{n+1}^{(1)} = v_{n+1}^{(2)} + \mathcal{O}\left(\left(u_n^{(1)}\right)^2\right),$$

where $\mathcal{O}(\cdot)$ is as in Lemma 3.2. Here the superscripts $(1)$and (2) refer to the version number in which the quantity is computed.

Proof: Suppose that $y_n^{(1)} = z \cdot \exp(v_n^{(2)})$. From (46) and (47) we then have $u_n^{(1)} = \mathrm{dexp}_{v_n^{(2)}}\left(u_n^{(2)}\right)$, where $u_n^{(1)}$ is defined by (28) and $u_n^{(2)}$ is determined by from (43). Thus

$$v_{n+1}^{(2)} = v_n^{(2)} + \mathrm{dexp}_{v_n^{(2)}}^{-1}\left(u_n^{(1)}\right),$$

while

$$y_{n+1}^{(1)} = y_n^{(1)} \cdot \exp\left(u_n^{(1)}\right) = z \cdot \exp(v_n^{(2)}) \cdot \exp\left(u_n^{(1)}\right) = z \cdot \exp\left(v_{n+1}^{(1)}\right)$$

where $v_{n+1}^{(1)}$ is given by the BCH formula. The result follows from Lemma 3.2 with $w = v_{n+1}^{(1)} = v_{n+1}^{(1)}\left(u_n^{(1)}\right)$. **Q.E.D.**

A direct consequence of Theorem 3.1 is that both versions of the algorithm coïncide whenever the bracket $[u_n^{(1)}, v_n^{(2)}]$ vanishes for all n.

Corollary 3.2 *If G is abelian, then the versions 1 and 2 of Newton's algorithm are identical.*

It is easy to see that both algorithms reduce to the standard Newton's method in the Euclidean case, since in that case the exponential map reduces to the identity map on $\mathfrak{g} = G$. The quantity $f'(y_n)$ is then the usual Jacobian matrix of partial derivatives $\dfrac{\partial f}{\partial y_{i,j}}$ evaluated at y_n, and $\mathrm{d}f_{y_n}(u_n)$ is simply the directional derivative in the (tangent) direction $\Delta_{y_n} = L'_{y_n}(u_n) = u_n$.

We also note that the update $y_{n+1} = y_n \cdot \exp(u_n)$ in Version 1 corresponds to the evaluation at $t = 1$ of the integral curve $\gamma_{u_n}(t) = y_n \cdot \exp(tu_n)$ solution of

$$\begin{cases} \gamma'_{u_n}(t) = X_{u_n}(\gamma_{u_n}(t)) \\ \gamma_{u_n}(0) = y_n, \end{cases}$$

i.e., is determined by moving from y_n in the direction of the tangent vector X_{u_n} on γ_{u_n}. In some Riemannian cases, this curve can be shown to be a geodesic (curve of minimal length) on G [5, eq.(34)].

4 Convergence analysis

In the following we consider the radius

$$\rho = \max_{||u||=1} \{t \in \mathbb{R}, su \in \mathcal{N}(0), \forall 0 \leq s \leq t\} > 0 \tag{48}$$

of the maximal ball in $\mathfrak{g}$ centered at 0 and contained in $\mathcal{N}(0)$ (defined in Section 2.3). For $\sigma \leq \rho$ we then introduce the corresponding ball of radius σ around $y \in G$ as

$$B_\sigma(y) = \{z \in G, y^{-1} \cdot z = \exp(u) \in N(e), ||u|| \leq \sigma\}.$$

For completeness we first restate the standard convergence result of Newton's method in Banach spaces which applies to the second version of the algorithm. We suppose that z (see Version 2) is in $B_{\tilde{\sigma}}(y)$. Then v denotes the solution of $\tilde{f}(v) = 0$, related to y by $y = z \cdot \exp(v)$.

Theorem 4.1 *Suppose that*

- *there exists a constant $\tilde{\tau}$ such that $0 < \tilde{\tau} \leq \rho$ and a constant $\tilde{\gamma} \geq 0$ such that*

$$||\mathrm{d}\tilde{f}_u - \mathrm{d}\tilde{f}_w|| \leq \tilde{\gamma}||u - w|| \tag{49}$$

  for all $u, w \in \mathfrak{g}$ such that $||u - v||, ||w - v|| \leq \tilde{\tau}$ (local Lipschitz condition),

- *the map $\mathrm{d}\tilde{f}_v$ is one-to-one and $\mathrm{d}\tilde{f}_v^{-1}$ is bounded with constant $\tilde{\eta}$, i.e., $||\mathrm{d}\tilde{f}_v^{-1}|| \leq \tilde{\eta}$.*

Then there exists a constant $\tilde{\sigma} > 0$ such that, if $||v - v_0|| \leq \tilde{\sigma}$

- *$||v_n - v_0|| \leq \tilde{\sigma}$ for all $n \geq 0$ (the sequence $\{v_n\}_{n\geq 0}$ is well-defined),*
- *$\lim_{n\to\infty} v_n = v$ and $||v - v_{n+1}|| \leq \tilde{\eta}\tilde{\gamma}||v - v_n||^2$ for all $n \geq 0$, i.e., the sequence $\{v_n\}_{n\geq 0}$ converges quadratically to v.*

Theorem 4.1 does not prove, as stated, that the sequence $(y_n)_{n\geq 0}$ obtained from Version 2 converges quadratically to $y = z \cdot \exp(v)$ in the sense of Definition 2.2, since

$$y^{-1} \cdot y_n = \exp(-v) \cdot z^{-1} \cdot z \cdot \exp(v_n) = \exp(-v) \cdot \exp(v_n) \neq \exp(-v + v_n)$$

in general. Nonetheless, it can serve as a basis to show the following result.

Theorem 4.2 (convergence of Version 2) *Suppose that*

- *there exists a constant* τ *such that* $0 < \tau \leq \rho$ *and a constant* $\gamma \geq 0$ *such that*
$$\|\mathrm{d}f_{z\cdot\exp(u)} - \mathrm{d}f_z\| \leq \gamma\|u\| \tag{50}$$
for all $z \in B_\tau(y)$ *and* $u \in \mathfrak{g}$ *such that* $\|u\| \leq \tau$ *(local Lipschitz condition),*
- *the map* $\mathrm{d}f_y$ *is one-to-one and* $\mathrm{d}f_y^{-1}$ *is bounded with constant* η*, i.e.,* $\|\mathrm{d}f_y^{-1}\| \leq \eta$*.*

Then there exists a constant $\sigma > 0$ *and two functions* $h_1(t)$ *and* $h_2(t)$*, analytic for* $|t| \leq \delta/\mu$ *(where* δ *and* μ *are as in Lemma 2.2), with* $h_1(0) = 2h_2(0) = 1$*, such that, if* $y_0 \in B_\sigma(y)$

- $y_n \in B_\sigma(y)$ *for all* $n \geq 0$ *(the sequence* $\{y_n\}_{n\geq 0}$ *is well-defined),*
- $y^{-1}y_n = \exp(v_n)$*,* $\lim_{n\to\infty} v_n = 0$ *and* $\|v_{n+1}\| \leq \eta\left(\gamma h_1(\mu\sigma) + \|\mathrm{d}f_y\|\mu h_2(\mu\sigma)\right)\|v_n\|^2$ *for all* $n \geq 0$*, i.e., the sequence* $\{y_n\}_{n\geq 0}$ *converges quadratically to* y*.*

Proof: We only give a sketch of the proof, and avoid details with the rigorous selection of an appropriate neighborhood of y (e.g., σ) is which the result holds. The result is shown by relating the Lipschitz conditions (49) and (50) as well as $\tilde{\eta}$ and η to one another, then applying Theorem 4.1, finally connecting v_n in Theorem 4.2 (hereby denoted w_n to avoid duplication of symbol) to $v - v_n$ in Theorem 4.1. For our purpose we shall suppose that all manipulations are valid by supposing that σ is "small enough". This implies in particular that $z = y \cdot \exp(v)$ with $\|v\| \leq \sigma$ small (v is the solution to $\tilde{f}(v) = 0$).

• First we have, for u and w such that $\|u\|, \|w\| \leq \sigma$

$$\begin{aligned}
\|\mathrm{d}\tilde{f}_u - \mathrm{d}\tilde{f}_w\| &\leq \|\mathrm{d}f_{z\cdot\exp(u)} \circ \mathrm{dexp}_u - \mathrm{d}f_{z\cdot\exp(w)} \circ \mathrm{dexp}_w\| \\
&\leq \|(\mathrm{d}f_{z\cdot\exp(u)} - \mathrm{d}f_{z\cdot\exp(w)}) \circ \mathrm{dexp}_u + \mathrm{d}f_{z\cdot\exp(w)} \circ (\mathrm{dexp}_u - \mathrm{dexp}_w)\| \\
&\leq \gamma\|\hat{v}\|\,\|\mathrm{dexp}_u\| + \|\mathrm{d}f_{z\cdot\exp(w)}\|\,\|\mathrm{dexp}_u - \mathrm{dexp}_w\|
\end{aligned}$$

where $\hat{v}$ is such that $\exp(\hat{v}) = \exp(-u)\cdot\exp(w)$. Since u and w are small, $\hat{v}$ is also small, and Lemma 2.2, equation (22) yields $\|\hat{v}\| \leq \|-u+w\|h(\mu\beta)$ for some small β. From (21), Lemma 2.1, and the fact that

$$\|\mathrm{d}f_{z\cdot\exp(w)}\| \leq \|\mathrm{d}f_{z\cdot\exp(w)} - \mathrm{d}f_z\| + \|\mathrm{d}f_z - \mathrm{d}f_y\| + \|\mathrm{d}f_y\| \leq \gamma(\|w\| + \|v\|) + \|\mathrm{d}f_y\|,$$

we obtain

$$\|\mathrm{d}\tilde{f}_u - \mathrm{d}\tilde{f}_w\| \leq \left(\gamma\varphi_1(\mu\sigma) + \|\mathrm{d}f_y\|\mu\Phi'(\mu\sigma)\right)\|u-w\| \stackrel{\text{def}}{=} \tilde{\gamma}\|u-w\|$$

with $\varphi_1(t) = h(t)\Theta(t) + 2t\Phi'(t)$ and $\varphi_1(0) = 2\Phi'(0) = 1$.

• Next we have

$$\|\mathrm{d}\tilde{f}_v^{-1}\| = \|\mathrm{dexp}_v^{-1} \circ \mathrm{d}f_y^{-1}\| \leq \|\mathrm{dexp}_v^{-1}\|\,\|\mathrm{d}f_y^{-1}\| \leq \Theta(\mu\sigma)\eta \stackrel{\text{def}}{=} \tilde{\eta}.$$

• Now $y^{-1}y_n = \exp(-v)\cdot\exp(v_n) = \exp(w_n)$ with $w_n = -v + v_n + \frac{1}{2}[-v, v_n] + \ldots$. Using successively Lemma 2.2 (22), Theorem 4.1 and Lemma 2.2 (23) (with $w = v_n$ after rewrting $\exp(v_n) = \exp(v)\cdot\exp(w_n)$), it comes

$$\begin{aligned}
\|w_{n+1}\| &\leq \|v - v_{n+1}\|h(2\mu\sigma) \\
&\leq \tilde{\eta}\tilde{\gamma}\|v - v_n\|^2 h(2\mu\sigma) \\
&\leq \tilde{\eta}\tilde{\gamma}\left(\|w_n\|h(2\mu\sigma)\right)^2 h(2\mu\sigma) \;=\; \eta\left(\gamma h_1(\mu\sigma) + \|\mathrm{d}f_y\|\mu h_2(\mu\sigma)\right)\|v_n\|^2
\end{aligned}$$

with $h_1(t) = (h(2t))^3\Theta(t)\varphi_1(t)$ and $h_2(t) = (h(2t))^3\Theta(t)\Phi'(t)$, for all $n \geq 0$. The verification that $h_1(0) = 2h_2(0) = 1$ is trivial. **Q.E.D.**

Theorem 4.3 (convergence of Version 1) *Suppose that*

- *there exists a constant τ such that $0 < \tau \leq \rho$ and a constant $\gamma \geq 0$ such that*

$$||\mathrm{d}f_{z\cdot\exp(u)} - \mathrm{d}f_z|| \leq \gamma ||u|| \tag{51}$$

  for all $z \in B_\tau(y)$ and $u \in \mathfrak{g}$ such that $||u|| \leq \tau$ (local Lipschitz condition),

- *the map $\mathrm{d}f_y$ is one-to-one and $\mathrm{d}f_y^{-1}$ is bounded with constant η, i.e., $||\mathrm{d}f_y^{-1}|| \leq \eta$.*

Then there exists a constant $\sigma > 0$ such that, if $y_0 \in B_\sigma(y)$

- *$y_n \in B_\sigma(y)$ for all $n \geq 0$ (the sequence $\{y_n\}_{n\geq 0}$ is well-defined),*
- *$y^{-1}y_n = \exp(v_n)$, $\lim_{n\to\infty} v_n = 0$ and $||v_{n+1}|| \leq \beta\gamma h(3\mu\sigma)||v_n||^2$ for all $n \geq 0$, where h is as in Lemma 2.2, i.e., the sequence $\{y_n\}_{n\geq 0}$ converges quadratically to y.*

Proof of Theorem 4.3: Let $\delta > 0$ and $h(-)$ be as in Lemma 2.2, $M = \sup_{t\in[0,\delta]} h(t) = h(\delta) \geq 1$ and pick $\sigma = \min\left(\frac{1}{2\eta\gamma M}, \frac{\tau}{2}, \frac{\delta}{3\mu}\right)$, where μ is defined in (18).

Consider $y_0 \in B_\sigma(y)$. Then $y^{-1}\cdot y_0 = \exp(v_0)$ for some $v_0 \in \mathfrak{g}$ such that $||v_0|| \leq \sigma$. First note that

$$||\mathrm{Id}_{\mathfrak{g}} - \mathrm{d}f_y^{-1}\circ \mathrm{d}f_{y_0}|| = ||\mathrm{d}f_y^{-1}\circ\left(\mathrm{d}f_y - \mathrm{d}f_{y_0}\right)|| \leq ||\mathrm{d}f_y^{-1}||\, ||\mathrm{d}f_y - \mathrm{d}f_{y_0}|| \leq \eta\gamma||v_0|| \leq \frac{1}{2M} \leq \frac{1}{2}.$$

Thus $\mathrm{d}f_{y_0}$ is bijective by virtue of the Banach lemma [6, p.59]. Furthermore,

$$||\mathrm{d}f_{y_0}^{-1}|| \leq \frac{||\mathrm{d}f_y^{-1}||}{1 - ||\mathrm{Id}_{\mathfrak{g}} - \mathrm{d}f_y^{-1}\circ\mathrm{d}f_{y_0}||} \leq 2||\mathrm{d}f_y^{-1}|| \leq 2\eta.$$

Then, if u_0 is defined by (28), we obtain, using Corollary 3.1,

$$\begin{aligned} u_0 + v_0 &= \mathrm{d}f_{y_0}^{-1}\left(f(y) - f(y_0) + \mathrm{d}f_{y_0}(v_0)\right) \\ &= \mathrm{d}f_{y_0}^{-1}\int_0^1 \left(\frac{d}{dt}f(y_0\cdot\exp(-tv_0)) + \mathrm{d}f_{y_0}(v_0)\right) dt \\ &= \mathrm{d}f_{y_0}^{-1}\int_0^1 \left(-\mathrm{d}f_{y_0\cdot\exp(-tv_0)}(v_0) + \mathrm{d}f_{y_0}(v_0)\right) dt \end{aligned}$$

so that (note that $||tv_0|| \leq \sigma$ for $t \in [0,1]$)

$$\begin{aligned} ||u_0 + v_0|| &\leq ||\mathrm{d}f_{y_0}^{-1}|| \int_0^1 ||\mathrm{d}f_{y_0}(v_0) - \mathrm{d}f_{y_0\cdot\exp(-tv_0)}(v_0)||\, dt \\ &\leq 2\eta\int_0^1 \gamma||tv_0||\,||v_0||\, dt \\ &= \eta\gamma||v_0||^2. \end{aligned}$$

In particular,

$$||u_0|| \leq ||v_0|| + ||u_0 + v_0|| \leq \left(1 + \frac{1}{2M}\right)||v_0|| \leq \frac{3}{2}||v_0|| \leq \frac{3}{2}\sigma \leq \frac{3}{4}\tau < \rho,$$

so that $u_0 \in \mathcal{N}(0)$. Thus

$$y^{-1}y_1 = \exp(v_0)\cdot\exp(u_0) = \exp(v_1)$$

with (see (12))

$$v_1 = v_0 + u_0 + \frac{1}{2}[v_0, u_0] + \ldots \in \mathfrak{g}.$$

The inequality (22) in Lemma (2.2) then yields, with $\beta = 3\sigma \leq \delta/\mu$,

$$\|v_1\| \leq \|v_0 + u_0\|\ h(3\mu\sigma) \leq h(3\mu\sigma)\eta\gamma\|v_0\|^2.$$

In particular, $\|v_1\| \leq \eta\gamma M\|v_0\|^2 \leq \frac{1}{2}\|v_0\| \leq \sigma$, so that $y_1 \in B_\sigma(y)$ as well. The result follows by induction (the fact that $v_n \to 0$ as $n \to \infty$ is a consequence of the fact that $\|v_{n+1}\| \leq \frac{1}{2}\|v_n\|$ for all n). **Q.E.D.**

Remark 4.1 *The Lipschitz condition (50)-(51) arises naturally in the proof of Theorem 4.3, and reduces to the standard Lipschitz condition of the type (49) in the Euclidean case (with $u \leftarrow z + u$ and $v \leftarrow z$).*

5 Numerical examples

In this section we consider the initial value problem

$$\begin{cases} y' = y \cdot g(y) \\ y(0) = y^{(0)} \end{cases}$$

for $y \in G = \mathrm{O}(N, \mathbb{R})$, where $g(y) \in \mathfrak{g} = \mathfrak{so}(N, \mathbb{R})$ and $y^{(0)}$ is a random starting point.

The application of one step of the backward Euler method on (54) leads to the fixed-point problem

$$y = y^{(0)} \cdot \exp(hg(y))$$

which was used in Examples 3.1 and 3.2 (with $z = y^{(0)}$), where h represents the size of the discretization step.

The quantity u_n determined by (28) was computed by solving a linear system of size $q = \dim(G) = \dim(\mathfrak{g}) = \frac{N(N-1)}{2}$ resulting from the expansion of $u_n = \sum_{i=1}^{q} \alpha_i \mathbf{b}_i$ onto a basis of $\mathfrak{g}$. This system reads

$$\mathbb{D}\alpha + \mathbb{F} = \mathbf{0},$$

where

$$\mathbb{D} = (\mathbb{D}_{i,j})_{1 \leq i,j \leq q}, \quad \alpha = (\alpha_1, \ldots, \alpha_q)^T, \quad \mathbb{F} = (f_1, \ldots, f_q)^T$$

with

$$\mathrm{d}f_{y_n}(\mathbf{b}_i) = \sum_{j=1}^{q} \mathbb{D}_{i,j}\mathbf{b}_j \quad \text{and} \quad f(y_n) = \exp^{-1}(y^{(0)-1}y_n) - hg(y_n) = \sum_{i=1}^{q} f_i \mathbf{b}_i.$$

A similar strategy was used to solve (43).

The two versions of the Newton iteration are compared to the fixed-point iteration (2) for a range of step-sizes $10^{-3} \simeq 2^{-10} \leq h \leq 2^{10} = 1024$ and dimensions $2 \leq N \leq 15$. In all cases, the algorithm is started with $y_0 = y^{(0)}$. Convergence is achieved in all cases when

$$\|f(y_n)\|_2 \leq tol = 10^{-13}$$

(note that convergence for Version 2 of the Newton algorithm would normally be measured by $\|\tilde{f}(v_n)\|_2 \leq tol$). A limit of 100 iterations is set for convergence.

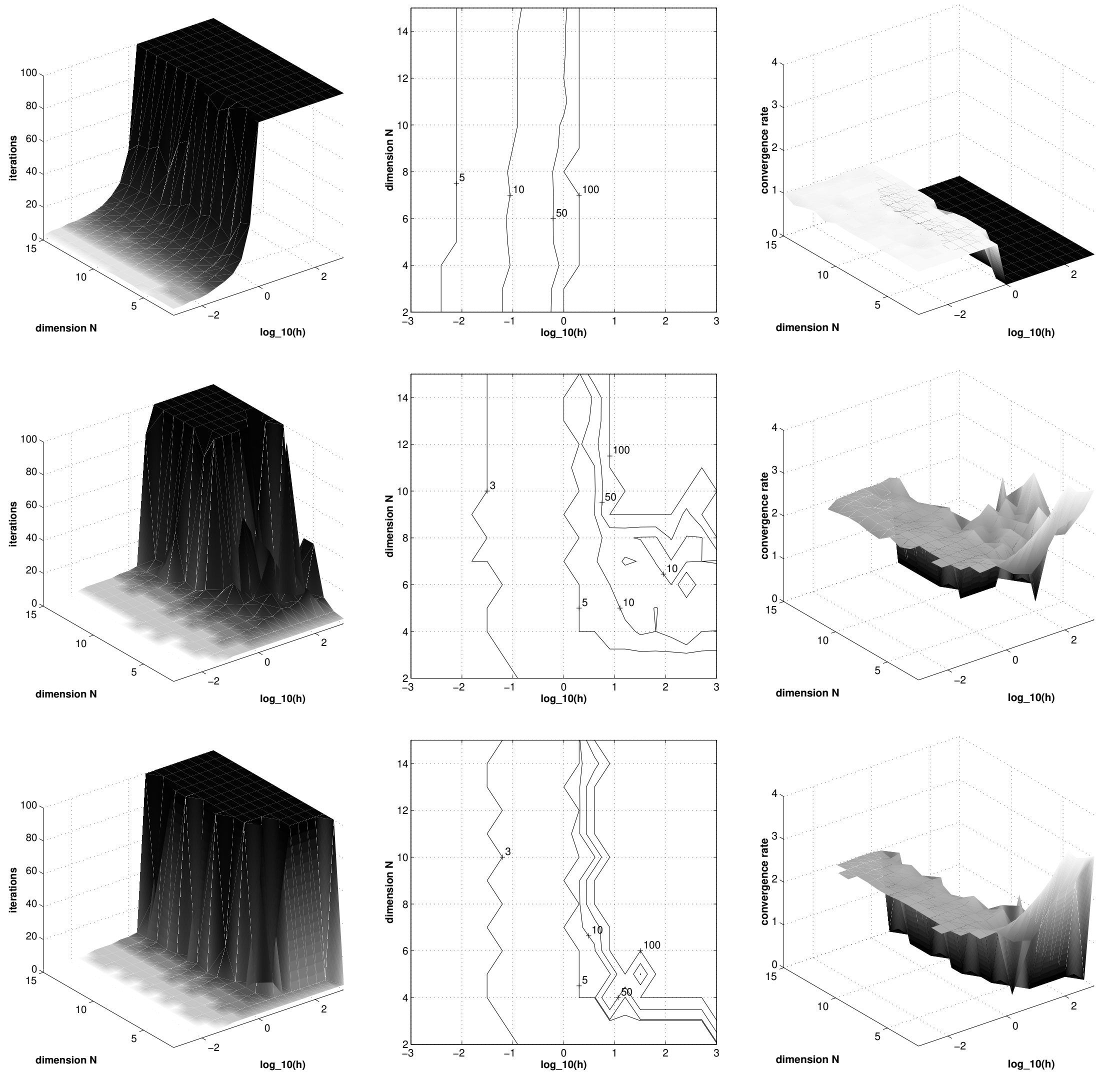

Figure 2: Number of iterations (left and center) and rate of convergence (right) to reach convergence to $tol = 10^{-13}$ for the fixed-point iteration (top), Version 1 (middle) and Version 2 (bottom) of the Newton algorithm applied to $g(y) = u - u^T$, with $\mathtt{u} = \mathrm{diag}(\mathrm{diag}(y,1),1)$, as a function of the step-size h and the dimension N of the Lie group G.

The convergence rate for each method is determined by considering the ratio

$$r = \log\left(\frac{\|f(y_{p-1})\|_2}{\|f(y_{p-2})\|_2}\right) \Big/ \log\left(\frac{\|f(y_{p-2})\|_2}{\|f(y_{p-3})\|_2}\right)$$

where p is the number of iterations necessary to reach convergence. This avoids wrong estimations of the rate of convergence due to the fact that we may have $\|f(y_p)\|_2 = \mathcal{O}(\mathbf{u})$, but is only possible if $p \geq 3$. The rate of convergence is not computed if $p < 3$.

• In the first example, g and $y^{(0)}$ are given by (in MATLAB notations)

$$\mathtt{g(y)} = \mathtt{diag(diag(y,1),1)} - \mathtt{diag(diag(y,1),-1)}$$

and

$$\mathtt{y}^{(0)} = \mathtt{qr(rand(N,N))},$$

respectively. This test problem was used in [9] and [15] to investigate the numerical order of explicit integrators on Lie groups. Note that g is linear in y, so that, using (8),

$$g(y \cdot \exp(tu)) = g\left(y \sum_{n \geq 0} \frac{t^n}{n!} u^n\right) = \sum_{n \geq 0} \frac{t^n}{n!} g(yu^n)$$

and

$$\mathrm{d}g_y(u) = \left.\frac{d}{dt}\right|_{t=0} g((y \cdot \exp(tu)) = g(yu),$$

yielding the explicit form

$$\mathrm{d}f_y(u) = \mathrm{dexp}^{-1}_{v(y)}(u) - h\mathrm{d}g_y(u) = u + \frac{1}{2}[\exp^{-1}(y^{(0)-1}y), u] + \ldots - hg(yu).$$

The numerical experiments were however conducted by using the finite difference approximation (41).

It is easy to check that the conditions of Theorems 4.2-4.3 are satisfied with $\gamma = 1 + \mathcal{O}(h)$ and $\eta = 1 + \mathcal{O}(h)$ (since $\mathrm{d}f_y(u) \simeq u - hg(yu)$).

Figure 2 shows that the Newton iteration (Version 1) converges sensibly better for larger step-sizes than the Version 2 for smaller values of N ($N = 3$ is relevant in practical problems in mechanics), while the fixed-point iteration does not converge. Note that in the case $N = 3$ the cost of both versions of the Newton algorithm is similar (one can use Rodrigues formula and its inverse to compute $\exp(u)$ and $\exp^{-1}(y)$).

As expected, the Newton iterations converge quadratically in the range of h- and $N-$ values where the methods converge, compared to the linear convergence of the fixed-point iteration.

- The second example is similar to the first one, except that g is now defined by

$$g(y) = u - u^T, \quad u = \sin(y)(2y - 5y^2).$$

The numerical results in Figure 3 show a behavior of the three algorithms similar to what occurred the first example. Again, the first version of the algorithm seems to converge better for smaller size problems.

From these experiments it appears that

- for small N, the Version 1 of Newton's method converges for a larger range of step-sizes compared to Version 2 and the fixed-point iteration.
- for larger N, the Newton iteration may not be competitive compared to the fixed-point iteration, but is guaranteed to converge for smaller step-sizes, while the fixed-point method is not.

6 Concluding remarks

We have presented two versions of the Newton iteration for solving equations of the type $f(y) = 0$ where f is a map from a Lie group to its coresponding Lie algebra. The main motivation has been to facilitate the use of implicit methods for solving ordinary differential equations on manifolds, an area that has recently been subject to considerable interest. Clearly, the approach presented here is only applicable to the case where the manifold in question is a Lie group. This is however not an unimportant case, there are many examples from applications of initial value problems whose solution evolve on a Lie group. For instance, the matrix groups $\mathrm{SO}(3, \mathbb{R})$ and $\mathrm{SL}(3, \mathbb{R})$ are frequently used to model rotations and volume preservation respectively in 3D space. It should be the subject future research to investigate the behavior of the proposed algorithms applied to such real life problems.

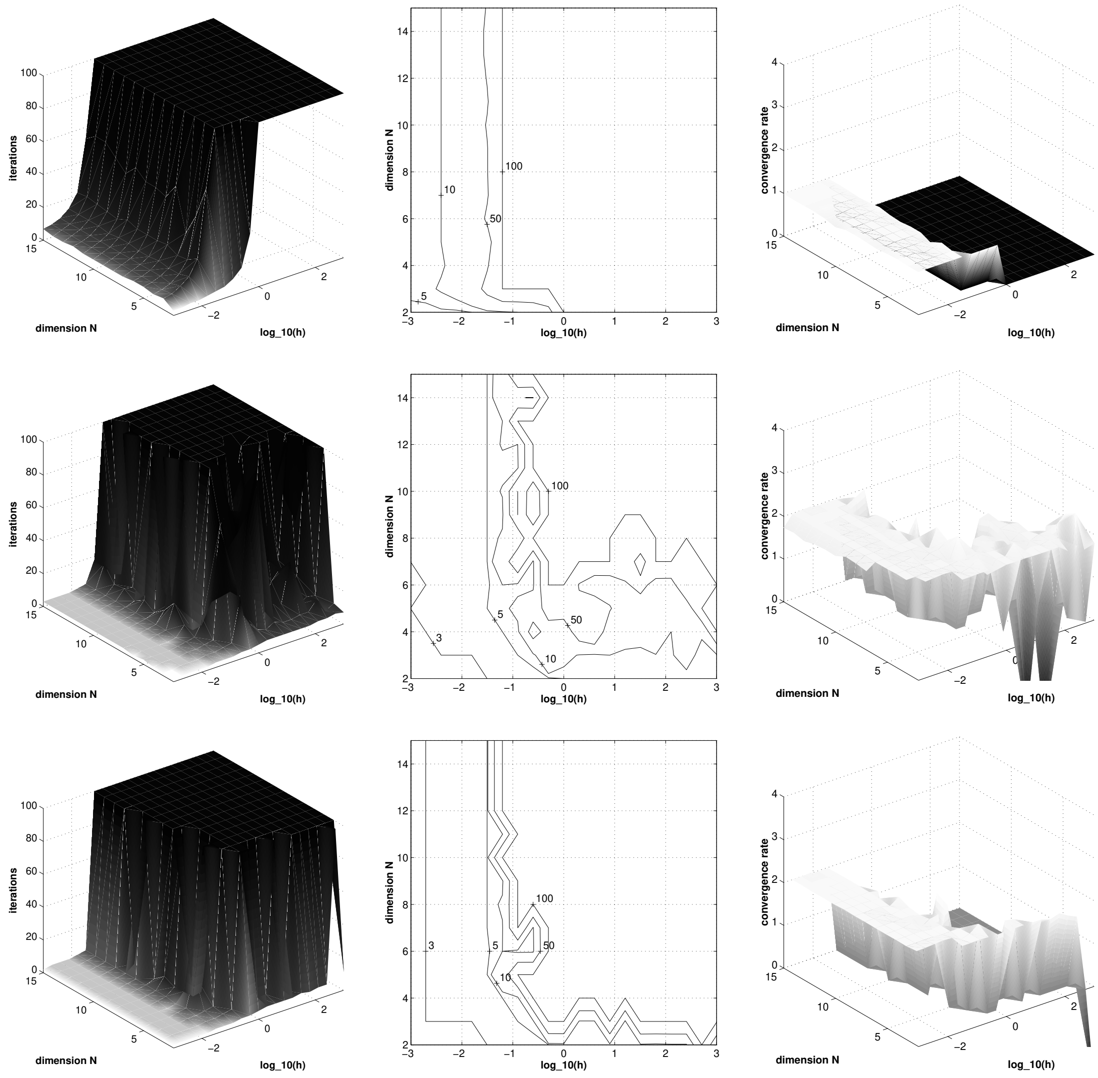

Figure 3: Number of iterations (left and center) and rate of convergence (right) to reach convergence to $tol = 10^{-13}$ for the fixed-point iteration (top), Version 1 (middle) and Version 2 (bottom) of the Newton algorithm applied to $g(y) = u - u^T$ with $u = \sin(y)(2y - 5y^2)$, as a function of the step-size h and the dimension N of the Lie group G.

- Generalization to homogeneous manifolds:

  A more general class of manifolds that contains the Lie groups as a special case are the *homogeneous manifolds*. Such a manifold consists of a Lie group G that acts transitively on a manifold $\mathcal{M}$ through a group action $\lambda : G \times \mathcal{M} \to \mathcal{M}$. It is a well known result [3] that $\mathcal{M}$ is naturally diffeomorphic to G/G_m for any $m \in \mathcal{M}$, the left cosets of the *stabilizer subgroup* $G_m = \{y \in G, \lambda(y, m) = m\}$. In the case where $G_m = \{e\}$, the trivial group, one obtains $\mathcal{M} = G$ and the group action λ reduces to the group multiplication $\cdot$ in G. However, if G_m contains more elements, the induced map $\lambda_m : y \mapsto \lambda(y, m)$ is not injective. The integration methods proposed by Crouch and Grossman [4] can be interpreted in the context of homogeneous manifolds, see [11]. Certain difficulties are encountered if a straightforward generalization of the Newton iteration presented in this paper is attempted. The obvious approach of identifying the iteration sequence on $\mathcal{M}$ with elements of the Lie algebra of G is impeded by the fact that there is no longer a unique local correspondence between points in $\mathcal{M}$ and elements of G relative to a reference point $m \in \mathcal{M}$, as is the case between G and $\mathfrak{g}$ via the exponential mapping around 0. This may suggest that for homogeneous manifolds, it may be more useful to consider representations in $T\mathcal{M}_m$ where m is a fixed reference point on $\mathcal{M}$. This amounts to the introduction of a Riemannian metric on $T\mathcal{M}$. This idea conforms well with the work of Edelman et al. [5].

- Application to high-order implicit integrators:

  To illustrate the use of the new algorithms for solving initial value problems with implicit methods, we have used the implicit Euler method. It should be noted that the technique we use can be extended in a straightforward way to general RK methods of Crouch-Grossman type for Lie groups. For instance, with DIRK type Crouch-Grossman methods one simply needs to solve, in sequence, equations of the type

$$\begin{aligned} & Y_i = Z_i \cdot \exp(ha_{ii} g(Y_i)), \\ \text{where} \quad & Z_i = y^{(k)} \cdot \exp(ha_{i,1} g(Y_1)) \cdots \exp(ha_{i,i-1} g(Y_{i-1})), \; i = 1, \ldots, s. \end{aligned}$$

Finally, it should be noted that there are still many interesting problems to be solved in order to obtain streamlined implementations of the Newton iteration on Lie groups. Although we believe that the abstract formulation of the iteration technique is satisfactory, several challenges remain on the more detailed level. Computing the differential of f and solving the iteration equations are examples of such challenges. Another difficulty is that in the evaluation of the function f in version 1, one makes use of the inverse of the exponential map. If G is a matrix Lie group, the **logm** function in Matlab may be used for this purpose, but the computation is fairly expensive. However, there are special cases in which an explicit formula is available for the inverse of the exponential. An example of this is $\mathrm{SO}(3, \mathbb{R})$ where the logarithm can be obtained in a very stable way and very efficiently (with less arithmetic operations than the Rodrigues formula which is commonly used for computing the matrix exponential in $\mathfrak{so}(3)$).

Although many questions are still to be addressed regarding the solution of nonlinear equations on manifolds, we believe that the numerical results from the algorithms proposed in this paper indicate that this new approach may be a worth while alternative to existing methods.

Acknowledgements

This work was completed while the second author was visiting the Numerical group at Institutt for matematiske fag at NTNU. He thanks everyone for making this collaboration possible and pleasant. Both authors also thank Hans Munthe-Kaas for valuable discussions and inputs.

Appendix

1. Proof of Lemma 2.2:

The proof of [13, Theorem 2.15.4] fails to provide a bound on $||w||$ in terms of $||u||$ and $||v||$ but contains most of the elements necessary for showing the result.

W.l.o.g., we suppose that $u \neq 0$ (the result is trivial for $u = 0$). Following [13], we write

$$w = \sum_{n \geq 1} c_n,$$

where c_n represents the contribution in w of all homogeneous terms of degree n in u and v, i.e.,

$$c_1 = u + v$$

$$
\begin{aligned}
c_2 &= \frac{1}{2}[u, v] \\
c_3 &= \frac{1}{12}([u,[u,v]] + [v,[v,u]]) \\
&\ldots
\end{aligned}
$$

The terms c_n can be computed inductively via the recursion formula [13, Lemma 2.15.3]

$$(n+1)c_{n+1} = \frac{1}{2}[u-v, c_n] + \sum_{p\geq 1, 2p\leq n} F_{2p} \sum_{k>0, |k|=n} [c_{k_1}, [\ldots [c_{k_{2p}}, u+v]\ldots], \tag{52}$$

where $k = (k_1, \ldots, k_{2p})$ is a multi-index with $|k| = k_1 + \ldots + k_{2p}$, such that $k_\ell > 0$ for $\ell = 1, \ldots, 2p$. The constants F_{2p}, $p \geq 1$, are defined by the series expansion of the even function

$$\frac{t}{1-e^{-t}} - \frac{t}{2} = 1 + \sum_{p\geq 1} F_{2p} t^{2p}. \tag{53}$$

Note that the series is well-defined and absolutely convergent for $|t| < 2\pi$. From the theory of differential equations, we know that the initial value problem

$$\begin{cases} \hat{h}'(t) = \dfrac{\hat{h}(t)}{2} + 1 + \displaystyle\sum_{p\geq 1} |F_{2p}|\, \hat{h}(t)^{2p} \\ \hat{h}(0) = 0 \end{cases} \tag{54}$$

admits a solution $\hat{h}(t)$ which is analytic for all $|t| \leq \delta$ for some (universal) constant $\delta > 0$, i.e.,

$$\hat{h}(t) = \sum_{n\geq 1} \hat{H}_n t^n. \tag{55}$$

Note that substituting (55) in (54) yields

$$(n+1)\hat{H}_{n+1} = \frac{\hat{H}_n}{2} + \sum_{p\geq 1} |F_{2p}| \sum_{k>0, |k|=n} \hat{H}_{k_1} \ldots \hat{H}_{k_{2p}} \tag{56}$$

for $n \geq 1$ with $\hat{H}_1 = 0$. In particular $\hat{H}_n > 0$ for $n \geq 1$.

Define $h(t) = \dfrac{\hat{h}(t)}{t}$, so that $h(0) = \hat{h}'(1) = \hat{H}_1 = 1$. Now let $||u|| + ||v|| \leq \beta$ with $\beta \leq \delta/\mu$, so that $h(\mu\beta)$ is well-defined.

- We show by induction that

$$||c_n|| \leq ||u+v|| \hat{H}_n (\mu\beta)^{n-1} \tag{57}$$

for $n \geq 1$. This is certainly true for $n = 1$ since $\hat{H}_1 = 1$. By induction, (52) yields (with $||u - v|| \leq ||u|| + ||v|| \leq \beta$)

$$
\begin{aligned}
(n+1)||c_{n+1}|| &\leq \frac{\mu}{2}||u-v||\, ||c_n|| + \sum_{p\geq 1, 2p\leq n} |F_{2p}| \sum_{k>0, |k|=n} \mu^{2p} ||c_{k_1}|| \ldots ||c_{k_{2p}}||\, ||u+v|| \\
&\leq ||u+v||(\mu\beta)^n \frac{\hat{H}_n}{2} + \sum_{p\geq 1, 2p\leq n} |F_{2p}|\, ||u+v||^{2p+1} \mu^n \beta^{n-2p} \sum_{k>0, |k|=n} \hat{H}_{k_1} \ldots \hat{H}_{k_{2p}} \\
&\leq (n+1)\hat{H}_{n+1} ||u+v|| (\mu\beta)^n,
\end{aligned}
$$

in view of (56), thus showing that (57) holds for $n \geq 1$.

We then obtain

$$||w|| \leq \sum_{n\geq 1} ||c_n|| \leq ||u+v|| \sum_{n\geq 1} \hat{H}_n (\mu\beta)^{n-1} = ||u+v||\ h(\mu\beta).$$

• Suppose now in addition that $||u|| \le \beta/2$. Similarly, we show by induction that

$$||c_n|| \le ||v|| \hat{H}_n (\mu\beta)^{n-1} \tag{58}$$

for $n \ge 2$. The result holds for $n = 2$ since $\hat{H}_2 = \frac{\hat{h}''(0)}{2} = \frac{1}{4}$ while

$$||c_2|| = \frac{1}{2}||[u,v]|| \le \frac{\mu}{2}||u||\,||v|| \le \frac{\mu\beta}{4}||v||.$$

By induction, (52) yields (with $||u-v|| \le ||u|| + ||v|| \le \beta$ and $||u+v|| \le ||u|| + ||v|| \le \beta$)

$$\begin{aligned}
(n+1)||c_{n+1}|| &\le \frac{\mu}{2}||u-v||\,||c_n|| + \sum_{p\ge 1, 2p\le n} |F_{2p}| \sum_{k>0, |k|=n} \mu^{2p} ||c_{k_1}|| \ldots ||c_{k_{2p}}||\,||u+v|| \\
&\le ||v||(\mu\beta)^n \frac{\hat{H}_n}{2} + \sum_{p\ge 1, 2p\le n} |F_{2p}|\,\mu^{2p}\beta^{2p-q+1}||v||^q \sum_{k>0,|k|=n} \hat{H}_{k_1} \ldots \hat{H}_{k_{2p}} (\mu\beta)^{\alpha}
\end{aligned}$$

where $\alpha = k_{i_1} + \ldots + k_{i_q} - q$, with $k_{i_j} \ge 2$ for $j = 1, \ldots, q$. Note that $q > 0$ since $k_{2p} \ge 2$, for $[c_{k_{2p}}, u+v] = [c_1, u+v] = 0$ in (52) otherwise. Also, $\alpha = k_1 + \ldots + k_{2p} - (2p-q) - q = n - 2p$. Thus (note that $||v|| \le \beta$)

$$\begin{aligned}
(n+1)||c_{n+1}|| &\le ||v||(\mu\beta)^n \left(\frac{\hat{H}_n}{2} + \sum_{p\ge 1, 2p\le n} |F_{2p}| \sum_{k>0,|k|=n} \hat{H}_{k_1} \ldots \hat{H}_{k_{2p}} \right) \\
&= (n+1)\hat{H}_{n+1}||v||(\mu\beta)^n,
\end{aligned}$$

which shows that (58) holds for $n \ge 2$. Finally,

$$||w-u|| = ||v + \sum_{n\ge 2} c_n|| \le ||v|| + \sum_{n\ge 2} ||c_n|| \le ||v|| \left(1 + \sum_{n\ge 2} \hat{H}_n (\mu\beta)^{n-1}\right) = ||v||\, h(\mu\beta).$$

Q.E.D.

Note that $\hat{h}(t) \ge 0$, so that $\hat{h}'(t) \ge \dfrac{\hat{h}(t)}{2} + 1$ and Gronwall's lemma yields

$$h(t) \ge \frac{e^{\frac{t}{2}} - 1}{\frac{t}{2}}$$

for $0 \le t \le \delta$.

2. MATLAB **files:**

Driver routine

```
% solve y = z exp(hg(y))

global h z ndim

% define equations f(y) = 0 and ft(v) = 0
% ndim   h
%  4    >10    only V1 seems to converge
ndim = 3;
h     = 20000;
rand('seed',0);
[z,r] = qr(rand(ndim));

% Fixed-point iteration (FP)
% y_{n+1} = z*exp(hg(y_n))
tol     = 1e-13;
itmax  = 100;
y = z;
[y0,cv0] = fxdpt('g',y,z,h,'f',tol,itmax);

% Newton's method (V1)
% write y = z exp(v(y))
% solve for f(y) = v(y) - hg(y) = 0
tol     = 1e-13;
itmax  = 100;
```

```
y      = z;
[y1,cv1] = anewton1('f',y,z,tol,itmax);

% Newton's method (V2)
% write y = z exp(v)
% solve for ft(v) = v - hg(z*exp(v)) = 0
tol    = 1e-13;
itmax  = 100;
v      = zeros(ndim);
[y2,cv2] = anewton2('ft',v,z,tol,itmax);

% output
...
```

```
function w = add(u,v)
% addition in Lie algebra
w = u+v;
```

```
function [y,cv] = anewton1(f,y,z,tol,itmax)
% Newton's method in Lie group (Version 1)
fy = feval(f,y);
e = anorm(fy);
cv = [0 e];
k = 0;
while (e>tol & k<itmax)
  F = -get_c(fy);
  D = df(f,y,z);
  u = put_c(D\F);
  y = mult(y,expmap(u));
  fy = feval(f,y);
  e = anorm(fy);
  k = k+1;
  cv = [cv;k e];
end
if (e>tol | k==itmax)
  disp(['WARNING: no convergence within ' ...
        ' tol = ' num2str(tol) ' after '  ...
        int2str(itmax) ' iterations.'])
end
```

```
function [y,cv] = anewton2(ft,v,z,tol,itmax)
% Newton's method in Lie group (Version 2)
ftv = feval(ft,v);
e = anorm(ftv);
%only for convergence comparison:
%y = mult(z,expmap(v));
%fy = feval('f',y);
%e = anorm(fy);
cv = [0 e];
k = 0;
while (e>tol & k<itmax)
  F = -get_c(ftv);
  D = dft(ft,v,z);
  u = put_c(D\F);
  v = add(v,u);
  ftv = feval(ft,v);
  e = anorm(ftv);
  %only for convergence comparison:
  %y = mult(z,expmap(v));
  %fy = feval('f',y);
  %e = anorm(fy);
  k = k+1;
  cv = [cv;k e];
end
y = mult(z,expmap(v));
if (e>tol | k==itmax)
  disp(['WARNING: no convergence within ' ...
        ' tol = ' num2str(tol) ' after '  ...
        int2str(itmax) ' iterations.'])
end
```

```
function nrm = anorm(u)
% norm in Lie algebra
nrm = norm(u,2);
```

```
function D = df(f,y,z)
% Jacobian of f at y
global ndim
p = ndim*(ndim-1)/2; % dimension of Lie algebra
D = zeros(p,p);
fy = feval(f,y);
t = sqrt(eps);
for i = 1:p
  cb = zeros(p,1);
  cb(i) = 1;
  b = put_c(cb);
  w = (feval(f,mult(y,expmap(t*b)))-fy)/t;
  D(:,i) = get_c(w);
end
```

```
function D = dft(ft,u,z)
% Jacobian of ft at u
p = length(get_c(u));
D = zeros(p,p);
ftu = feval(ft,u);
t = sqrt(eps);
for i = 1:p
  cb = zeros(p,1);
  cb(i) = 1;
  b = put_c(cb);
  w = (feval(ft,add(u,t*b))-ftu)/t;
  D(:,i) = get_c(w);
end
```

```
function y = expmap(u)
% exponential map from Lie algebra to Lie group
y = expm(u);
```

```
function fy = f(y)
global h z
fy = logmap(multinv(z,y))-h*feval('g',y);
```

```
function ftv = ft(v)
global h z
ftv = v-h*feval('g',mult(z,expmap(v)));
```

```
function [y,cv] = fxdpt(g,y,z,h,f,tol,itmax)
% Fixed-point iteration
fy = feval(f,y);
e = anorm(fy);
cv = [0 e];
k = 0;
```

```
while (e>tol & k<itmax)
  y = mult(z,expmap(h*feval(g,y)));
  fy = feval(f,y);
  e = anorm(fy);
  k = k+1;
  cv = [cv;k e];
end
if (e>tol | k==itmax)
  disp(['WARNING: no convergence within ' ...
        ' tol = ' num2str(tol) ' after '  ...
        int2str(itmax) ' iterations.'])
end
```

```
function gy = g(y)
x = diag(y,1);
gy = diag(x,1)-diag(x,-1);
```

```
function w = get_c(u)
% coordinates w (given as a vector of R^n) of
% element u in the Lie algebra onto a basis of
% the Lie algebra.
% this example works for any u in so(p)
% ex: p = 3, w = [u(1,2:3) u(2,3:3)]';
p = size(u,1);
w = zeros(p*(p-1)/2,1);
k2 = 0;
for i = 1:p-1
  k1 = k2+1;
  k2 = k1+p-i-1;
  w(k1:k2) = u(i,(i+1):p);
end
u = u-u';
```

```
function u = logmap(y)
% inverse of exponential map
u = real(logm(y));
v = triu(u,1);
u = v-v';
```

```
function z = mult(x,y)
% multiplication z = x*y in Lie group
z = x*y;
```

```
function z = multinv(x,y)
% inverse and multiplication z = x^(-1)*y
% in Lie group
z = x\y;
```

```
function u = put_c(w)
% element u of Lie algebra as a function of its
% coordinates w (given as a vector of R^n) onto
% a basis of the Lie algebra.
% this example works for any u in so(p)
% ex: p = 3, u = [0 w(1:2);0 0 w(3:3);0 0 0];
w = (w(:))';
p = round(1+sqrt(1+8*length(w)))/2;
u = zeros(p,p);
k2 = 0;
for i = 1:p-1
  k1 = k2+1;
  k2 = k1+p-i-1;
  u(i,(i+1):p) = w(k1:k2);
end
u = u-u';
```

References

[1] R. Abraham, J. E. Marsden and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, Springer-Verlag (1980).

[2] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd edition, Springer GTM no. 60, New York, (1989).

[3] R. L. Bryant, *An Introduction to Lie Groups and Symplectic Geometry*, in Geometry and Quantum field Theory, Eds. Freed and Uhlenbeck, AMS IAS/Park City math. series vol. 1 (1995).

[4] P. E. Crouch and R. Grossman, *Numerical Integration of Ordinary Differential Equations on Manifolds*, J. Nonlinear Sci. Vol.3 (1993) 1-33.

[5] A. Edelman, T. Arias and S. T. Smith, *Conjugate Gradient and Newton's Method on the Grassmann and Stiefel Manifolds*, Preprint, Department of Mathematics, MIT (1996).

[6] G. Golub and C. Van Loan, *Matrix Computations*, John Hopkins University Press (1992).

[7] A. Iserles, *Solving Linear Ordinary Differential Equations by Exponentials of Iterated Commutators*, Numer. Math. 45 (1984) 183-199.

[8] A. Marthinsen and B. Owren, *Order Conditions for Integration Methods Based on Rigid Frames*, Preprint, Department of Mathematical Sciences, NTNU (1996).

[9] H. Munthe-Kaas, *Runge-Kutta Methods on Lie Groups*, Preprint, Department of Informatics, University of Bergen (1996).

[10] H. Munthe-Kaas, *Lie-Butcher Theory for Runge-Kutta Methods*, BIT 35 (1996) 572-587.

[11] H. Munthe-Kaas and A. Zanna, *Numerical Integration of Differential Equations on Homogeneous Manifolds*, Preprint, Department of Informatics, University of Bergen (1996).

[12] P. J. Olver, *Applications of Lie Groups to Differential Equations*, 2nd edition, Springer GTM no. 107, New York (1993).

[13] V. S. Varadarajan, *Lie Groups, Lie Algebras and Their Representations*, Springer GTM no. 102, New York (1984).

[14] F. W. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, Springer GTM no. 94, New York (1983).

[15] A. Zanna, *The Method of Iterated Commutators for Ordinary Differential Equations on Lie Groups*, Tech. Rept, University of Cambridge, DAMTP 1996/NA12 (1996).