Accurate small deformation exponential approximant to integrate large velocity fields: Application to image registration

Sebastiano Ferraris, Marco Lorenzi, Pankaj Daga, Marc Modat and Tom Vercauteren
Translational Imaging Group, Centre for Medical Image Computing,
Wolfson House, 4 Stephenson Way, University College London.
Email: s.ferraris@ucl.ac.uk, m.lorenzi@ucl.ac.uk, p.daga@ucl.ac.uk, m.modat@ucl.ac.uk, t.vercauteren@ucl.ac.uk

Abstract

One of the basic components of diffeomorphic image registration algorithms based on velocity fields is the numerical method used to integrate velocity parameters and obtain spatial transformations as displacement fields. When the input velocity field does not depend on the time parameter, the solution is often referred to as the Lie exponential of the velocity field. In this work, we present an integration method for its numerical computation based both on a generalization of the scaling and squaring algorithm *and on a class of numerical integrators aimed to solve systems of ordinary differential equations called* exponential integrators*. This new method led to the introduction of three numerical integrators, and the subsequent validation are performed on synthetic deformations and real medical images.*

1. Introduction

Diffeomorphisms in medical image registration: There are multiple medical applications where it is interesting to combine and compare information obtained from different patients' images: in longitudinal studies, when a temporal sequence of medical images of the same subject is studied, in cross-sectional studies, when images are compared across subjects, and when images are acquired using different modalities [22]. In brain imaging, for example, registration techniques are widely used to correct the subject motion and, in longitudinal scans, to help in comparing and analysing anatomical differences. In the practical implementations of image registration algorithms, one of the crucial choice is the set of constraints that reflects the geometrical properties that are expected to remain unchanged between scans. A reasonable choice for the set of spatial transformations modelling continuous deformations between images, is the group of diffeomorphisms.

Parametrisation of diffeomorphisms: Two main problems do arise in the practical implementation of diffeomorphic image registration frameworks: their parametrization and the statistical analysis [13]. A sensible choice to partially solve these technical issues is to parametrize each diffeomorphism with its tangent vector field, that makes them easier to be manipulated, and where the computation of statistics is simplified. Two possible options are available according to the dependence of the vector field on the time parameter.

From time-varying to stationary velocity fields: While theoretically the parametrisation in the tangent space partially solves the above mentioned problems, difficulties emerges in its computation. An analytic (or exact) solution of the differential equation system that provides the related geometrical transformation of the vector field, is almost never available. To approximate it, it is necessarily to rely on numerical schemes, called *integrators*, such as the widely used Euler method or Runge Kutta 4 [7]. These methods are applied in medical imaging both when dealing with vector fields that are time-dependent (time-varying velocity field or TVVF) or time-independent (called stationary velocity fields or SVF). TVVFs appear in diffeomorphic image registration as a parametrization model of the deformations, where the geodesic motion on the diffeomorphism group is defined by the Euler-Poincaré equations [5, 20]. Authors of [3] have indicated that restricting the parametrization to the SVF, allows to use an algorithm called (classical) scaling and squaring, that is in general faster than the previously proposed methods. In [24], a similar algorithm is developed for non-linear stationary ordinary differential equations applied to wave propagation alongside a rigorously error analysis. A comparison of some of the methods to exponentiate an SVF has been performed in [6], while SVF-based diffeomorphic image registration has found successful applications [23, 15, 17, 21, 4].

Contributions: This paper is concerned with the numerical computation of the integrator of SVF, also called *Lie exponential*. Its main contribution is the introduction of a methodological approach based on a combination of the generalised scaling and squaring proposed in [2] and on the exponential integrators, developed in the numerical analysis community, and presented in [12].

Paper organization: In the next section we introduce the relevant theory leading to three numerical integrator algorithms, while formalizing the problem and introducing some of the available schemes in the state of the art. In section 3, we present the five datasets used for numerical experiments, and in section 4 we provide the results of the new numerical scheme compared with other algorithms.

2. Numerical computations of the exponential of a Stationary Velocity Field

Stationary and time-varying velocity field: We indicate by $\Omega \subset \mathbf{R}^d$ the image's domain ($d = 2, 3$ for bi- and tri-dimensional images), by $\phi : \Omega \to \Omega$ a diffeomorphic spatial transformation and by $\mathbf{u} : \Omega \to \mathbf{R}^d$ a stationary velocity field (or tangent vector field).

A diffeomorphism can be obtained from a velocity field $\mathbf{u}$ through a first order ordinary differential equation system (ODE) of the same dimension of Ω:

$$\frac{\mathrm{d}\phi_t}{\mathrm{d}t} = \mathbf{u}(\phi_t), \qquad \phi_0 = \mathrm{Id}\,. \tag{1}$$

whose solution at $t = 1$ is indicated with $\phi = \phi_1$. The set of diffeomorphisms continuously parametrized by a time-parameter t, indicated by $\{\phi_t\}_{t\in\mathbf{R}}$ satisfies the one-parameter subgroup properties: ϕ_0 corresponds to the identity and $\phi_t \circ \phi_s = \phi_{t+s}$ ([14], p. 102). When the velocity vector field is time-dependent, the vector field $\mathbf{u} = \mathbf{u}(t, \phi_t)$, it is called time-varying velocity field (TVVF), and the previous ODE becomes $\frac{\mathrm{d}\phi_t}{\mathrm{d}t} = \mathbf{u}(t, \phi_t)$.

We introduce two examples that will be of particular interest for our purpose: the affine ODE system, in particular when defined over elements of the Lie algebra of rigid body transformations of the d-dimensional real space (special Euclidean group $\mathrm{SE}(d)$), and the non-linear ODE system defined by an element of the projective general linear algebra, whose Lie exponentials are the homographies of $\mathbf{R}^d$. In both cases an analytic solution of the Lie exponential is available and therefore there is a ground truth to validate the numerical methods.

Linear and Affine ODE: When the ODE is considered from a given point $\mathbf{x}$ in Ω, the transformed point parametrized at the time t, given by $\phi_t(\mathbf{x})$ is denoted with $\mathbf{x}(t)$ and its time derivative with $\dot{\mathbf{x}}(t)$. Using this notation, when $\mathbf{u}$ is linear, equation (1) becomes $\dot{\mathbf{x}}(t) = L\mathbf{x}(t)$, for a $d \times d$ square matrix L and $\mathbf{x}(0) = \mathbf{x}_0$ initial condition. The solution of this linear case uses the matrix exponential: at time t it is provided by $\mathbf{x}(t) = \varphi_0(tL)\mathbf{x}_0$ where $\varphi_0(tL) = \sum_{j=0}^{\infty}(tL)^j/j!$ is the matrix exponential function. When the ODE is affine, $\dot{\mathbf{x}}(t) = L\mathbf{x}(t)+\mathbf{r}$, and the initial condition is $\mathbf{x}(0) = \mathbf{0}$, then its analytic solution is given by $\mathbf{x}(t) = t\varphi_1(tL)\mathbf{r}$, where $\varphi_1(tL)$ is the shifted Taylor expansion of the exponential given by $\sum_{j=0}^{\infty}(tL)^j/(j+1)!$ ([11] p. 10). In case the initial condition is not centred in the origin, is it possible to translate the coordinate frame by $\mathbf{y}(t) := \mathbf{x}(t) - \mathbf{x}(0)$, so that $\mathbf{y}(0) = \mathbf{0}$ and $\mathbf{x}(t) = \mathbf{y}(t) + \mathbf{x}(0)$. The affine ODE can be written as

$$\begin{bmatrix} \dot{\mathbf{y}}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} L & \mathbf{r} + L\mathbf{x}(0) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}(t) \\ 1 \end{bmatrix}, \tag{2}$$

with initial condition $[\mathbf{y}(0)\ \ 1]^T = [\mathbf{0}\ \ 1]^T$. Its analytic solution is given by

$$\begin{bmatrix} \mathbf{y}(t) \\ 1 \end{bmatrix} = \varphi_0\left(\begin{bmatrix} L & \mathbf{r} + L\mathbf{x}(0) \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}. \tag{3}$$

while using φ_1, and in Euclidean coordinates, is

$$\mathbf{x}(t) = \mathbf{x}(0) + t\varphi_1(tL)(\mathbf{r} + L\mathbf{x}(0))\,. \tag{4}$$

The numerical computation of φ_0 and φ_1 is an open field of research, and a comparison of the commonly used methods can be found in [18]. There are several numerical methods available to approximate $\varphi_0(A)\mathbf{X}_0$ or $\varphi_1(A)\mathbf{r}$, not involving any direct computation of φ_0. However, their efficiency over a simple truncation of φ_0 appears only when large matrices are involved ([8, 1, 19]).

Homography group: The affine case can be extended in homogeneous coordinates to the homography group. This second example will provide a class of polynomial ODE, where an analytic solution is still available as the quotient of the exponential of the linear matrix in projective coordinates. Indicating with $\mathbf{X}(t)$ the $(d+1)$-dimensional vector $\mathbf{x}(t)$ in projective coordinates, where all the components depends on time, and with H an element of the projective general linear algebra $\mathfrak{pgl}(d)$ ($(d+1) \times (d+1)$ matrices defined up to a constant), a linear ODE in homogeneous coordinates can be written as $\dot{\mathbf{X}}(t) = H\mathbf{X}(t)$. The derivative of the Euclidean coordinates $x_i(t) = X_i(t)/X_{d+1}(t)$, is given by the quotient rule

$$\dot{x}_i(t) = \frac{\dot{X}_i(t)X_{d+1}(t) - X_i(t)\dot{X}_{d+1}(t)}{X_{d+1}(t)^2} \qquad i = 1, \ldots, d\,. \tag{5}$$

For $X_{d+1}(t) = 1$ and $X_i(t) = x_i(t)$, the ODE can be written, as the non-linear system

$$\dot{x}_i(t) = H_{(i,:)} \cdot [\mathbf{x}(t)\ \ 1]^T - x_i(t)\big(H_{(d+1,:)} \cdot [\mathbf{x}(t)\ \ 1]^T\big) \tag{6}$$

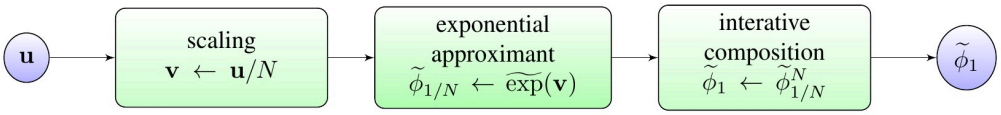


Figure 1. Scaling and composing scheme, as a generalization of the scaling and squaring. Given a stationary velocity field $\mathbf{u}$, this is first scaled by a factor of N, then an appropriate numerical integrator is applied to the scaled $\mathbf{u}$, and its result, indicated with $\widetilde{\exp}(\mathbf{u}/N)$, is iteratively composed by itself N-times. When $\widetilde{\exp}(\mathbf{u}/N)$ is computed as $\mathrm{Id} + \mathbf{u}/N$ and the iterative composition is computed by composition of the previous step with itself, then the algorithm is the classical scaling and squaring proposed in [3]. When the iterative composition is performed sequentially backward, it is equivalent to the Euler method ([7] chapter 5).

where $H_{(i,;)}$ is the i-th row of H. The analytic solution of the previous non-linear ODE in Euclidean coordinates can be recovered from the solution in homogeneous coordinates $S(t) = \varphi_0(tH)\mathbf{X}(0)$, for the initial condition $\mathbf{X}(0) = [\mathbf{x}(0)\ \ 1]^T$, as $S_{1:d}(t)/S_{d+1}(t)$.

Numerical integrators: Any numerical method whose aim is to approximate the solution of equation (1) is called *numerical integrator*. For the general case, the flow of diffeomorphism ϕ_t that solves equation (1), called *Lie exponential* of the velocity field $\mathbf{u}$ and is indicated with $\exp(t\mathbf{u})$ ([14], ch. 5), does not have an analytical solution. To address this limitation several integrators have been proposed in the literature to compute a numerical approximation of the Lie exponential (for example in [7]). Their improvement in accuracy, stability and computational time is an open field of research: authors of [6] presented a comparison of different numerical approaches for the integration of SVFs.

Particularly relevant to our purposes are the scaling and squaring and the Euler method ([3] and [24] where proofs and bounds on the accuracy of these methods are derived).

Generalized scaling and squaring: The most commonly used numerical methods to integrate a vector field are based on the discretization of the time parameter and on the Taylor expansion of the unknown solution (as for example Euler, Euler modified, Midpoint and Runge Kutta 4). When dealing with SVFs, the one-parameter subgroup property of the vector field $\phi_t = \exp(t\mathbf{u})$ and its direct implication $\exp(\mathbf{u}/N)^N = \exp(\mathbf{u})$ ([14], p. 204) is at the core of the generalized scaling and squaring numerical methods. Indicating with $\widetilde{\exp}$ the Lie exponential approximant (any appropriate numerical integrator of the Lie exponential) that deals with small SVFs, and denoting $\mathbf{u}/N$ with $\mathbf{v}$, it follows

$$\phi_1 = \exp(\mathbf{u}) = \big(\exp(\mathbf{u}/N)\big)^N \simeq \big(\widetilde{\exp}(\mathbf{v})\big)^N . \qquad (7)$$

After the scaling of $\mathbf{u}$ and the computation of $\widetilde{\exp}$, the result is iteratively composed in order to obtain the solution in the original scale. The final iterative composition can be performed in 3 ways, called forward composition, backward composition or, when $N = 2^M$, squared composition (in [6] respectively forward Euler, composition method, and scaling and squaring). At the k-th step, the computation can be performed as $\widetilde{\exp}(\mathbf{v})^{k+1} = \widetilde{\exp}(\mathbf{v}) \circ \widetilde{\exp}(\mathbf{v})^k$, $\widetilde{\exp}(\mathbf{v})^{k+1} = \widetilde{\exp}(\mathbf{v})^k \circ \widetilde{\exp}(\mathbf{v})$ or $\widetilde{\exp}(\mathbf{v})^{2^{k+1}} = \widetilde{\exp}(\mathbf{v})^{2^k} \circ \widetilde{\exp}(\mathbf{v})^{2^k}$. Even if mathematically equivalent, the squared composition is computationally faster. Authors of [2] proposed to compute $\widetilde{\exp}(\mathbf{v})$ with a generic adequate exponential integrator, as for example the Runge-Kutta 4 method (figure 1).

Exponential integrators: Another class of numerical integrators was introduced in 1958 to solve some particular stiff ODEs for which the numerical integrators based on the Taylor expansion have not provided reasonable results [12]. Called *exponential integrators*, they originates from the strategy of separating the linear part and the non-linear part of the tangent vector field. In this paper we exploit this approach, combined with the generalized scaling and squaring to produce a new numerical integrator method.

Combining generalized scaling and squaring and exponential integrators: The scaling step, as the first step of the scaling and squaring algorithm, appears to be particularly convenient when dealing with large deformations between images. Once the SVF is reduced in scale, it is possible to integrate it, separating the linear from the non-linear part, following the exponential integrator methodology. After scaling $\mathbf{u}$ to $\mathbf{v} = \mathbf{u}/N$ and changing the coordinate frame to have the initial condition centred in the origin, equation (1) becomes $\dot{\mathbf{y}}(t) = \mathbf{v}(\mathbf{y}(t) + \mathbf{x}(0))$. Separating the linear from the non-linear part with the Taylor series expansion of $\mathbf{v}$ around $\mathbf{x}(0)$ in $\mathbf{x}(t)$, the ODE becomes:

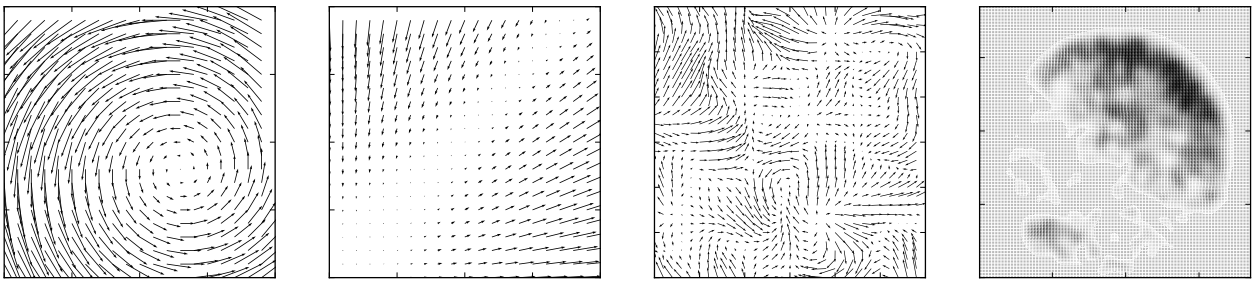
$$\dot{\mathbf{y}}(t) = \mathbf{v}(\mathbf{y}(t) + \mathbf{x}(0)) = \mathbf{v}_0 + \mathrm{J}_{\mathbf{v}_0}\mathbf{y}(t) + \mathcal{N}_{\mathbf{v}}(\mathbf{y}(t)), \quad (8)$$

where $\mathbf{v}(\mathbf{x}(0))$ is written as $\mathbf{v}_0$ for brevity and $\mathrm{J}_{\mathbf{v}_0}$ is the Jacobian matrix. The term $\mathcal{N}_{\mathbf{v}}$ is a non-linear operator obtained subtracting the linear part of the the SVF computed with the Taylor expansion of $\mathbf{v}$ in $\mathbf{x}(t)$ around $\mathbf{x}(0)$:

$$\mathcal{N}_{\mathbf{v}}(\mathbf{y}(t)) = \mathbf{v}(\mathbf{y}(t) + \mathbf{x}(0)) - \big(\mathbf{v}_0 + \mathrm{J}_{\mathbf{v}_0}\mathbf{y}(t)\big) . \qquad (9)$$

It follows that $\mathcal{N}_{\mathbf{v}}(\mathbf{y}(0)) = \mathbf{0}$ and $\mathcal{N}_{\mathbf{v}}(\mathbf{y}(t)) \in \mathcal{O}\big(\mathbf{y}(t)^2\big)$ for $\mathbf{x}(t) \to \mathbf{x}(0)$, or equivalently $t \to 0$.

Considering only the linear part, it is possible to apply φ_0, as proposed in equation (3), obtaining the solution $\mathbf{x}(t)$ as the product of the matrix $\varphi_0(t[\mathrm{J}_{\mathbf{v}_0}, \mathbf{v}_0; 0, 0])$ times



(a) (b) (c) (d)

Figure 2. Visual description of the data set used in the experiments. *(a)* Axial section of a velocity field generated by element of the rigid body transformation of the space with rotation randomly chosen in the interval $(-\frac{\pi}{8}, \frac{\pi}{8})$ and with the centre randomly selected in the field of view. An analytic solution of the ODE is available for any choice of rotation and translation. *(b)* Axial section of a velocity field generated by an homography H scaled according to the dimension of the image, such that its matrix Logarithm has real entries and the last row of H multiplied by its column is positive. For this dataset an analytic solution is available. *(c)* Axial section of an SVF with values randomly generated from a normal distribution with $\sigma_{\text{init}} = 5$ and smoothed with a Gaussian filter of $\sigma_{\text{gf}} = 2$. A wide number of samples of any dimension can be quickly generated, but there is no analytic solution available. *(d)* SFV obtained from the registration of two longitudinal weighted MR brain images from MIRIAD dataset.

$[\mathbf{0}, 1]^T$. Since this choice is computationally expensive the function φ_1, as proposed in equation (4), provides a second option that reduces the computational time. With this choice, we obtain: $\mathbf{x}(t) = \mathbf{x}(0) + t\varphi_1(t\mathrm{J}_{\mathbf{v}_0})\mathbf{v}_0$. When dealing with sufficiently small time-step, the Jacobian is close to the identity and an additional term proportional to $\mathrm{J}^2_{\mathbf{v}_0}$ did not provide any sensible improvement in the numerical results. For this reason φ_1 can be truncated at its second order:

$$\mathbf{x}(t) = \mathbf{x} + t\mathbf{v}_0 + \frac{t^2}{2}\mathrm{J}_{\mathbf{v}_0}\mathbf{v}_0 + \mathcal{O}\big(t^3\mathrm{J}^2_{\mathbf{v}_0}\mathbf{v}_0\big) \ . \tag{10}$$

In conclusion, an approximation of the solution of the linearised equation (1), after the scaling step, can be written as

$$\widetilde{\exp}(t\mathbf{v})(\mathbf{x}) := \mathbf{x} + t\mathbf{v}_0 + \frac{t^2\mathrm{J}_{\mathbf{v}_0}\mathbf{v}_0}{2} \ , \tag{11}$$

with asympthotic error $\mathcal{O}\big(\mathbf{y}(t)^2 + t^3\mathrm{J}^2_{\mathbf{v}_0}\mathbf{v}_0\big)$, $t \to 0$ that arises from the truncation of φ_1 and from the error introduced when neglecting the non-linear part.

According to the choice of the exponential approximant $\widetilde{\exp}$, to the truncation of φ_1 and to the iterative composition $(\varphi_{1/N})^N$, the method proposed can be implemented with three possible algorithms:

1. *Scaling and squaring based on exponential integrators* (ss_ei): $\widetilde{\exp}$ is computed approximating the exponential of the matrix $[\mathrm{J}_{\mathbf{v}_0}, \mathbf{v}_0; 0, 0]$ (equation (3)) and the composition is computed by squaring.

2. *Scaling and squaring based on approximated exponential integrators* (ss_aei): $\widetilde{\exp}$ is computed by truncating φ_1 (equation (11), with $t = 1$) and the composition is computed by squaring.

3. *Scaling and composing based on approximated exponential integrators* (euler_aei) $\widetilde{\exp}$ is computed by truncating φ_1 (equation (11), with $t = 1$) and the composition is computed with forward integrations.

The remaining part of this paper is devoted to the validation and the comparison of these algorithms with the state of the art.

3. Dataset Descriptions

Synthetic dataset 1: The first dataset consists of 50 linear velocity fields sampled on a regular grid of size $60 \times 60 \times 60$ generated by elements of the Lie algebra $\mathfrak{se}(3)$ of the rigid body transformations of the d dimensional space ([14], p. 191, 214). Rotations are randomly chosen in the interval $(-\frac{\pi}{8}, \frac{\pi}{8})$ and with the centre randomly selected in the field of view. Studying these linear ODEs is particularly relevant: a closed form of the analytic solution $\phi = \exp(\mathbf{v})$ is available, therefore results obtained from numerical integrators can be compared with a ground truth.

Synthetic dataset 2: The SVFs belonging to the second dataset consists of 50 SVFs sampled on a regular grid of size $60 \times 60 \times 60$ and generated by homographies in the projective linear group introduced in section 2. Each homography H is a $(d+1) \times (d+1)$ random matrix whose elements are sampled from a Gaussian distribution with standard deviation 1 and scaled by a factor of 60. To avoid foldings, degenerate cases and to avoid the exponential matrix to have complex entries, two additional constraints are added: the matrix logarithm of H must have real entries and each row of H when multiplied by any of its columns always provides a positive number. Despite the fact that the corresponding ODE is non-linear, the analytic solution of the Lie exponential is available.

Synthetic dataset 3: In the third dataset, 50 stationary velocity fields are directly generated on a grid of size $60 \times 60 \times 60$ as random values from a normal distribution with $\sigma_{\text{init}} = 5$ and smoothed with a Gaussian filter with $\sigma_{\text{gf}} = 2$. Since the Lie exponential does not have any analytic solution, a numerical benchmark is chosen among the results of the implemented numerical integration methods.

Real cases data 1: This dataset consists of 10 SVFs from longitudinal registrations performed with the last version of NiftyReg (with default parameters) [17] of T1 MRI images collected from the MIRIAD dataset [16].

Real cases data 2: To compare the classical scaling and squaring with the approximated exponential integrators, a segmentation propagation is performed 35 scans a manually segmented, as provided by Neuromorphometrics for the MICCAI 2012 Grand Challenge on Label Fusion https://masi.vuse.vanderbilt.edu/workshop2012/index.php /Challenge_Details. The pairwise registrations of the 35 selected images resulted in 1190 propagated segmentations for each considered algorithm.

Registrations are performed using NiftyReg and the accuracy is measured as the pixelwise sum of the squared relative difference between the analytic solution (when available) or the chosen numerical benchmark. The computational time is measured on a Mac book pro 2014 16Gb 1600 MHz DDR3. The python code implemented to produce the images and elaborate the data is publicly available on the authors' repository https://github.com/giftsurg/lie_exponential.

4. Experiments and Validations

This section presents the results obtained with the three proposed algorithms. Their performance are compared with other numerical integrators available in the literature.

4.1. Accuracy and computational time comparison

Figure 3 shows the outcome of an experiment performed with the *synthetic dataset 2* described in section 3. The mean errors are computed as the Euclidean norm of the difference of the displacement fields obtained from the numerical computation of the Lie exponential with different algorithms.

Figure 4 shows the same analysis for four datasets and for a choice of significant numerical methods. In *(a)*, the sampled SVFs belong to the *synthetic dataset 1*: we observe that for all the methods that are not based on the exponential integrators, increasing the number of steps induces a decrease in the error. Since they provide exact solutions for the linear part of the SVFs, the methods ss_ei, ss_aei and euler_aei are more accurate at the first or second step for this particular case of linear ODEs. Resampling error is however accumulated with each additional step. In *(c)* SVFs

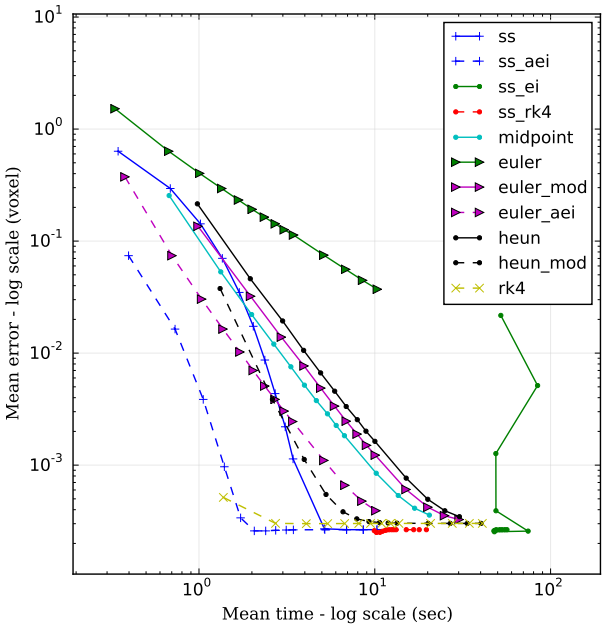


Figure 3. The three numerical methods to approximate Lie exponential proposed (scaling and squaring based on exponential integrators ss_ei, scaling and squaring based on approximated exponential integrators ss_aei and scaling and composing based on approximated exponential integrators euler_aei) are compared in error and computational time for a different number of steps, with the classical scaling and squaring (ss), the midpoint method (midpoint), the Euler method (euler), the Euler modified method (euler_mod), the Heun method (heun), the Heun modified method (heun_mod) and the explicit Runge-Kutta 4 method (rk4), described in [3, 7]. The additional method ss_rk4 combines the scaling and squaring framework and compute $\widetilde{\exp}$ with rk4. The number of steps for each method is selected in the list $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30]$ to a dataset of 50, sampled on a regular grid of size $60 \times 60 \times 60$ and generated by homographies. The error is computed as the mean of the voxelwise normalized Euclidean norm of the difference between the deformation obtained with the numerical algorithm and the available analytic solution.

belong to the *synthetic dataset 3* (non-linear random generated and smoothed with a Gaussian filter) and in *(d)* SVFs are from real patients' images, belonging to the *real cases 1* dataset. In *(b)*, the same non-linear dataset showed in figure 3 and generated by homographies, is proposed again, to emphasise the differences between the four datasets described in figure 2. The classical scaling and squaring is, in each case, more accurate than the Euler method, while it is systematically outperformed by the scaling and squaring based on approximated exponential integrators (ss_aei). When SVFs are sampled from the *synthetic dataset 3* and *real case 1* (figure 4 *(c)* and *(d)*), the *Runge Kutta 4* with fixed number of steps equals to 7, is used as numerical benchmark, having

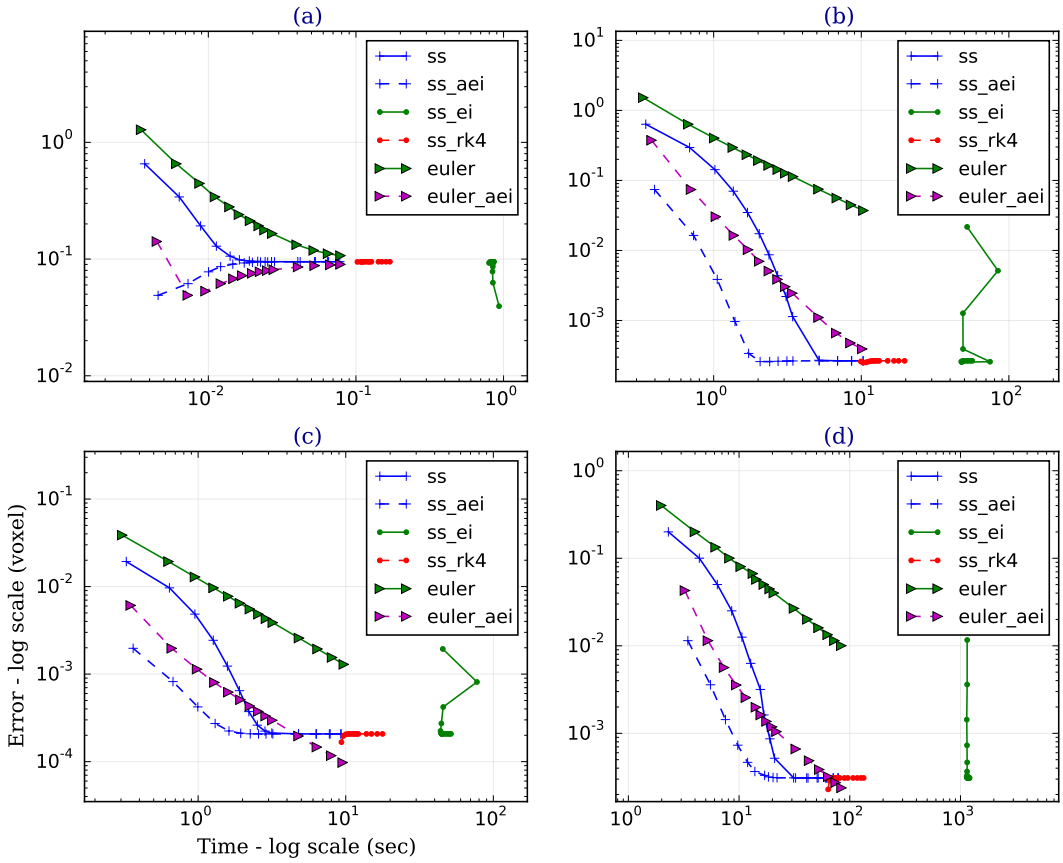


Figure 4. Computational time and errors are compared as in figure 3 for each of the datasets presented in figure 2. The methods selected are the classical scaling and squaring (ss), the generalized scaling and squaring with approximated exponential integrators (ss_aei), with exponential integrators (ss_ei), the scaling and squaring based on Runghe-Kutta 4 (ss_rk4), the Euler method (euler) and the Euler method based approximated exponential integrators (euler_aei). The number of steps are selected in the list $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30]$ for figure *(a)*, *(b)*, *(c)*, and with the two additional steps $[35, 40]$ for figure *(d)*. Three dataset of 50 SVFs generated with elements in the Euclidean group SE(3) *(a)*, with homographies *(b)* and using Gaussian filters *(c)*, are sampled on a regular grid of size $60 \times 60 \times 60$. In *(d)* the selected dataset consists of 10 SFV obtained registering two longitudinal weighted MR brain images. When the ground truth is not available *(c, d)* , the selected numerical benchmark is the Runge Kutta 4 method computed with 7 steps, since it provided the lowest error in the exponentiating SVFs generated by homographies (see figure 3) the computational time of the benchmark is in the order of 10^1 seconds for the synthetic generated SVFs *(c)*, and in the order of 10^2 seconds for the real patients images *(d)*.

had the best performance non-linear SVFs showed in figure 3, when a ground truth was available. The optimal accuracy performance of the generalized scaling and squaring with Runge-Kutta 4 is a consequence of the similarity between this method and the chosen numerical benchmark; the increase in accuracy of the scaling and composing based on approximated exponential integrators (euler_aei) is biased as well by this choice. The scaling and squaring based on exponential integrators (ss_ei), grounded on equation (3), has very similar accuracy of the scaling and squaring based on approximated exponential integrators (ss_aei), with a much higher computational cost.

4.2. Segmentation propagation accuracy test

We compared the performance of the open-source NiftyReg registration framework with a second version implemented to compute the Lie exponential with scaling and squaring based on approximated exponential integrator in a segmentation propagation test. The obtained propagated segmentations originating from a dataset of 35 scans from the Neuromorphometric project presented in section 3 as *real case dataset 2*, were compared to the ground truth using the Dice similarity coefficient (DSC) as a measure of overlap for each segmented label. The results were not statistically different when using the original or the modified version of NiftyReg (p-value 0.99, DSC 0.590.03). This is likely to be a consequence of the fact that in NyftiReg, a

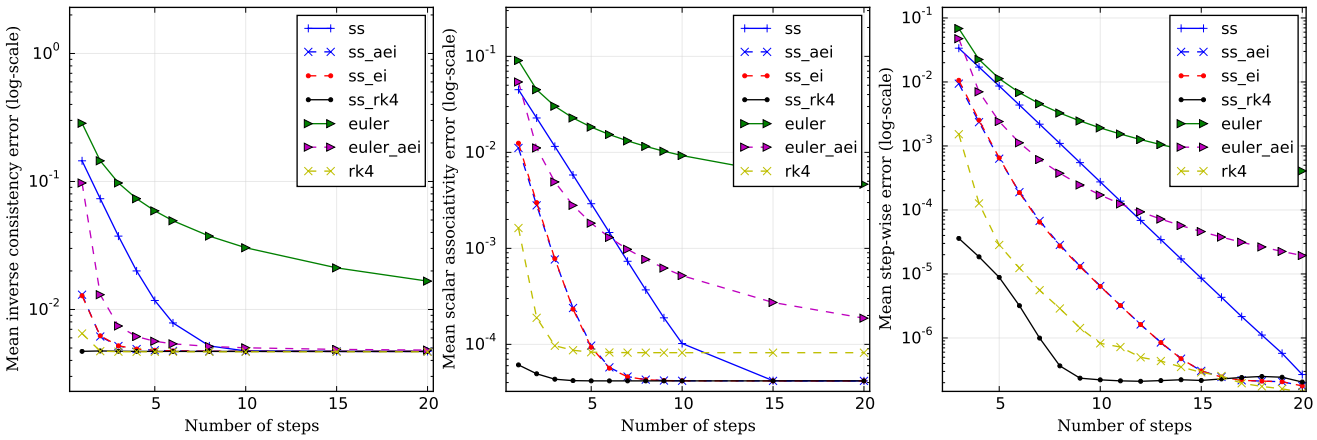


Figure 5. Inverse consistency, scalar associativity and step-wise error for the real cases data 1, consisting of 10 SVFs obtained from the longitudinal registration of T1 MRI patients scans. For computational time reasons, SVFs where resticted to a subset of the whole image, sequences of tests showed this choice does not affect the trend of the these results. In each graph, the methods ss_ei, ss_aei are almost coincident.

conservative number of step is used, leading the both algorithms converging to a very similar field.

4.3. Inverse consistency, scalar associativity and step-wise error

Three validation methods that do not require the availability of a ground truth are the inverse consistency, the scalar associativity and the step-wise error.

The inverse consistency error of a given SVF $\mathbf{u}$ for the numerical approximation of the Lie exponential $\widetilde{\exp}$ is computed as: $\mathrm{E} = \frac{1}{2}\left(\|\widetilde{\exp}(\mathbf{u}) \circ \widetilde{\exp}(-\mathbf{u}) - \mathrm{Id}\| + \|\widetilde{\exp}(-\mathbf{u}) \circ \widetilde{\exp}(\mathbf{u}) - \mathrm{Id}\|\right)$ where the norm considered is the Eulcidean norm performed in the space of vector fields. The scalar associativity exploits the one parameter subgroup property to evaluate the performance of the computation of $\widetilde{\exp}$. Given a, b, c such that $a+b+c=1$, the composition of Lie exponentials $\exp(a\mathbf{u}) \circ \exp(b\mathbf{u}) \circ \exp(c\mathbf{u})$ is theoretically coincident with $\exp(\mathbf{u})$ and the normed differences in the computations measures the lack of accuracy of the numerical integrator and the resampling error. For the shown experiments we selected $a = 0.3$, $b = 0.3$, $c = 0.4$. Small variations in this choice do not impact significantly the results. The step-wise error is computed here as the Euclidean norm of the difference between the exponential performed with $i+1$ steps and the exponential performed with i steps, with $i = 3, ..., 20$. Figure 5 shows the results obtained for the inverse consistency, scalar associativity and step-wise error of seven relevant methods for the *real dataset 1* presented in section 3. Due to the high computational cost of these computation results are obtained from 2d randomly selected axial slices of the SVFs, in central regions. The same experiment with 3d dataset randomly generated provided the same relative difference for each method. We observe that ss_ei and ss_aei have similar accuracy performance in both the inverse consistency, the scalar associativity and the step-wise error. Method euler_aei has similar trend to the Euler method but converges faster to the asymptotic error. For the scalar associativity the distinction between the methods based on the forward composition and the squared composition is translated in two different asympthotic behaviour.

5. Conclusions and Perspectives

We have proposed to combine the exponential integrator methods with the generalized scaling and squaring framework to produce a novel approach to approximate the Lie exponential. Three algorithms, called *scaling and squaring based on exponential integrators*, *scaling and squaring based on approximated exponential integrators* and *scaling and composing based on approximated exponential integrators*, have been developed and compared with respect to their accuracy and computational cost, both on synthetic and clinical data.

From this preliminary investigation, the generalized scaling and squaring based on approximated exponential integrator appears to be a valid alternative to the classic scaling and squaring commonly used in registration frameworks. While at the core of the classic version there is the approximation $\widetilde{\exp}(\mathbf{v}) \leftarrow \mathrm{Id} + \mathbf{v}$ computed before a squaring composition, the scaling and squaring based on approximated exponential integrators, here proposed, is based on approximating $\widetilde{\exp}(\mathbf{v})$ with $\mathrm{Id} + \mathbf{v} + \frac{1}{2}\mathrm{J}_{\mathbf{v}}\mathbf{v}$. We observe that this approximation, in case the SVF is not scaled before its applications, coincides with a second order truncation of the Lie series integrators [9, 10] where $\mathbf{v}^2$ is computed as the directional derivative of $\mathbf{v}$ over its components. This method has been reported in [6] to yield poor results in the case of large deformations, but it is used in this paper with small

velocity fields thanks to the scaling step. In addition, the original proof of its convergence was proposed for complex holomorphic functions, while an extension to the case of SVFs is not known to the authors.

Experiments in this work are restricted to stationary velocity field, but the Euler method based on exponential integrators can be easily extended to deal with non-stationary velocity fields. Moreover, the approach here proposed did not fully exploited the vast possibilities provided by the exponential integrators. Other algorithms, such as the ET-DRK4 [12], certainly deserve to be taken into account for further investigations in diffeomorphic image registration.

Acknowledgments

Sebastiano Ferraris is supported by the EPSRC-funded UCL Centre for Doctoral Training in Medical Imaging (EP/L016478/1) and Doctoral Training Grant (EP/M506448/1). Pankaj Daga was funded through an Innovative Engineering for Health award by Wellcome Trust [WT101957]; Engineering and Physical Sciences Research Council (EPSRC) [NS/A000027/1]. Marc Modat is supported by the UCL Leonard Wolfson Experimental Neurology Centre. Tom Vecauteren is supported by an Innovative Engineering for Health award by the Wellcome Trust [WT101957]; Engineering and Physical Sciences Research Council (EPSRC) [NS/A000027/1] The MIRIAD dataset is made available through the support of the UK Alzheimer's Society (Grant RF116). The original data collection was funded through an unrestricted educational grant from GlaxoSmithKline (Grant 6GKC).

References

[1] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM journal on scientific computing*, 33(2):488–511, 2011.

[2] V. Arsigny, O. Commowick, N. Ayache, and X. Pennec. A fast and log-euclidean polyaffine framework for locally linear registration. *Journal of Mathematical Imaging and Vision*, 33(2):222–238, 2009.

[3] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A log-euclidean framework for statistics on diffeomorphisms. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*, pages 924–931. Springer, 2006.

[4] J. Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

[5] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.

[6] M. Bossa, E. Zacur, and S. Olmos. Algorithms for computing the group exponential of diffeomorphisms: Performance evaluation. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[7] J. F. Burden, Richard and A. Burden. Numerical analysis. 9th edition, Nelson Education, 2015.

[8] M. Caliari, P. Kandolf, A. Ostermann, and S. Rainer. Comparison of software for computing the action of the matrix exponential. *BIT Numerical Mathematics*, 54(1):113–128, 2014.

[9] W. Grobner and H. Knapp. Contributions to the method of lie series. 1967.

[10] A. Hanslmeier and R. Dvorak. Numerical integration with lie series. *Astronomy and Astrophysics*, 132:203, 1984.

[11] N. J. Higham and L. Lijing. Matrix functions: A short course, 2013.

[12] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer*, 19:209–286, 2010.

[13] M. Holden. A review of geometric transformations for nonrigid body registration. *Medical Imaging, IEEE Transactions on*, 27(1):111–128, 2008.

[14] D. D. Holm, T. Schmah, and C. Stoica. Geometric mechanics and symmetry: From finite to infinite dimensions (oxford texts in applied and engineering mathematics). 2009.

[15] M. Lorenzi, N. Ayache, G. B. Frisoni, X. Pennec, and ADNI. Lcc-demons: a robust and accurate symmetric diffeomorphic registration algorithm. *NeuroImage*, 81:470–483, 2013.

[16] I. B. Malone, D. Cash, G. R. Ridgway, D. G. MacManus, S. Ourselin, N. C. Fox, and J. M. Schott. Miriadpublic release of a multiple time point alzheimer's mr imaging dataset. *NeuroImage*, 70:33–36, 2013.

[17] M. Modat, P. Daga, M. J. Cardoso, S. Ourselin, G. R. Ridgway, and J. Ashburner. Parametric non-rigid registration using a stationary velocity field. In *Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012 IEEE Workshop on*, pages 145–150. IEEE, 2012.

[18] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[19] J. Niesen and W. M. Wright. Algorithm 919: A krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Transactions on Mathematical Software (TOMS)*, 38(3):22, 2012.

[20] M. Niethammer, Y. Huang, and F.-X. Vialard. Geodesic regression for image time-series. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 655–662. Springer, 2011.

[21] A. Pai, S. Sommer, L. Sørensen, S. Darkner, J. Sporring, and M. Nielsen. Image registration using stationary velocity fields parameterized by norm-minimizing wendland kernel. In *SPIE Medical Imaging*, pages 941335–941335. International Society for Optics and Photonics, 2015.

[22] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable medical image registration: A survey. *Medical Imaging, IEEE Transactions on*, 32(7):1153–1190, 2013.

[23] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Symmetric log-domain diffeomorphic registration: A demons-based approach. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pages 754–761. Springer, 2008.

[24] L. Ying and E. J. Candès. The phase flow method. *Journal of Computational Physics*, 220(1):184–215, 2006.