IOP Publishing Inverse Problems

Inverse Problems **36** (2020) 034004 (33pp) https://doi.org/10.1088/1361-6420/ab2772

Geometric numerical integration of the assignment flow

Alexander Zeilmann[1,5], Fabrizio Savarino[1], Stefania Petra[2,3] and Christoph Schnörr[1,4]

[1] Image and Pattern Analysis Group, Heidelberg University, Germany
[2] Mathematical Imaging Group, Heidelberg University, Germany

E-mail: alexander.zeilmann@iwr.uni-heidelberg.de, fabrizio.savarino@iwr.uni-heidelberg.de, petra@math.uni-heidelberg.de and schnoerr@math.uni-heidelberg.de

Received 5 October 2018, revised 16 April 2019
Accepted for publication 22 May 2019
Published 30 January 2020

Abstract

The *assignment flow* is a smooth dynamical system that evolves on an elementary statistical manifold and performs contextual data labeling on a graph. We derive and introduce the *linear assignment flow* that evolves nonlinearly on the manifold, but is governed by a linear ODE on the tangent space. Various numerical schemes adapted to the mathematical structure of these two models are designed and studied, for the geometric numerical integration of both flows: embedded Runge–Kutta–Munthe–Kaas schemes for the nonlinear flow, adaptive Runge–Kutta schemes and exponential integrators for the linear flow. All algorithms are parameter free, except for setting a tolerance value that specifies adaptive step size selection by monitoring the local integration error, or fixing the dimension of the Krylov subspace approximation. These algorithms provide a basis for applying the assignment flow to machine learning scenarios beyond supervised labeling, including unsupervised labeling and learning from controlled assignment flows.

Keywords: image labeling, assignment flow, assignment manifold, geometric integration, adaptive step size selection, Krylov subspace approximation

(Some figures may appear in colour only in the online journal)

[3] https://www.stpetra.com
[4] https://ipa.math.uni-heidelberg.de
[5] Author to whom any correspondence should be addressed.

1. Introduction

1.1. Overview, motivation

The *assignment flow*, recently introduced by [1] and detailed in section 2, denotes a *smooth dynamical system* evolving on an elementary statistical manifold, for the contextual classification of a finite set of data given on an arbitrary graph. Vertices of the graph are associated with the elements of the data set and correspond to locations in space and/or time in typical applications. *Classification* means to assign each datum exactly to one class representative, called *label*, out of a finite set of predetermined labels. *Contextual* classification means that these decision directly depend on each other, as encoded by the edges (adjacency relation) of the underlying graph. In the context of image analysis, classifying given image data on an image grid graph in this way is called the *image labeling problem*. We point out, however, that the assignment flow applies to arbitrary data represented on a graph.

A key property of the assignment flow is that decision variables do *not* live in the space used to model the data. Rather, a probability simplex is associated with each datum, on which a flow evolves until it converges to one of the vertices of the simplex that encode the labels. Each simplex is equipped with the Fisher–Rao metric which turns the relative interior of the simplex into a smooth Riemannian manifold. It is this particular geometry that effectively promotes discrete decisions that interact in a *smooth* way. Replacing in addition the Riemannian (Levi-Civita) connection by the α-connection (with $\alpha = 1$) introduced by Amari and Chentsov [2], enables to carry out basic geometric operations in a computationally *efficient* way. Keeping the assignment flow as 'inference engine' separate from the data space and model allows to flexibly apply it to a broad range of contextual data classification problems. We refer to [2, 3] as basic texts on information geometry and to [4] for more information on the image labeling problem.

From a more distant viewpoint, our work ties in with the recent trend to explore the mathematics of deep networks from a dynamical systems perspective [5]. A frequently cited paper in this respect is [6] where a connection was made between the so-called residual architecture of networks and explicit Euler integration steps of a corresponding system of nonlinear *ordinary differential equations (ODEs)*. We refer to [7] for a good exposition. While this offers a novel and fresh perspective on the *learning problem* of network parameters, it does not alter the basic ingredients of such networks that apparently have been adopted in an ad-hoc way, like parametrized static layers connected by nonlinear transition functions, ReLU activations etc.

By contrast, the assignment flow provides a smooth dynamical system on a graph (network), where all ingredients coherently fit into the overall mathematical framework. Based on this, we recently showed how discrete graphical models for image labeling can be evaluated using the assignment flow [8], and how *unsupervised* labeling can be modeled by coupling the assignment flow and Riemannian gradient flows for label evolution on feature manifolds [9]. Our current work, to be reported elsewhere, studies machine learning problems based on *controlling* the assignment flow. Here, in particular, *algorithms* play a decisive role that accurately *integrate the assignment flow numerically on the manifold* where it evolves. A thorough study of such algorithms is the subject of the present paper.

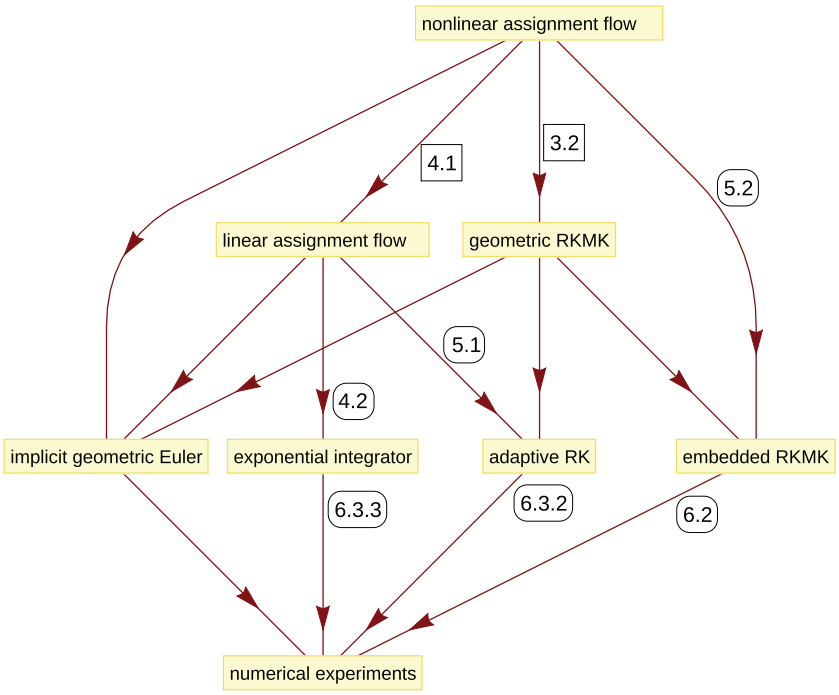


Figure 1. Topics addressed in this paper and their interrelations. Edge labels refer to the corresponding sections. Section numbers framed by squares address modeling aspects, whereas those framed by rounded squares address the design of algorithms and their numerical evaluation. Unlabeled edges mean 'is derived from' or 'provides the basis for'.

1.2. Contribution, organization

This paper presents two interrelated contributions, as illustrated by figure 1.

(1) We derive from the assignment flow—henceforth called *nonlinear assignment flow*—the *linear assignment flow*, that still is nonlinear but governed by a *linear* ODE on the tangent space. This property is attractive for modeling tasks (e.g. parameter estimation and control) as well as for the design of numerical algorithms. In particular, our experiments show that the linear flow closely approximates the nonlinear flow, as far as concerns the final *labeling* results.
(2) We study a range of algorithms for numerically integrating both the nonlinear and the linear assignment flow, respectively, while properly taking into account the underlying *geometry*.

   (a) Regarding the *nonlinear* assignment flow, we adopt the machinery of Lie group methods for the numerical integration of ODEs on manifolds [10] and devise corresponding extensions of classical Runge–Kutta (RK) schemes, called *RKMK schemes* (Runge–Kutta–Munthe–Kaas) in the literature [11]. We combine pairs of these extensions to form *embedded* RKMK schemes for adaptive step size control, analogous to classical embedded RK schemes [12].

(b) Regarding the *linear* assignment flow, we take advantage in two alternative ways of the *linearity* of the flow on the tangent space.

(i) On the one hand, we derive a local error estimate in order to apply classical RK schemes [12] on the tangent space, with step sizes that adapt automatically.
(ii) On the other hand, we evaluate the integral representation of the linear flow, due to Duhamels formula, and approximately evaluate this integral using Krylov subspace methods, as has been developed in the literature on exponential integrators [13–15].

All these *explicit* numerical schemes are evaluated and discussed in section 6, using 'ground truth' flows as a baseline that were computed using the *implicit* geometric Euler scheme with a sufficiently small step size. All iterative algorithms are parameter free, except for specifying a single tolerance value with respect to the local error, that governs adaptive step size selection. Our experiments indicate a value for this parameter that 'works' regarding integration accuracy and labeling quality, but is not too conservative (i.e. small). In the case of the exponential integrator, we merely have to supply the final point of time T at which the linear assignment flow should be evaluated, in addition to the dimension of the Krylov subspace which controls the quality of the approximation. We conclude with a synopsis of our results in section 7.

1.3. Basic notation

Index sets I and J index vertices $i \in I$ of the underlying graph and labels $j \in J$, respectively. $\mathcal{S}$ and $\mathcal{W}$ denote the basic statistical manifolds that we work with, defined in section 2. Points $p, q \in \mathcal{S}$ are strictly positive probability vectors, and we denote efficiently by

$$qp = (q_1 \cdot p_1, \ldots, q_{|J|} \cdot p_{|J|})^\top, \qquad \frac{q}{p} = \left(\frac{q_1}{p_1}, \ldots, \frac{q_{|J|}}{p_{|J|}}\right)^\top \tag{1.1}$$

componentwise multiplication for general vectors, and componentwise subdivision only if $p \in \mathcal{S}$. Likewise, functions like the exponential function and the logarithm with vectors as arguments apply componentwise,

$$e^v = (e^{v_1}, e^{v_2}, \ldots)^\top, \qquad \log v = (\log v_1, \log v_2, \ldots)^\top. \tag{1.2}$$

Exp and $\exp$ denote exponential mappings defined in section 2, whereas expm denotes the *matrix* exponential in section 4.2. The ordinary exponential function defined on the real line $\mathbb{R}$ is always denoted by e^x, $x \in \mathbb{R}$.

$\mathbb{1} = (1, \ldots, 1)^\top$ denotes the constant 1-vector with appropriate number of components depending on the context. We use the common shorthand $[n] = \{1, 2, \ldots, n\}$ with $n \in \mathbb{N}$. $\|\cdot\|$ denotes the ℓ_2-norm and $\|\cdot\|_p$ the ℓ_p-norm if $p \neq 2$.

2. The assignment flow

We summarize the assignment flow introduced by [1] and related concepts required in this paper. Let $G = (I, E)$ be a given graph and let

$$\mathcal{F}_I = \{f_i : i \in I\} \subset \mathcal{F} \tag{2.1a}$$

be data given in a metric space

$$(\mathcal{F}, d). \tag{2.1b}$$

We call $\mathcal{F}_I$ *image data* even though the f_i typically represent *features* extracted from raw image at pixel $i \in I$ as a preprocessing step. G may be a grid graph (with self-loops) as in low-level image processing or a less structured graph, with arbitrary connectivity in terms of the neighborhoods

$$\mathcal{N}_i = \{k \in I \colon ik = ki \in E\} \cup \{i\}. \tag{2.2}$$

We associate with each neighborhood $\mathcal{N}_i$ weights satisfying

$$w_{ik} > 0, \qquad \sum_{k \in \mathcal{N}_i} w_{ik} = 1, \qquad \forall i \in I. \tag{2.3}$$

These weights parametrize the regularization property of the assignment flow and are assumed to be given. How to learn them from data in order to control the assignment flow will be reported elsewhere.

Along with $\mathcal{F}_I$ we assume prototypical data

$$\mathcal{G}_J = \big\{g_j \in \mathcal{F} \colon j \in J\big\} \tag{2.4}$$

to be given, henceforth called *labels*. Each label g_j represents the data of class j. *Image labeling* denotes the problem to assign class labels to image data depending on the local context encoded by the graph G. We refer to [8] for more details and background on the image labeling problem.

Assignments of labels to data are represented by discrete probability distributions

$$W_i = (W_{i1}, \ldots, W_{i|J|})^\top \in \mathcal{S}, \quad i \in I, \tag{2.5}$$

where

$$\mathcal{S} = \big\{ p \in \mathbb{R}^{|J|} \colon p_j > 0,\ \forall j \in J,\ \langle \mathbb{1}, p \rangle = 1 \big\} \tag{2.6}$$

denotes the relatively open probability simplex equipped with the Fisher–Rao metric

$$g_p(u,v) = \sum_{j \in J} \frac{u_j v_j}{p_j}, \qquad u, v \in T_0 = \{ p \in \mathbb{R}^{|J|} \colon \langle \mathbb{1}, p \rangle = 0 \}, \qquad p \in \mathcal{S}, \tag{2.7}$$

which turns $\mathcal{S}$ into a Riemannian manifold. In connection with $\mathcal{S}$, we define the

$$\mathbb{1}_{\mathcal{S}} = \frac{1}{|J|}(1, \ldots, 1)^\top \in \mathbb{R}^{|J|} \qquad (\textbf{barycenter}) \tag{2.8}$$

of $\mathcal{S}$, i.e. the uniform distribution, the orthogonal projection

$$\Pi_{T_0} \colon \mathbb{R}^{|J|} \to T_0, \qquad \Pi_{T_0}(z) = \big(\mathrm{Diag}(\mathbb{1}) - \mathbb{1}\mathbb{1}_{\mathcal{S}}^\top\big) z, \tag{2.9a}$$

and the linear replicator map

$$R_p \colon \mathbb{R}^{|J|} \to T_0, \qquad R_p(z) = \big(\mathrm{Diag}(p) - pp^\top\big) z, \qquad p \in \mathcal{S} \tag{2.9b}$$

satisfying

$$R_p = R_p \Pi_{T_0} = \Pi_{T_0} R_p. \tag{2.10}$$

Adopting the α-connection with $\alpha = 1$ from information geometry as introduced by Amari and Chentsov [2, section 2.3], [3], the exponential map based on the corresponding affine geodesics reads

$$\mathrm{Exp}\colon \mathcal{S} \times T_0 \to \mathcal{S}, \qquad (p, v) \mapsto \mathrm{Exp}_p(v) = \frac{e^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}} \rangle} p \tag{2.11a}$$

with inverse [1, appendix]

$$\mathrm{Exp}^{-1}\colon \mathcal{S} \times \mathcal{S} \to T_0, \qquad (p, q) \mapsto \mathrm{Exp}_p^{-1}(q) = R_p \log \frac{q}{p}. \tag{2.11b}$$

Specifically, we define

$$\exp_p = \mathrm{Exp}_p \circ R_p\colon \mathbb{R}^{|J|} = T_0 \oplus \mathbb{R}\mathbb{1} \to \mathcal{S}, \qquad z \mapsto \frac{p e^z}{\langle p, e^z \rangle}, \qquad \forall p \in \mathcal{S} \tag{2.12a}$$

with inverse [1, appendix]

$$\exp_p^{-1}\colon \mathcal{S} \to T_0, \qquad q \mapsto \Pi_{T_0} \log \frac{q}{p}. \tag{2.12b}$$

Remark 2.1. Calling (2.12*b*) the 'inverse' map is justified by the fact that $\exp_p$ does not depend on any constant component $\mathbb{R}\mathbb{1} \in T_0^{\perp}$ of the argument vector z. Yet, we choose $\mathbb{R}^{|J|}$ as domain because $\exp_p$ will be applied to arbitrary distance vectors $D_i \in \mathbb{R}^{|J|}$ (see (2.17)) arising from given data, and the notation indicates that the implementation does not need to remove this component explicitly [1, remark 4].

These mappings naturally extend to the collections of assignment vectors (2.5), regarded as points on the

$$\mathcal{W} = \mathcal{S} \times \ldots \times \mathcal{S} \qquad (|I| \text{ times}) \qquad (\textbf{assignment manifold}) \tag{2.13}$$

with tangent space

$$\mathcal{T}_0 = T_0 \times \ldots \times T_0 \qquad (|I| \text{ times}) \tag{2.14}$$

and the corresponding mappings

$$\mathbb{1}_{\mathcal{W}} = (\mathbb{1}_{\mathcal{S}}, \ldots, \mathbb{1}_{\mathcal{S}}) \in \mathcal{W} \qquad (\textbf{barycenter}) \tag{2.15a}$$

$$\Pi_{\mathcal{T}_0}(Z) = \big(\Pi_{T_0}(Z_1), \ldots, \Pi_{T_0}(Z_{|I|})\big) \in \mathcal{T}_0, \qquad W \in \mathcal{W}, \quad Z \in \mathbb{R}^{|I||J|} \tag{2.15b}$$

$$R_W(Z) = \big(R_{W_1}(Z_1), \ldots, R_{W_{|I|}}(Z_{|I|})\big) \in \mathcal{T}_0, \qquad W \in \mathcal{W}, \quad Z \in \mathbb{R}^{|I||J|} \tag{2.15c}$$

$$\mathrm{Exp}_W(V) = \big(\mathrm{Exp}_{W_1}(V_1), \ldots, \mathrm{Exp}_{W_{|I|}}(V_{|I|})\big) \in \mathcal{W}, \quad W \in \mathcal{W}, \quad V \in \mathcal{T}_0 \tag{2.15d}$$

and $\exp_W, \mathrm{Exp}_W^{-1}, \exp_W^{-1}$ similarly defined based on (2.11*b*), (2.12*a*) and (2.12*b*). Finally, we define the *geometric mean* of assignment vectors [1, lemma 5]

$$\mathcal{G}_i^w(W) = \mathrm{Exp}_{W_i}\Big(\sum_{k \in \mathcal{N}_i} w_{ik} \mathrm{Exp}_{W_i}^{-1}(W_k)\Big) = \exp_{W_i}\Big(\log \frac{\prod_{k \in \mathcal{N}_i} W_k^{w_{ik}}}{W_i}\Big), \qquad i \in I \tag{2.16}$$

where $W_k^{w_{ik}}$ is the componentwise exponentiation of W_k with w_{ik}.

Using this setting, the assignment flow accomplishes image labeling as follows. Based on (2.1), (2.4), distance vectors

$$D = (D_1, \ldots, D_{|I|}) \in \mathbb{R}^{|I||J|} \qquad (\textbf{distance vectors}) \tag{2.17a}$$

$$D_i = \big(d(f_i, g_1), \ldots, d(f_i, g_{|J|})\big)^\top, \quad i \in I \tag{2.17b}$$

are defined and mapped to

$$L(W) = \exp_W(-\frac{1}{\rho} D) \in \mathcal{W}, \qquad (\textbf{likelihood vectors}) \tag{2.18a}$$

$$L_i(W_i) = \frac{W_i e^{-\frac{1}{\rho} D_i}}{\langle W_i, e^{-\frac{1}{\rho} D_i} \rangle}, \quad \rho > 0, \qquad i \in I, \tag{2.18b}$$

where ρ is a user parameter to normalize the distances induced by the specific features f_i at hand. This representation of the data is regularized by local geometric smoothing to obtain

$$S(W) \in \mathcal{W}, \qquad S_i(W) = \mathcal{G}_i^w\big(L(W)\big), \quad i \in I, \qquad (\textbf{similarity vectors}) \tag{2.19}$$

which in turn evolves the assignment vectors W_i, $i \in I$ through the

$$\dot{W} = R_W\big(S(W)\big), \qquad W(0) = \mathbb{1}_{\mathcal{W}}. \qquad (\textbf{assignment flow}). \tag{2.20}$$

Methods for numerically integrating this flow are examined in the following sections.

3. Geometric Runge–Kutta integration

We apply the general approach of [11] to our problem. For background and more details, we refer to [10] and [16, chapter 4].

3.1. General approach

In order to apply Lie group methods to the integration of an ODE on a manifold $\mathcal{M}$, one has to check first if the ODE can be represented properly. Let

$$\Lambda \colon \mathrm{G} \times \mathcal{M} \to \mathcal{M} \tag{3.1a}$$

denote the action of a Lie group G on $\mathcal{M}$ satisfying

$$\Lambda(e, p) = p \qquad \text{with identity } e \in \mathrm{G}, \tag{3.1b}$$

$$\Lambda(g_1 \cdot g_2, p) = \Lambda(g_1, \Lambda(g_2, p)), \qquad \text{for all } g_1, g_2 \in \mathrm{G},\ p \in \mathcal{M}. \tag{3.1c}$$

Furthermore, let $\mathfrak{g}$ denote the Lie algebra of G, $\mathfrak{X}(\mathcal{M})$ the set of all smooth vector fields on $\mathcal{M}$,

$$\lambda \colon \mathfrak{g} \times \mathcal{M} \to \mathcal{M} \tag{3.2}$$

a smooth function and λ_* the induced map defined by

$$\lambda_* \colon \mathfrak{g} \to \mathfrak{X}(\mathcal{M}), \qquad (\lambda_* v)_p = \frac{\mathrm{d}}{\mathrm{d}t} \lambda(tv, p)\big|_{t=0} \qquad \text{for all } v \in \mathfrak{g},\ p \in \mathcal{M}. \tag{3.3}$$

Then λ is a Lie algebra action if the induced map λ_* is a Lie algebra homomorphism, i.e. λ_* is linear and satisfies $\lambda_*[u,v] = [\lambda_* u, \lambda_* v]$, $u, v \in \mathfrak{g}$, with the Lie brackets on $\mathfrak{g}$ and $\mathfrak{X}(\mathcal{M})$ on the left-hand side and the right-hand side, respectively. In particular, based on a Lie group action Λ, a Lie algebra action is given by [11, lemma 4]

$$\lambda(v,p) = \Lambda(\exp_{\mathrm{G}}(v), p), \tag{3.4}$$

where $\exp_{\mathrm{G}}\colon \mathfrak{g} \to \mathrm{G}$ denotes the exponential map of G. Thus, for this choice of λ, the induced map (3.3) is given by [11, theorem 5]

$$(\lambda_* v)_p = \frac{\mathrm{d}}{\mathrm{d}t}\Lambda(\exp_{\mathrm{G}}(tv), p)\big|_{t=0} \qquad \text{for all } v \in \mathfrak{g},\ p \in \mathcal{M}. \tag{3.5}$$

Now, given an ODE on $\mathcal{M}$, the *basic assumption* underlying the application of Lie group methods is the existence of a function $f\colon \mathbb{R} \times \mathcal{M} \to \mathfrak{g}$ such that the ODE admits the representation

$$\dot{y} = \big(\lambda_* f(t,y)\big)_y, \qquad y(0) = p. \tag{3.6}$$

For sufficiently small t, the solution of (3.6) then can be parametrized as

$$y(t) = \lambda\big(v(t), p\big), \tag{3.7a}$$

where $v(t) \in \mathfrak{g}$ satisfies the ODE

$$\dot{v} = (\mathrm{dexp}_{\mathrm{G}}^{-1})_v\big(f(t, \lambda(v,p))\big), \qquad v(0) = 0, \tag{3.7b}$$

with the inverse differential $(\mathrm{dexp}_{\mathrm{G}}^{-1})_v$ of $\exp_{\mathrm{G}}$ evaluated at $v \in \mathfrak{g}$. A major advantage of the representation (3.7) is that the task of numerical integration concerns the ODE (3.7*b*) evolving on the vector space $\mathfrak{g}$, rather than the original ODE evolving on the manifold $\mathcal{M}$. As a consequence, established methods can be applied to (3.7*b*), possibly after approximating $\mathrm{dexp}_{\mathrm{G}}^{-1}$ by a truncated series in a computationally feasible form.

3.2. Application to the assignment flow

Assume an ODE on $\mathcal{S}$ defined by (2.6) is given. The application of the approach of section 3.1 is considerably simplified by identifying $\mathrm{G} = T_0$ with the *flat* tangent space (2.7) and consequently also $T_0 \cong \mathfrak{g} = T_e\mathrm{G}$. One easily verifies that the action $\Lambda\colon T_0 \times \mathcal{S} \to \mathcal{S}$ defined as

$$\Lambda(v,p) = \exp_p(v), \tag{3.8}$$

with the right-hand side given by (2.12*a*), satisfies (3.1), i.e.

$$\Lambda(0,p) = p, \tag{3.9a}$$

$$\Lambda(v_1 + v_2, p) = \frac{p e^{v_1+v_2}}{\langle p, e^{v_1+v_2}\rangle} = \Lambda(v_1, \Lambda(v_2,p)). \tag{3.9b}$$

Proposition 3.1. *The solution $W(t)$ to assignment flow (2.20) emanating from any $W_0 = W(0)$ admits the representation*

$$W(t) = \exp_{W_0}\big(V(t)\big) \tag{3.10a}$$

where $V(t) \in \mathcal{T}_0$ solves

$$\dot{V} = \Pi_{\mathcal{T}_0} S\big(\exp_{W_0}(V)\big), \qquad V(0) = 0. \tag{3.10b}$$

Proof. Since geodesics through $0 \in T_0$ in directions $v \in T_0$ have the form $\gamma(t) = tv$, the differential of the exponential map of $T_0 = \mathrm{G}$, $\exp_{T_0}(v) = \gamma(1) = v$, is the identity and thus (3.5) gives

$$(\lambda_* v)_p = \frac{\mathrm{d}}{\mathrm{d}t} \Lambda\big(\gamma(t), p\big)\big|_{t=0} = R_p(v), \tag{3.11}$$

with R_p defined by (2.9*b*). As a result, the basic assumption (3.6) concerns ODEs on $\mathcal{S}$ that admit the representation

$$\dot{p} = R_p\big(f(t,p)\big), \qquad p(0) = p_0, \tag{3.12}$$

for some function $f\colon \mathbb{R} \times \mathcal{S} \to T_0$ and some $p_0 \in \mathcal{S}$. Since $\lambda = \Lambda$ by (3.4), the parametrization (3.7) reads

$$p(t) = \Lambda\big(v(t), p_0\big) \tag{3.13a}$$

where $v(t) \in T_0$ solves

$$\dot{v} = f\big(t, \Lambda(v, p_0)\big), \qquad v(0) = 0. \tag{3.13b}$$

This setting extends to the assignment flow by defining (see (2.15)) $\Lambda\colon \mathcal{T}_0 \times \mathcal{W} \to \mathcal{W}$ and $\lambda_*\colon \mathcal{T}_0 \to \mathcal{T}_0$ as

$$\Lambda(V, W) = \exp_W(V), \qquad \big(\lambda_*(V)\big)_W = R_W(V). \tag{3.14}$$

The basic assumption (3.6) then reads

$$\dot{W} = \big(\lambda_* f(t, W)\big)_W = R_W\big(\Pi_{\mathcal{T}_0} S(W)\big) \overset{(2.10)}{=} R_W\big(S(W)\big), \qquad W(0) = \mathbb{1}_{\mathcal{W}}, \tag{3.15}$$

which is the assignment flow (2.20). Due to (3.6), for any $W_0 = W(0)$, it admits the representation

$$W(t) = \Lambda\big(V(t), W_0\big), \tag{3.16a}$$

where $V(t) \in \mathcal{T}_0$ solves

$$\dot{V} = f\big(t, \Lambda(V, W_0)\big) = \Pi_{\mathcal{T}_0} S(\exp_{W_0}(V)), \qquad V(0) = 0, \tag{3.16b}$$

which is (3.10). □

Remark 3.2. While the basic formulation (2.20) of the assignment flow is *autonomous*, we keep in what follows the explicit time dependency of the function $f(t, \cdot)$ of the parametrization (3.10), because in more advanced scenarios the flow may become *non-autonomous*. A basic example concerns *unsupervised* problems [9] where labels *vary*, and hence the distance vectors (2.17) and in turn the vector field defining the assignment flow depend on t.

Using the above representation and taking into account the simplifications of the general approach of section 3.1, the RKMK algorithm [11] for integrating the assignment flow from $t = 0$ to $t = h$ is specified as follows. Let $a_{i,j}, b_j$ be the coefficients of an s-stage, qth order

classical RK method satisfying the consistency condition $c_i = \sum_{j\in[s]} a_{i,j}$ [12, chapter II]. Starting from any point

$$W_0 = W(0), \tag{3.17a}$$

the algorithm amounts to compute the vector fields

$$U^i = h\sum_{j\in[s]} a_{i,j}\widetilde{U}^j, \qquad i \in [s] \tag{3.17b}$$

$$\widetilde{U}^i = f\big(hc_i, \Lambda(U^i, W_0)\big), \qquad i \in [s] \tag{3.17c}$$

$$V = h\sum_{j\in[s]} b_j\widetilde{U}^j \tag{3.17d}$$

and the update

$$W(h) = \Lambda(V, W_0). \tag{3.17e}$$

Replacing $W_0 \leftarrow W(h)$, computing the update and iterating this procedure generates the sequence $(W^{(k)})_{k\geqslant 0}$ which approximates $(W(t_k))_{k\geqslant 0}$, $t_k = kh$.

A *s-stage RKMK scheme* is specified using the corresponding Butcher tableau of the form

0				
c_2	a_{21}			
c_3	a_{31}	a_{32}		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
c_s	a_{s1}	a_{s2}	$\dots$	$a_{s(s-1)}$
	b_1	b_2	$\dots$	b_{s-1} b_s

Specifically, we consider the following *explicit RKMK schemes* of order 1, 2, 3, 4:

0	
	1

Forward Euler (**FE**)

0		
1	1	
	1/2	1/2

Heun-2 (**H2**)

0			
1/3	1/3		
2/3	0	2/3	
	1/4	0	3/4

Heun-3 (**H3**)

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Classical RK (**RK4**)

Note the increasing number of stages that raise the approximation order. This comes at a price, however, because each stage evaluates at step (3.17*c*) the right-hand side of (3.10*b*) which is the most expensive operation. As a consequence, it is not clear *a priori* if using a multi-stage scheme and a larger step size h is superior to a simpler scheme that is evaluated more frequently using a smaller step size.

In addition to the above explicit schemes, we consider the simplest *implicit RKMK scheme*

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

Backward Euler (**BE**)

Implicit schemes are known to be stable for much larger step sizes. Yet, they require to solve at every step a fixed point equation which is done by an iterative inner loop.

The performances of these numerical schemes are examined in section 6.

4. Linear assignment flow, exponential integrator

The ODE (3.10*b*) which parametrizes the assignment flow together with (3.10*a*), evolves on a linear space but is a *nonlinear* system of ordinary differential equations. In this section, we provide an *approximate representation* of the assignment flow within time intervals through a *linear* ODE evolving on the tangent space (section 4.1), and a corresponding numerical scheme (section 4.2).

The resulting flow on the assignment *manifold* is still nonlinear, though. The basic idea is to capture locally a major part of the nonlinearity of the (full) assignment flow, by a linear ODE on the tangent space that enables to apply alternative integration schemes.

4.1. Linear assignment flow

Our ansatz has two ingredients. Firstly, we adopt the parametrization

$$W(t) = \mathrm{Exp}_{W_0}\big(V(t)\big), \qquad V(t) \in \mathcal{T}_0 \tag{4.1}$$

of the solution $W(t)$ to the assignment flow by a trajectory in the tangent space $\mathcal{T}_0$, similar to (3.10*a*), except for using the 'true' exponential map (2.11*a*) and (2.15*d*), respectively, corresponding to the underlying affine connection. Secondly, we use an *affine approximation* of the vector field on the right-hand side of (2.20), that defines the assignment flow. The following corresponding definition generalizes the flow studied by [17] from the barycenter to arbitrary base points W_0, and from a flow on $\mathcal{S}$ to a flow on $\mathcal{W}$.

Definition 4.1 (linear assignment flow). We call *linear assignment flow* every flow induced by an ODE of the form

$$\dot{W} = R_W\Big(s_0 + S_0 R_{W_0} \log \frac{W}{W_0}\Big), \qquad W(0) = W_0 \in \mathcal{W}, \tag{4.2}$$

with a fixed vector s_0 and a fixed matrix S_0, for arbitrary W_0.

An important property of the flow (4.2)—which explains its name—is the possibility to parametrize it by a *linear* ODE evolving on the tangent space $\mathcal{T}_0$.

Proposition 4.2. *The linear assignment flow (4.2) admits the representation*

$$W(t) = \mathrm{Exp}_{W_0}\big(V(t)\big), \tag{4.3a}$$

where $V(t) \in \mathcal{T}_0$ *solves*

$$\dot{V} = R_{W_0}(s_0 + S_0 V), \qquad V(0) = 0. \tag{4.3b}$$

Proof. Parametrization (4.1) yields

$$V(t) = \operatorname{Exp}_{W_0}^{-1}(W(t)) \overset{(2.11b)}{=} R_{W_0} \log \frac{W(t)}{W_0} \tag{4.4}$$

and by differentiation

$$\dot{V}(t) = R_{W_0}\Big(\frac{\dot{W}(t)}{W(t)}\Big). \tag{4.5a}$$

Solving (4.2) for $\frac{\dot{W}}{W}$ after inserting (2.15*c*), and substitution in the preceding equation gives

$$= R_{W_0}\Big(s_0 + S_0 \operatorname{Exp}_{W_0}^{-1}\big(W(t)\big) - \big\langle W(t), s_0 + S_0 \operatorname{Exp}_{W_0}^{-1}\big(W(t)\big)\big\rangle \mathbb{1}\Big), \tag{4.5b}$$

and since $R_{W_0}\mathbb{1} = 0$ by (2.9*b*)

$$= R_{W_0}\Big(s_0 + S_0 \operatorname{Exp}_{W_0}^{-1}\big(W(t)\big)\Big) \overset{(4.4)}{=} R_{W_0}\big(s_0 + S_0 V(t)\big). \tag{4.5c}$$

The initial condition follows from $V(0) = \operatorname{Exp}_{W_0}^{-1}(W_0) = 0$. □

Remark 4.3. Note that, despite the linearity of (4.3*b*), the resulting flow (4.3*a*) solving (4.2) is *nonlinear*. Thus, one may hope to capture the major nonlinearity of the full assignment flow (2.20) by a linear ODE on the tangent space, at least locally in some time interval. Within this interval, the evaluation of (2.19) is *not* required, and the linearity of the tangent space ODE (4.3*b*) can be exploited for integration.

We conclude this section by computing the natural choice

$$s_0 = S(W_0), \qquad S_0 = \mathrm{d}S_{W_0} \tag{4.6}$$

of the parameters of the linear assignment flow (4.2) in explicit form, where s_0 is immediate due to (2.19), but the Jacobian $S_0 = \mathrm{d}S_{W_0}$ of $S(W)$, evaluated at W_0, is not.

Proposition 4.4. *Let $S(W) \in \mathbb{R}^{|I||J|}$ denote the global similarity vector obtained by stacking the local similarity vectors $S_1(W), \ldots, S_{|I|}(W)$ of (2.19). Then, with*

$$s_0 = S(W_0), \qquad s_{0i} = S_i(W_0), \qquad s_{0i,j} = S_{ij}(W_0) = \big(S_i(W_0)\big)_j, \quad i \in I,\ j \in J \tag{4.7a}$$

and the replicator map $R_{s_{0,i}}$ defined by (2.9b), the Jacobian of $S(W)$ at $W_0 \in \mathcal{W}$ is given by

$$S_0 = \mathrm{d}S_{W_0} = \begin{pmatrix} A_{11}(W_0) & \ldots & A_{1|I|}(W_0) \\ \vdots & \ddots & \vdots \\ A_{|I|1}(W_0) & \ldots & A_{|I||I|}(W_0) \end{pmatrix} \in \mathbb{R}^{|I||J| \times |I||J|}, \tag{4.7b}$$

where the action of each $|J| \times |J|$ block matrix has the form

$$A_{ik}(W_0)(V_k) = \begin{cases} w_{ik} R_{s_{0,i}}\Big(\frac{V_k}{W_{0,k}}\Big), & \text{if } k \in \mathcal{N}_i \\ 0, & \text{if } k \notin \mathcal{N}_i \end{cases}, \qquad W_0 \in \mathcal{W}, \quad V_k \in T_0, \qquad i,k \in I \tag{4.7c}$$

and the non-zero entries if $k \in \mathcal{N}_i$ *(using (4.7a))*

$$A_{ik,jl}(W_0) = \big(A_{ik}(W_0)\big)_{jl} = w_{ik} \begin{cases} (1 - s_{0i,j}) \frac{s_{0i,j}}{W_{0,kj}}, & \text{if } j = l \\ -s_{0i,j} \frac{s_{0i,l}}{W_{0,kl}}, & \text{if } j \neq l \end{cases}, \qquad j, l \in J. \tag{4.7d}$$

The proof follows below after two preparatory lemmata.

Lemma 4.5. *Let* $p \in \mathcal{S}$*. Then the differential of* $\exp_p \colon T_0 \to \mathcal{S}$ *at* $u \in T_0$ *applied to* $v \in T_0$ *is given by*

$$\mathrm{d} \exp_p(u)(v) = R_{\exp_p(u)}(v), \quad u, v \in T_0,\ p \in \mathcal{S}. \tag{4.8}$$

Moreover, we have

$$\exp_p(v - \log p) = \exp_{\mathbb{1}_\mathcal{S}}(v), \quad v \in T_0,\ p \in \mathcal{S}. \tag{4.9}$$

Proof. Let $\gamma(t)$ be a smooth curve in T_0 with $\gamma(0) = u$ and $\dot\gamma(0) = v$. Using (2.12*a*), we compute

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{p e^{\gamma(t)}}{\langle p, e^{\gamma(t)} \rangle} \bigg|_{t=0} = \frac{\langle p, e^u \rangle p v e^u - \langle p, v e^u \rangle p e^u}{\langle p, e^u \rangle^2} = \frac{p e^u}{\langle p, e^u \rangle} \Big(v - \frac{\langle p e^u, v \rangle}{\langle p, e^u \rangle} \mathbb{1} \Big) = R_{\exp_p(u)}(v), \tag{4.10}$$

which is (4.8). As for (4.9), using the representation

$$p \overset{(2.12a)}{=} \exp_{\mathbb{1}_\mathcal{S}}(\log p) = \exp_{\mathbb{1}_\mathcal{S}}(\Pi_{T_0} \log p), \tag{4.11}$$

where the last equation takes into account remark 2.1, we obtain

$$\exp_p(v - \log p)$$

$$= \exp_p(v - \Pi_{T_0} \log p) \overset{(4.11)}{=} \exp_{\exp_{\mathbb{1}_\mathcal{S}}(\Pi_{T_0} \log p)}(v - \Pi_{T_0} \log p) \tag{4.12a}$$

$$\overset{(3.8)}{=} \Lambda\big(v - \Pi_{T_0} \log p, \Lambda(\Pi_{T_0} \log p, \mathbb{1}_\mathcal{S})\big) \overset{(3.9)}{=} \Lambda(v - \Pi_{T_0} \log p + \Pi_{T_0} \log p, \mathbb{1}_\mathcal{S}) \tag{4.12b}$$

$$= \Lambda(v, \mathbb{1}_\mathcal{S}) = \exp_{\mathbb{1}_\mathcal{S}}(v). \tag{4.12c}$$

□

We use this lemma to represent the similarity vectors in a convenient form for subsequently proving proposition 4.4.

Lemma 4.6. *The similarity vectors (2.19) admit the representation*

$$S_i(W) = \exp_{\mathbb{1}_\mathcal{S}} \Big(\sum_{k \in \mathcal{N}_i} w_{ik} \big(\log W_k - \frac{1}{\rho} D_k \big) \Big), \quad i \in I. \tag{4.13}$$

Proof. By (2.19) and (2.16), we obtain

$$S_i(W) = \exp_{W_i} \Big(\log \frac{\prod_{k \in \mathcal{N}_i} L_k(W_k)^{w_{ik}}}{W_i} \Big) = \exp_{W_i} \Big(\sum_{k \in \mathcal{N}_i} w_{ik} \log L_k(W_k) - \log W_i \Big) \tag{4.14a}$$

$$\overset{(2.18b)}{=} \exp_{W_i}\Big(\sum_{k\in\mathcal{N}_i} w_{ik}\Big(\log W_k - \frac{1}{\rho}D_k - \log\big(\langle W_k, e^{-\frac{1}{\rho}D_k}\rangle\big)\mathbb{1}\Big) - \log W_i\Big) \tag{4.14b}$$

using again $\exp_p(v+\lambda\mathbb{1}) = \exp_p(v)$ for all $\lambda\in\mathbb{R}$, $v\in T_0$, $p\in\mathcal{S}$ (see remark 2.1)

$$= \exp_{W_i}\Big(\sum_{k\in\mathcal{N}_i} w_{ik}\big(\log W_k - \frac{1}{\rho}D_k\big) - \log W_i\Big) \overset{(4.9)}{=} \exp_{\mathbb{1}_\mathcal{S}}\Big(\sum_{k\in\mathcal{N}_i} w_{ik}\big(\log W_k - \frac{1}{\rho}D_k\big)\Big). \tag{4.14c}$$

□

Proof of proposition 4.4. Setting

$$S_i(W) \overset{(4.13)}{=} \exp_{\mathbb{1}_\mathcal{S}} \circ Z_i(W), \qquad Z_i(W) = \sum_{k\in\mathcal{N}_i} w_{ik}\big(\log W_k - \frac{1}{\rho}D_k\big), \tag{4.15}$$

we compute using a smooth curve $\gamma(t)$ in $\mathcal{W}$ with $\gamma(0)=W$ and $\dot\gamma(0)=V$,

$$\mathrm{d}Z_i(W)(V) = \frac{\mathrm{d}}{\mathrm{d}t} Z_i\big(\gamma(t)\big)\big|_{t=0} = \sum_{k\in\mathcal{N}_i} w_{ik}\frac{\mathrm{d}}{\mathrm{d}t}\log\big(\gamma(t)\big)\big|_{t=0} = \sum_{k\in\mathcal{N}_i} w_{ik}\frac{V_k}{W_k}. \tag{4.16}$$

Thus, using (4.15) and (4.8) gives

$$\mathrm{d}S_i(W)(V) \overset{(4.15)}{=} \mathrm{d}\exp_{\mathbb{1}_\mathcal{S}}\big(Z_i(W)\big)\big(\mathrm{d}Z_i(W)(V)\big) \overset{(4.8),(4.15)}{=} R_{S_i(W)}\big(\mathrm{d}Z_i(W)(V)\big), \tag{4.17a}$$

and using the linearity of the map $R_{S_i(W)}$ and (4.16),

$$= \sum_{k\in\mathcal{N}_i} w_{ik} R_{S_i(W)}\Big(\frac{V_k}{W_k}\Big), \tag{4.17b}$$

which proves (4.7*c*). Inserting $R_{S_i(W)}$ due to (2.9*b*) yields (4.7*d*).

The following section specifies an alternative integration scheme for the linear assignment flow (4.2). Its approximation properties are numerically examined in section 6.

4.2. Exponential integrator

We focus on the linear ODE (4.3*b*) that together with (4.3*a*) determines the linear assignment flow due to (4.2). The solution to (4.3*b*) is given by Duhamel's formula [18],

$$V(t) = \operatorname{expm}(tA)\,V(0) + \int_0^t \operatorname{expm}\big((t-\tau)A\big)a\,\mathrm{d}\tau \quad \text{where} \quad A = R_{W_0}S_0, \quad a = R_{W_0}s_0, \tag{4.18}$$

which involves the matrix exponential of the matrix A of dimension $|I||J| \times |I||J|$ (square of number of pixels $\times$ number of labels), which can be quite large in image labeling problems (10^4–10^7 variables). Explicitly computing the matrix exponential is neither feasible, because it is dense even if A is sparse, nor required in view of the multiplication with the vector a. Rather, taking into account $V(0)=0$ and that uniformly converging series can be integrated term by term, we set $t=T$ large enough and evaluate

$$V(T) = \int_0^T \operatorname{expm}\big((T-\tau)A\big)a\,\mathrm{d}\tau = \operatorname{expm}(TA)\int_0^T \sum_{k=0}^{\infty} \frac{(-\tau A)^k}{k!}a\,\mathrm{d}\tau \tag{4.19a}$$

$$= \operatorname{expm}(TA)\sum_{k=0}^{\infty}\Big[\frac{\tau(-\tau A)^k}{(k+1)!}\Big]_{\tau=0}^{T} a = T\operatorname{expm}(TA)\sum_{k=0}^{\infty}\frac{(-TA)^k}{(k+1)!}a \tag{4.19b}$$

$$= T\varphi_1(TA)a \tag{4.19c}$$

where φ_1 is the entire function

$$\varphi_1(z) = \sum_{k=0}^{\infty}\frac{z^k}{(k+1)!} = \frac{e^z-1}{z}, \tag{4.20}$$

whose series representation yields valid expressions (4.19*c*) also if the matrix A is singular.

We refer to [19] for a detailed exposition of matrix functions and to [20] and [19, sections 10 and 13] for a survey of methods for computing the matrix exponential and the product of matrix functions times a vector. For large problem sizes, the established methods of the two latter references are known to deteriorate, however, and methods based on Krylov subspaces have been developed [13, 14] and become the method of choice in connection with exponential integrators [15].

We confine ourselves with sketching below a state-of-the-art method [21] for the approximate numerical evaluation of (4.19). The evaluation of its performance for integrating the linear assignment flow and a comparison to the methods of section 5.1, are reported in section 6. A more comprehensive evaluation of further recent methods for evaluating (4.19) that cope with large problem sizes as well (e.g. [22]), is beyond the scope of this paper.

In order to compute approximately $\varphi_1(TA)a$, one considers the Krylov subspace

$$\mathcal{K}_m = \operatorname{span}\{a, Aa, \ldots, A^{m-1}a\}, \tag{4.21}$$

with orthogonal basis $\mathcal{V}_m = (v_1, \ldots, v_m)$ arranged as column vectors of an orthogonal matrix $\mathcal{V}_m$ and computed using the basic Arnoldi iteration [13]. The action of A is approximated by

$$\mathcal{H}_m = \mathcal{V}_m^{\top} A \mathcal{V}_m \tag{4.22}$$

which in turn yields the approximation

$$\varphi_1(A)a \approx \varphi_1(\mathcal{V}_m\mathcal{H}_m\mathcal{V}_m^{\top})a = \mathcal{V}_m\varphi_1(\mathcal{H}_m)\mathcal{V}_m^{\top}a = \|a\|\mathcal{V}_m\varphi_1(\mathcal{H}_m)e_1, \tag{4.23}$$

where $e_1 = (1, 0, \ldots, 0)^{\top}$ denotes the first unit vector and the last equality is implied by the Arnoldi iteration producing $\mathcal{V}_m, \mathcal{H}_m$, which sets $v_1 = a/\|a\|$. Note that φ_1 merely has to be applied to the much smaller $m \times m$ matrix $\mathcal{H}_m$, which can be safely and efficiently computed using standard methods [19, 20]. The vector $\varphi_1(\mathcal{H}_m)e_1$ can be recovered [23, theorem 1] in form of the upper m entries of the last column of $\operatorname{expm}(\widehat{\mathcal{H}}_m)$ with the extended matrix

$$\widehat{\mathcal{H}}_m = \begin{pmatrix} \mathcal{H}_m & e_1 \\ 0 & 0 \end{pmatrix}. \tag{4.24}$$

If the degree of the minimal polynomial of a (i.e. the nonzero monic polynomial p of lowest degree such that $p(A)a = 0$) is equal to m, then the approximation (4.23) is even exact [13, theorem 3.6].

5. Step sizes, adaptivity

We specify in this section the step size selection for the numerical RKMK schemes of section 3. In addition, for the *linear assignment flow* (section 4.1), we conduct a local error analysis in section 5.1 for RK schemes based on the linearity of the tangent space ODE that governs this flow. A corresponding explicit error estimate enables to determine a sequence $(h_k)_{k\geqslant 0}$ of step sizes that ensure a prespecified local accuracy at each step k.

In order to determine step sizes for the *nonlinear assignment flow* (2.20), we proceed differently, because the corresponding vector field depends nonlinearly on the current iterate and estimating local Lipschitz constants will be expensive and less sharp. We therefore adapt in section 5.2 classical methods for local error estimation and step size selection for nonlinear ODEs based on *embedded* Runge–Kutta methods [12, section II.4], to the *geometric* RKMK methods of section 3.

The experimental evaluation of both approaches is reported in section 6.

5.1. Linear assignment flow

We focus on the linear ODE (4.3*b*) that together with (4.3*a*) determines the linear assignment flow (4.2). Due to its approximation property demonstrated in section 6.3.1, we only consider the linearization point $W_0 = \mathbb{1}_{\mathcal{W}}$. Since the ODE (4.2) evolves on the linear space $\mathcal{T}_0$, we apply the *classical* s-stage explicit RK scheme, rather than the *geometric* s-stage RKMK scheme (3.17), to obtain

$$U^i = R_{W_0} s_0 + R_{W_0} S_0 \Big(V^{(k)} + h \sum_{j\in[s-1]} a_{ij} U^j \Big), \qquad i \in [s], \tag{5.1a}$$

$$V^{(k+1)} = V^{(k)} + h \sum_{i\in[s]} b_i U^i, \qquad V^{(0)} = V(0) = 0. \tag{5.1b}$$

Specifically, regarding the *explicit schemes* listed at the end of section 3.2 in terms of their Butcher tableaus, consecutively inserting (5.1*a*) into (5.1*b*) yields with the shorthands a, A defined by (4.18),

$$V^{(k+1)} = ha + (I + hA)V^k, \qquad (\mathbf{FE}) \tag{5.2a}$$

$$V^{(k+1)} = \Big(h + \frac{h^2}{2}A\Big)a + \Big(I + hA + \frac{h^2}{2}A^2\Big)V^{(k)}, \qquad (\mathbf{H2}) \tag{5.2b}$$

$$V^{(k+1)} = \Big(h + \frac{h^2}{2}A + \frac{h^3}{6}A^2\Big)a + \Big(I + hA + \frac{h^2}{2}A^2 + \frac{h^3}{6}A^3\Big)V^{(k)}, \qquad (\mathbf{H3}) \tag{5.2c}$$

$$V^{(k+1)} = \Big(h + \frac{h^2}{2}A + \frac{h^3}{6}A^2 + \frac{h^4}{24}A^3\Big)a + \Big(I + hA + \frac{h^2}{2}A^2 + \frac{h^3}{6}A^3 + \frac{h^4}{24}A^4\Big)V^{(k)}. \qquad (\mathbf{RK4}) \tag{5.2d}$$

Comparison with (4.18) shows that due to the linearity of the ODE, each scheme results in a corresponding Taylor series approximation, depending on its order q, of the equation

$$V(t_{k+1}) = h\varphi_1(hA)a + \operatorname{expm}(hA)V(t_k) \tag{5.3}$$

that is,

$$V^{(k+1)} = h\Big(\sum_{i=0}^{q-1} \frac{(hA)^i}{(i+1)!}\Big)a + \Big(\sum_{i=0}^{q} \frac{(hA)^i}{i!}\Big)V^{(k)} \tag{5.4a}$$

$$= p_{1,q}(hA)a + p_{2,q}(hA)V^{(k)}, \tag{5.4b}$$

with matrix-valued polynomials $p_{1,q}, p_{2,q}$. Our strategy for choosing the step size h is based on the local error estimate specified below as theorem 5.2 and prepared by the following Lemma.

Lemma 5.1 ([24, equation (8.4.8)]). *Let $q \in \mathbb{N}$ and $t \in \mathbb{R}$. Then*

$$\sum_{i=0}^{q} \frac{t^i}{i!} = e^t \frac{\Gamma(1+q,t)}{q!} \tag{5.5}$$

with the incomplete Gamma function

$$\Gamma(1+q,t) = \int_t^\infty \tau^q e^{-\tau} \mathrm{d}\tau. \tag{5.6}$$

Theorem 5.2. *Let $V(t)$ solve (4.2) with $W_0 = \mathbb{1}_{\mathcal{W}}$, and let $(V^{(k)})_{k>0}$ be a sequence generated by a RK scheme (5.1) of order q. Set $V(t_k) = V^{(k)}$ in (5.3). Then $V(t_{k+1})$ in (5.3) is the exact value of the linear assignment flow emanating from $V^{(k)}$, and regarding (5.4) the local error estimate*

$$\|V(t_{k+1}) - V^{(k+1)}\| \leqslant e^{h\|A\|}\Big(1 - \frac{\Gamma(1+q,h\|A\|)}{q!}\Big)\Big(\frac{\|a\|}{\|A\|} + \|V^{(k)}\|\Big) \tag{5.7a}$$

$$< e^{h\|A\|}(1 - e^{-h\|A\|})^{(1+q)}\Big(\frac{\|a\|}{\|A\|} + \|V^{(k)}\|\Big) \tag{5.7b}$$

holds, where $\Gamma(1+q,h\|A\|)$ is given by (5.6) and $\|A\|$ denotes the spectral norm of the matrix $A = R_{W_0}(S_0)$.

Proof. Using (5.3) and (5.4) and $V(t_k) = V^{(k)}$, we bound the local error by

$$\|V(t_{k+1}) - V^{(k+1)}\| \leqslant \|h\varphi_1(hA) - p_{1,q}(hA)\|\|a\| + \|\mathrm{expm}(hA) - p_{2,q}(hA)\|\|V^{(k)}\|, \tag{5.8a}$$

and inserting the series (5.4a) gives

$$\leqslant h\Big(\sum_{i=q}^{\infty} \frac{(h\|A\|)^i}{(i+1)!}\Big)\|a\| + \Big(\sum_{i=q+1}^{\infty} \frac{(h\|A\|)^i}{i!}\Big)\|V^{(k)}\|. \tag{5.8b}$$

Both series absolutely converge for any h. By Lemma 5.1, we have

$$\sum_{i=q}^{\infty} \frac{t^i}{(i+1)!} \overset{j=i+1}{=} \sum_{j=q+1}^{\infty} \frac{1}{t} \cdot \frac{t^j}{j!} = \frac{e^t}{t}\Big(1 - \frac{\Gamma(1+q,t)}{q!}\Big), \tag{5.9a}$$

$$\sum_{i=q+1}^{\infty} \frac{t^i}{i!} = e^t\Big(1 - \frac{\Gamma(1+q,t)}{q!}\Big). \tag{5.9b}$$

Applying these equations to (5.8*b*) yields

$$h\sum_{i=q}^{\infty} \frac{(h\|A\|)^i}{(i+1)!} = \frac{e^{h\|A\|}}{\|A\|}\Big(1 - \frac{\Gamma(1+q,h\|A\|)}{q!}\Big), \tag{5.10a}$$

$$\sum_{i=q+1}^{\infty} \frac{(h\|A\|)^i}{i!} = e^{h\|A\|}\Big(1 - \frac{\Gamma(1+q,h\|A\|)}{q!}\Big), \tag{5.10b}$$

and substitution into (5.8)

$$\|V(t_{k+1}) - V^{(k+1)}\| \leqslant e^{h\|A\|}\Big(1 - \frac{\Gamma(1+q,h\|A\|)}{q!}\Big)\Big(\frac{\|a\|}{\|A\|} + \|V^{(k)}\|\Big), \tag{5.11}$$

which is (5.7*a*). To show (5.7*b*), we use the representation

$$\frac{1}{p}\Gamma\Big(\frac{1}{p}, x^p\Big) \overset{(5.6)}{=} \frac{1}{p}\int_{x^p}^{\infty} t^{\frac{1}{p}-1}e^{-t}\mathrm{d}t \overset{t=\tau^p}{=} \int_{x}^{\infty} e^{-\tau^p}\mathrm{d}\tau \tag{5.12}$$

and the lower bound [25, corollary of theorem 1]

$$\frac{1}{\Gamma(1+1/p)}\int_x^{\infty} e^{-t^p}\mathrm{d}t > 1 - (1 - e^{-\alpha x^p})^{1/p}, \qquad \alpha \geqslant \max\big\{1, \big(\Gamma(1+1/p)\big)^{-p}\big\}, \tag{5.13}$$

that holds for all $x > 0$ and $0 < p \neq 1$, with the Gamma function

$$\Gamma(q) = \int_0^{\infty} \tau^{q-1}e^{-\tau}\mathrm{d}\tau, \qquad \Gamma(n+1) = n! \quad \text{if } n \in \mathbb{N}. \tag{5.14}$$

Put

$$x = h\|A\| \qquad \text{and} \qquad p = \frac{1}{1+q}. \tag{5.15}$$

Since $q \geqslant 1$ and $\big(\Gamma(1+1/p)\big)^{-p} = \Gamma(2+q)^{-\frac{1}{1+q}} = \big(\frac{1}{(q+1)!}\big)^{\frac{1}{1+q}} < 1$, we set $\alpha = 1$ in view of (5.13). Furthermore, we have

$$\Gamma(1+q,h\|A\|) \overset{(5.15)}{=} \Gamma\big(1/p,x\big) = \Gamma\big(1/p,(x^{1/p})^p\big) \overset{(5.12)}{=} p\int_{x^{1/p}}^{\infty} e^{-t^p}\mathrm{d}t \tag{5.16a}$$

$$\overset{(5.13),\alpha=1}{>} p\Gamma\big(1+1/p\big)\big(1 - \big(1 - e^{-x}\big)^{1/p}\big), \tag{5.16b}$$

and using $p\,\Gamma\big(1+1/p\big) \overset{(5.15)}{=} \frac{\Gamma(2+q)}{1+q} = \frac{(1+q)!}{1+q} = q!$, since q is integer,

$$= q!\big(1 - (1 - e^{-x})^{(1+q)}\big). \tag{5.16c}$$

Thus,

$$e^x\Big(1 - \frac{\Gamma\big(1/p,x\big)}{q!}\Big) < e^x\big(1 - e^{-x}\big)^{(1+q)} \tag{5.17}$$

which after substituting (5.15) and in turn into (5.11), proves (5.7*b*). □

Theorem 5.2 enables to control the local integration error by choosing the step size h, using the simple form of the bound (5.7), depending on the constants $\|a\|$, $\|A\|$ and the norm $\|V^{(k)}\|$ of the current iterate. Specifically, we choose

$$h = h_k \qquad \text{such that} \qquad \|V(t_{k+1}) - V^{(k+1)}\| \leqslant \tau \tag{5.18}$$

by (5.7), for some prespecified value τ. Inspecting the parametrization (4.3) shows that $\|V(t)\|$ grows—and hence the step sizes (5.18) *decrease*—until $W(t)$ is close enough to a vertex of $\mathcal{W}$ (which represents a labeling) and satisfies a termination criterion that stops the chosen iterative RK scheme (5.1).

In order to check how large $\|V(t)\|$ then will be, assume

$$W_i = (\varepsilon, \ldots, \varepsilon, 1 - (|J| - 1)\varepsilon, \varepsilon, \ldots, \varepsilon) \in \mathbb{R}^{|J|} \qquad \text{and} \qquad \varepsilon \ll \frac{1}{|J| - 1} \leqslant 1, \tag{5.19}$$

that is $W_{ij} \approx 1$ and $W_{il} \approx 0$ if $l \neq j$. Then with $W_0 = \mathbb{1}_{\mathcal{W}}$ and by (4.3) and (2.12*a*)

$$V_i = \operatorname{Exp}_{\mathbb{1}_{\mathcal{S}}}^{-1}(W_i) = R_{\mathbb{1}_{\mathcal{S}}}\big(\log W_i - \log \mathbb{1}_{\mathcal{S}}\big) = \frac{1}{|J|}\Big(\log W_i - \frac{1}{|J|}\langle \mathbb{1}, \log W_i\rangle \mathbb{1}\Big) \tag{5.20a}$$

$$\log W_i \approx (\log \varepsilon, \ldots, \log \varepsilon, 0, \log \varepsilon, \ldots, \log \varepsilon), \qquad \frac{1}{|J|}\langle \mathbb{1}, \log W_i\rangle \approx \frac{|J| - 1}{|J|} \log \varepsilon \approx \log \varepsilon \tag{5.20b}$$

and hence

$$\|V_i\| \approx \frac{1}{|J|} \log \frac{1}{\varepsilon}, \qquad \|V\| \approx \frac{|I|}{|J|} \log \frac{1}{\varepsilon}. \tag{5.20c}$$

Thus, as soon as the norm $\|V(t)\|$ has grown to the order $\log \frac{1}{\varepsilon}$, a termination criterion that checks if $W(t)$ is ε-close to some vertex of $\mathcal{W}$, will be satisfied.

Figure 2 quantitatively illustrates how much the factor $e^{h\|A\|}(1 - e^{h\|A\|})^{(1+q)}$ of the upper bound (5.7) overestimates the exact factor (5.11) computed in the proof of theorem 5.2, and hence how conservative (i.e. too small) the step size h_k will be chosen to achieve (5.18). The curves of figure 2 show that the estimate (5.7) is fairly tight and suited to adapt the step size. Furthermore, comparing the ordinate values of both panels for $q = 1$ and $q = 4$ shows that, in order to achieve a fixed accuracy τ in (5.18), using a *higher*-order RK scheme (5.1) enables to choose a *larger* step size.

5.2. Nonlinear assignment flow

Similar to the preceding section, we wish to select step sizes $(h_k)_{k\geqslant 0}$ in order to control the local error on the left-hand side of (5.7). Because an estimate like the right-hand side of (5.7) that is valid at each step k, is not available for the *nonlinear* assignment flow, we adapt established *embedded RK methods* [12, section II.5] to the geometric RKMK schemes (3.17).

The basic strategy is to evaluate *twice* step (3.17*d*)

$$V = h\sum_{j\in[s]} b_j \widetilde{U}^j, \qquad \widehat{V} = h\sum_{j\in[s]} \widehat{b}_j \widetilde{U}^j, \tag{5.21}$$

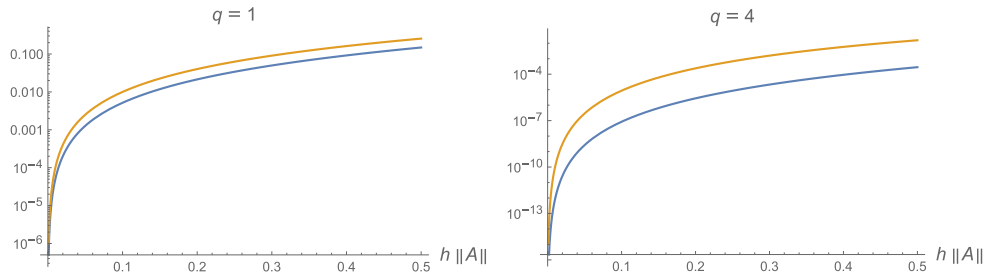


Figure 2. The factor $e^{h\|A\|}(1-e^{-h\|A\|})^{(1+q)}$ of the upper bound (5.7) (beige curve) and the numerically computed exact factor due to (5.11) (blue line), as a function of $h\|A\|$, for $q=1$ (left panel) and $q=4$ (right panel). The *relative* overestimation factor increases with the order q and leads to more or less conservative step size choices (5.18). Comparing the *absolute* ordinate values of both panels shows that in order to achieve (5.18), using a higher-order RK-scheme (5.1) enables to choose a larger step size.

using a second collection of coefficients $\widehat{b}_j$, $j \in s$, but with the *same* vector fields U^i, $\widetilde{U}^i$, $i \in [s]$. Thus each embedded method can be specified by *augmenting* the corresponding Butcher tableau accordingly,

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s(s-1)} \\ \hline & b_1 & b_2 & \dots & b_{s-1} \quad b_s \\ \hline & \widehat{b}_1 & \widehat{b}_2 & \cdots & \widehat{b}_{s-1} \quad \widehat{b}_s \end{array}$$

Proper embedded methods combine a pair of RK schemes of *different* order $q, \widehat{q}$ so that $\|V - \widehat{V}\|$ indicates if the step size h is small enough, at each step k of the overall iteration. Since the vectors U^i, $\widetilde{U}^i$, $i \in [s]$ are used twice, this comes at little additional costs. We also point out that unlike the linear case, the magnitude $\|V\|$ of tangent vectors has much less influence, because the scheme (3.17) that is consecutively applied at each step k, is based on (3.10*b*) with the initial condition $V(0) = 0$. As a consequence, the magnitude $\|V\|$ of the update (3.17*d*) will be relatively small at each step k.

We list the tableaus of two embedded methods that we evaluate in section 6.

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1 & 0 \\ & 1/2 & 1/2 \end{array}$$

(**RK-1/2**)

$$\begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 2/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \\ & 1/3 & 2/3 & 0 \end{array}$$

(**RK-3/2**)

The first method combines the forward Euler scheme and Heun's method of order 2. The second method complements Heun's method of order 3 as specified on [12, p 166]. We call **RKMK-1/2** and **RKMK-3/2** the geometric versions of these schemes when they are applied in connection with (3.17).

We conclude this section by specifying the extension of (3.17) in order to include adaptive step size control.

Algorithm 1. Embedded RKMK with adaptive step size control.

Data: Tolerance $0 < \tau \ll 1$,
Tolerance factor $n_\tau \in \mathbb{N}$,
sufficiently small initial step size $h = h_0$.
$k \leftarrow 1$
while termination criterion (6.2) is not satisfied **do**
  Compute $U^i, \widetilde{U}^i$, $i \in [s]$ by (3.17*b*) and (3.17*c*).
  Compute $V, \widehat{V}$ by (5.21).
  if $d_I(V, \widehat{V}) < \frac{\tau}{n_\tau}$ **then**
    $h \leftarrow 1.25h$ /* increase the step size */
    compute the update $W(h)$ by (3.17*e*).
    $W_0 \leftarrow W(h)$
    $k \leftarrow k + 1$
  else if $d_I(V, \widehat{V}) < \tau$ **then**
    /* keep the step size */
    compute the update $W(h)$ by (3.17*e*),
    $W_0 \leftarrow W(h)$
    $k \leftarrow k + 1$
  else
    $h \leftarrow \frac{h}{2}$ /* decrease the step size */
    /* do not increase k, but repeat iteration k */
end

The distance function d_I defined by (6.1). Typical parameter values are $\tau = 0.01, n_\tau = 20$. Starting with a small initial step size h_0, the algorithm adaptively generates a sequence (h_k) whose values increase whenever the local error estimate is much smaller (by a factor n_τ) than the prescribed tolerance τ.

6. Experiments and discussion

This section is organized as follows (see also figure 1). We specify details of our implementation in section 6.1. Section 6.2 reports the evaluation of the geometric RKMK schemes (3.17) with embedded step size control (algorithm 1), for integrating the *nonlinear* assignment flow. Section 6.3 is devoted to the *linear* assignment flow: assessment of how closely it approximates the nonlinear assignment flow, evaluation of the RK schemes (5.1) with adaptive step size selection (5.18), and evaluation of the exponential integrator introduced in section 4.2.

6.1. Implementation details

All algorithms were implemented and evaluated using Mathematica. We did *not* apply any assignment normalization as suggested by [1, section 3.3.1], since mathematica can work with arbitrary numerical precision. Our experiments thus illustrate *intrinsic* properties of the assignment flow the hold on the open assignment manifold $\mathcal{W}$, and the reliability and efficiency of a variety of algorithms for integrating this flow numerically while respecting the underlying geometry. We do not examine the flow *on* the boundary of $\overline{\mathcal{W}}$ which does not belong to the domain of our approach, by definition. For an investigation of a particular numerical scheme in this specific context, we refer to [26].

Throughout the experiments, we used *uniform* weights in (2.19) and (2.16), respectively, since how to choose these 'control variables' in a proper way depending on the application at hand, is subject of our current work. Yet, we point out that the algorithms of the present paper *do* cover such more general scenarios. For example, the simplest geometric RKMK scheme (3.17) was recently used for integrating the assignment flow in *unsupervised* scenarios [9], where labels evolve and hence distance vectors (2.17) no longer are static but vary with time $D = D(t)$, too.

6.1.1. Ground truth flows. In order to obtain a baseline for assessing the performance of linearizations, of approximate numerical integration by various schemes or both, we always solved the assignment flow (nonlinear or linear) with high numerical accuracy using the geometric implicit Euler scheme (nonlinear flow) or the euclidean implicit Euler scheme (linear flow), with a sufficiently small step size h. This requires to solve a fixed point equation as part of every iteration k, involving the nonlinear mapping on the right-hand side of (3.10*b*) in case of the nonlinear assignment flow, or the linear mapping on the right-hand side of (4.3*b*) in case of the linear assignment flow. These fixed point equations were iteratively solved as well, and the corresponding iterations terminated when subsequent elements of the corresponding subsequences $(V^{k_i})_{i\geqslant 0}$ that measure the residual of the fixed point equation, satisfied

$$d_I(V^{k_{i+1}}, V^{k_i}) = \frac{1}{|J|}\max_{i\in I}\|V_i^{k_{i+1}} - V_i^{k_i}\| \leqslant 10^{-8}. \tag{6.1}$$

Starting these inner iterative loops with $V^{k_0} = V^{(k)}$ and terminating with $V^{k_{i,\mathrm{end}}}$, we set $V^{(k+1)} = V^{k_{i,\mathrm{end}}}$ and continued with the outer iteration $k+1$.

6.1.2. Termination criterion. As suggested by [1], all iterative numerical schemes generating sequences $(W^{(k)})$ were terminated when the average entropy of the assignment vectors dropped below the threshold

$$-\frac{1}{|I||J|}\sum_{i\in I}\sum_{j\in J} W_{ij}^{(k)}\log W_{ij}^{(k)} < 10^{-3}. \tag{6.2}$$

If this happens, then—possibly up to a tiny subset of pixels $i \in I$—all assignment vectors W_i are very close to a unit vector and hence almost uniquely indicate a labeling.

6.1.3. Data. Besides using the one-dimensional signal shown by figure 7 that enables to visualize the entire evolution of the flow as plot in a single simplex (see figure 9), we used two further labeling scenarios for evaluating the numerical integration schemes.

(a) (b)

Figure 3. Labeling scenarios for evaluating numerical schemes for integrating the assignment flow. (a) Noisy computer-generated data (left) made from a ground truth with 31 labels (right). (b) A color image used as input data (left) using four color values as labels. These labels are illustrated on the right where each pixel has been replaced by the closest label.

Figure 3(a) shows a scenario adopted from [1, figure 6] using $\rho = 0.1$ and $|\mathcal{N}_i| = 7 \times 7$. The input data (right panel) comprise 31 labels encoded as vertices (unit vectors) of a corresponding simplex. This results in *uniform* distances (2.17) and enables to assess in an unbiased way the effect of regularization by geometric diffusion in terms of the similarity map (2.19).

Figure 3(b) shows a color image together with four color vectors used as labels, as illustrated by the panel on the right. In contrast to the data of figure 3(a) with a high level of noise and a uniform data term (as motivated and explained above), the input data shown on the left of figure 3(b) are not noisy but comprise spatial structures at quite different scales (fine texture, large homogeneous regions), causing a nonuniform data term and a more complex assignment flow.

Both scenarios together provide a testbed in order to check and compare schemes for numerically integrating the assignment flow.

6.2. Nonlinear flow: embedded RKMK-schemes

Figures 4 and 5 show the results of the two embedded RKMK schemes of section 5.2 used to integrate the full nonlinear assignment flow (2.20), for the data shown by left panels of figures 3(a) and (b).

The two embedded RKMK schemes combine RKMK schemes of different approximation order q/q', 1/2 and 3/2, respectively, which reuse vector field evaluations (3.17) in order to produce sequences of tangent vectors $(V^{(k)})$, $(\hat{V}^{(k)})$ that enable to estimate the local approximation error by monitoring the distances $d_I(V^{(k)}, \hat{V}^{(k)})$. As specified by algorithm 1, step sizes h_k adaptively increase provided a prescribed error tolerance is not violated.

The parameter values $\tau = 0.01$ (tolerance) and $n_\tau = 20$ (tolerance factor), used to produce the results shown by figures 4 and 5, suffice to integrate accurately the full nonlinear assignment flow by the respective *explicit* schemes, as the comparison with the ground truth labeling generated by the implicit geometric Euler scheme shows. Since RKMK-3/2 has higher order q than RKMK-1/2, larger step sizes can be tolerated. On the other hand, each iteration of RKMK-3/2 is about twice expensive as RKMK-1/2.

Both plots of the step size sequences (h_k) reveal that the initial step size $h_0 = 0.01$ was much too small (conservative), and that a fixed value of h_k is adequate for most of the iterations.

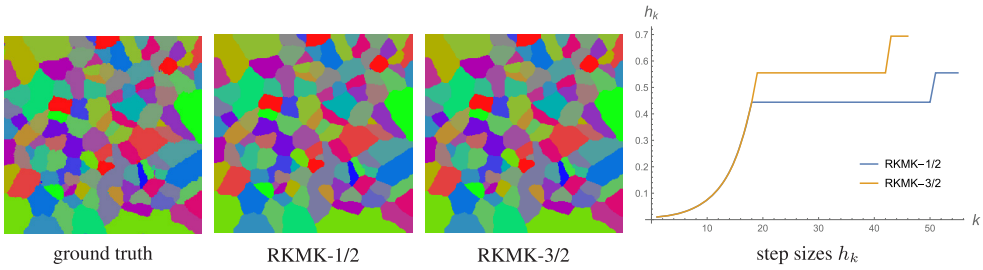


Figure 4. Nonlinear assignment flow, embedded RKMK schemes. Results of processing the data shown by figure 3(a), left panel (parameter: $\rho = 0.1, |\mathcal{N}_i| = 7 \times 7$). The ground truth labeling resulting from integrating the (full nonlinear) assignment flow using the *implicit* geometric Euler scheme (step size $h = 0.01$) is displayed in the left panel. The labeling results of these *explicit* schemes are shown in the middle. Because there is almost no difference to the ground truth result, both explicit schemes integrate the assignment flow sufficiently accurate. The corresponding sequences of adaptive step sizes (h_k) generated by the embedded geometric schemes RKMK-1/2 and RKMK-3/2 of section 5.2 are shown in the rightmost panel.

This value was larger for the experiment corresponding to figure 4 due to the uniform data term (by construction, as explained in section 6.1.3) and the more uniform scale of spatial structures. By contrast, the presence of spatial structures at quite different scales in the data corresponding to figure 5 causes a more involved assignment flow to be integrated, and hence to a smaller step size after adaption. Comparing the two rightmost panels of figure 5 shows that the strength of regularization (neighborhood size $|\mathcal{N}_i|$) had only little influence on the sequences of step sizes.

We never observed decreasing step sizes in these *supervised* scenarios, that is the corresponding step of algorithm 1 never was active. This may change in more involved scenarios, however (see remark 3.2).

Overall, a few dozens of explicit iterations suffice for accurate geometric numerical integration of the assignment flow with well below 1% wrongly labeled pixels (figure 6). Each iteration may be implemented in a fine-grained parallel way and has computational costs roughly equivalent to a convolution, besides mapping to the tangent space $\mathcal{T}_0$ and back to the assignment manifold $\mathcal{W}$, at each iteration.

6.3. Linear assignment flow

The approach of section 4 involves two different approximations:

(i) the *linear* assignment flow (4.2) approximating the *full* assignment flow (2.20), and
(ii) the numerical integration of the linear assignment flow using two alternative numerical schemes:

(a) adaptive RK schemes (section 5.1) based on the parametrization of proposition 4.2 and
(b) the exponential integrator (section 4.2).

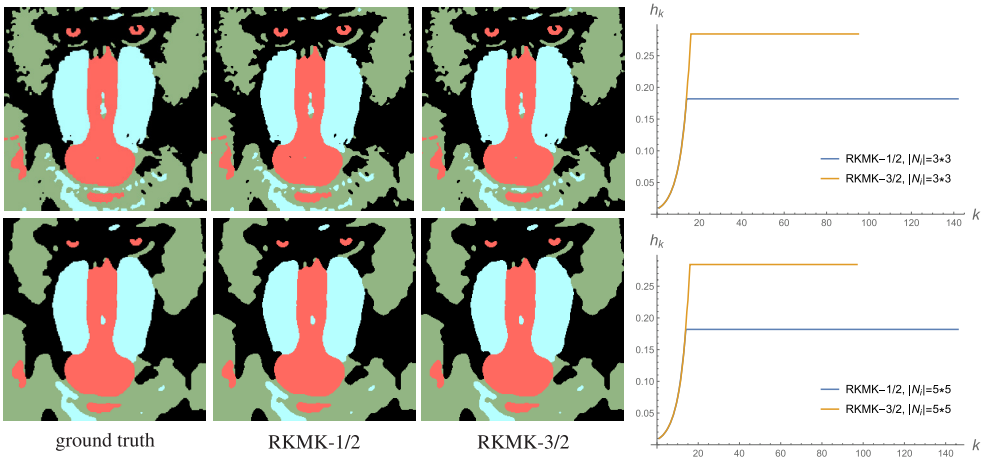


Figure 5. Nonlinear assignment flow, embedded RKMK schemes. Results of processing the data shown by figure 3(b), left panel (parameters: $\rho = 0.5$, $|\mathcal{N}_i| = 3 \times 3$ (less regularization; top row) and $|\mathcal{N}_i| = 5 \times 5$ (more regularization; bottom row). The ground truth labeling was computed by integrating the (full nonlinear) assignment flow using the *implicit* geometric Euler scheme (step size $h = 0.01$). The embedded RKMK schemes (section 5.2) generated the adaptive step sizes shown in the panels on the right and almost identical labeling results.

Due to the remarkable approximation properties of the linear assignment flow when a *single* linearization at the barycenter is only used (section 6.3.1), we entirely focused on this flow when evaluating the numerical schemes (a) and (b) in sections 6.3.2 and 6.3.3.

6.3.1. Approximation property. We report a series of experiments for the 1D signal depicted by figure 7 using both the full and the linear assignment flow in order to check how closely the latter approximates the former. Then we discuss the linear assignment flow for the two 2D scenarios shown by figure 3.

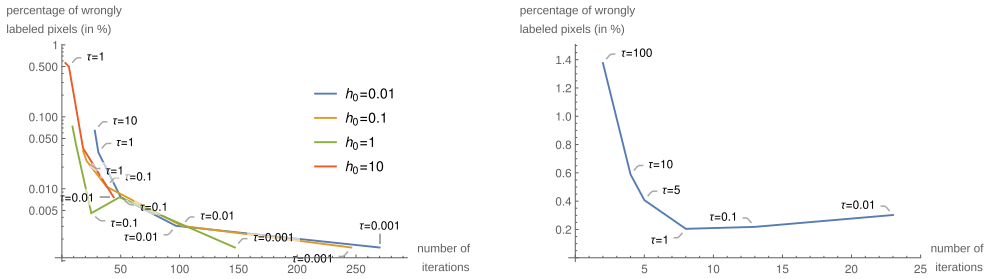
The parameter value $\rho = 0.1$ for scaling in (2.18) the data, for all 1D experiments discussed below. This gave a larger weight to the 'data term' so that—in view of the noisy data (figure 7)—the regularization property of the assignment flow (2.20), in terms of the similarity vectors $S_i(W)$ interacting through (2.19), was essential for labeling.

We first explain how the linearizations of the assignment flow were controlled. According to proposition 4.2, using the parametrization (4.3*a*) and the linear ODE (4.3*b*) is equivalent to the linear assignment flow (4.2). Using again the parametrization (4.3*a*) and repeating the proof of proposition 4.2 shows that the full assignment flow (2.20) is locally governed by the *nonlinear* ODE

$$\dot{V} = R_{W_0}\big(S(\exp_{W_0}(V))\big), \qquad V(0) = 0. \tag{6.3}$$

Taking into account (4.6) and subtracting the right-hand side of the approximation (4.3*b*) from the above right-hand side gives

$$R_{W_0}\big(S(\exp_{W_0}(V))\big) - R_{W_0}(s_0 + S_0 V) = R_{W_0}\big(S(\exp_{W_0}(V)) - S(W_0) - \mathrm{d}S_{W_0}(V)\big), \tag{6.4}$$



nonlinear assignment flow linear assignment flow

Figure 6. Number of iterations versus wrongly labeled pixels. We relate the number of iterations to the percentage of wrongly labeled pixels (compared to the ground truth) for the nonlinear assignment flow (integrated by RKMK-3/2 with tolerance τ and initial step size h_0 as in algorithm 1) and the linear assignment flow (integrated by an adaptive RK scheme of order 4, with τ defined by (5.18)) for the mandrill experiment. In general, parameter choices which increase the number of iterations, yield a better labeling. Thus the parameters of the (non-)linear assignment flow can be chosen to achieve an application-specific compromise between runtime and accuracy.

which shows that this approximation deteriorates with increasingly large tangent vectors V. As a consequence, we first solved the linear flow (4.2) using (4.3) *without* updating the point of linearization $W_0 = \mathbb{1}_{\mathcal{W}}$ and fixed after termination at k_{end} (=number of required outer iterations) the constant

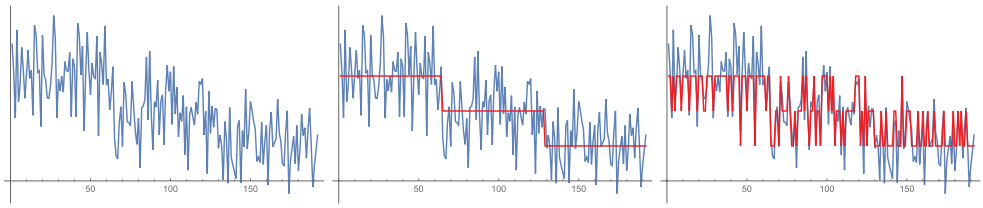
$$\|V\|_{\max} = \max_{i \in I} \|V_i^{(k_{\text{end}})}\|. \tag{6.5}$$

Then we solved the linear assignment flow again and updated the linearization point W_0 in view of (6.4) whenever

$$\max_{i \in I} \|V_i^{(k)}\| > \frac{\|V\|_{\max}}{c}, \qquad c \geqslant 1, \tag{6.6}$$

using the parameter c to control the number of linearizations: a single linearization and no linearization update if $c = 1$ and an increasing number of updates for larger values of c. We updated components of the linearization point $W_{0,i}$ by $W_i(h)$, $i \in I$ only when $\min_{j \in J} W_{ij}(h) > 0.01$, in order to keep linearization points inside the simplex, in view of the entries (4.7*d*) of $\mathrm{d}S_{W_0}$ normalized by components of $W_{0,k}$.

After termination, the induced labelings were compared to those of the full assignment flow, and the number of wrongly assigned labels was taken as a quantitative measure for the approximation property of the linear assignment flow: Except for the minimal neighborhood size $|\mathcal{N}_i| = 3$, a single linearization almost suffices to obtain a correct labeling. Overall, the maximal number of 3 labeling errors (out of 192) is very small, and these errors merely correspond to shifts by a single pixel position of the signal transition in the case $|\mathcal{N}_i| = 9$ (see figure 8). We conclude that for supervised labeling, the linear assignment flow (4.2) (which is nonlinear(!)—see remark 4.3) indeed captures a major part of nonlinearity of the full assignment flow (2.20). Figure 9 illustrates the similarity of the two flows (and the dissimilarity



(a) (b) (c)

Figure 7. (a) A noisy 1D signal used for the experiments of section 6.3.1. (b) The piecewise constant signal (red) used to generate (a) by superimposing noise. (c) *Local* rounding to the next label and comparing to (b) indicates the noise level. Local rounding is equivalent to omitting regularization in the assignment flow by replacing the interacting similarity vectors $S(W)$ in (2.20) by the non-interacting likelihood vectors $L(W)$ of (2.18).

in the case $|\mathcal{N}_i| = 3$) in terms of all $|I| = 192$ sequences $(W_i^{(k)})$, $i \in |I|$, plotted as piecewise linear trajectories.

We cannot assure, however, that this approximation property persists in more general cases (see remark 3.2) whose study is beyond the scope of the present paper.

We now turn to the scenarios shown by figure 3. Figure 10 shows the results obtained using the implicit Euler scheme and the *same* parameter settings that were used to integrate the nonlinear flow, to obtain the ground truth flows and results depicted by the leftmost panels of figures 4 and 5. Comparing the labelings returned by the linear and nonlinear assignment flow, respectively, confirms the discussion of the 1D experiments detailed above: the results agree except for a very small subset of pixels close to signal transitions which are immaterial for subsequent image interpretation.

The results shown by figure 10 served as ground truth for studying the *explicit* numerical schemes of sections 6.3.2 and 6.3.3 for integrating the linear assignment flow.

6.3.2. Adaptive RK schemes. We evaluated the adaptive RK schemes (FE) of order $q = 1$ and (RK4) of order $q = 4$, due to (5.1), supposed to integrate the linear ODE (4.3*b*), after rearranging the polynomials of (5.1) in Horner form.

Figures 11 and 12 show the results for the *linear* assignment flow based on a single linearization at the barycenter, using the results shown by figure 10 as ground truth. The step sizes h_k were computed at each iteration k using the local error estimate (5.7) such that $\frac{1}{|I|^{1/2}}\|V(t_{k+1}) - V^{(k+1)}\| \leqslant \tau = 0.01$, that is on average $\|V_i(t_{k+1}) - V_i^{(k+1)}\| \leqslant \tau$ for all pixels $i \in I$. The spectral norm $\|A\|$ was computed beforehand using the basic power iteration.

As explained above when the criterion (5.18) was introduced, step sizes must decrease due to the increasing norms $\|V^{(k)}\|$, in order to keep the local integration error bounded. Furthermore, in agreement with figure 2, raising the order q of the integration scheme leads to significantly larger step sizes and hence to smaller total numbers of iterations, at the cost of more expensive iterative steps. Yet, roughly taking these additional costs into account by multiplying the total iteration numbers for $q = 4$ by 4, indicates that raising the order q reduces the overall costs. In this respect our findings for the linear assignment flow differ from the corresponding findings for the nonlinear assignment flow and the embedded RKMK schemes, discussed in section 6.2.

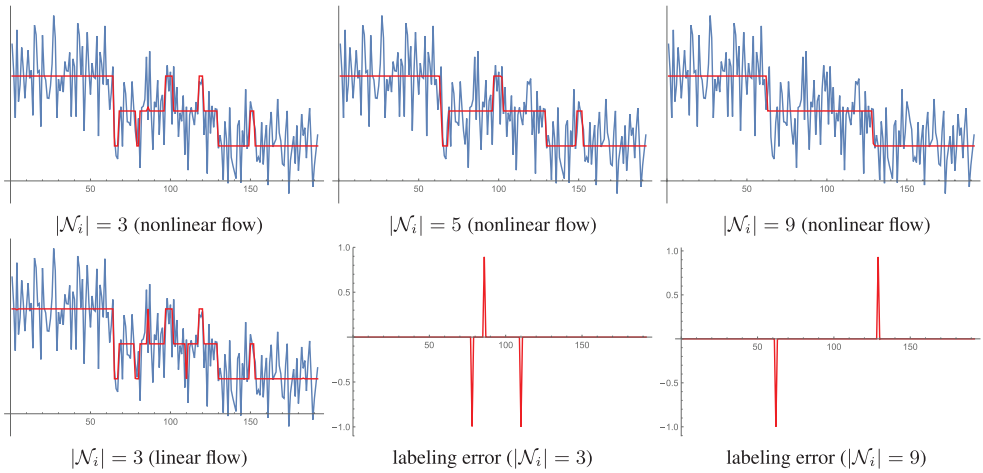


Figure 8. Nonlinear versus linear assignment flow. TOP ROW: labelings determined by the assignment flow as a reference for the linear assignment flow, using different neighborhood sizes $|\mathcal{N}_i|$. BOTTOM ROW: labeling determined by the linear assignment flow that differs in 3 pixels from the corresponding above result of the full assignment flow. These errors and the two errors in the case $|\mathcal{N}_i| = 9$ are shown in the center and right panel. These and further results listed as table 1 show that the linear assignment flow achieves high-quality labelings.

Table 1. Approximation property of the linear assignment flow. The entries x, y specify the number x of linearizations and the number y of wrongly assigned labels (out of 192 assigned labels), depending on the neighborhood size $|\mathcal{N}_i|$ (strength of regularization) and the parameter c specifying the tangent space threshold (6.6).

		c				
		1	2	3	4	5
$\lvert\mathcal{N}_i\rvert$	3	1, 3	4, 3	7, 3	10, 1	13, 0
	5	1, 0				
	9	1, 2	5, 0			

6.3.3. Exponential integrator. For integrating the linearized assignment flow with exponential integrators, we consider equation (4.19) and the Krylov space approximation (4.23)

$$V(T) = T\varphi_1\big(TA\big)a \approx T\|a\|\mathcal{V}_m\varphi_1(T\mathcal{H}_m)e_1. \tag{6.7}$$

As the evaluation of $\varphi_1(T\mathcal{H}_m)e_1$ is explained in section 4.2, we only discuss here the choice of the parameters m and T.

The dimension m of the Krylov subspace controls the quality of the approximation, where larger values theoretically lead to a better approximation. In our experiments, rather small numbers, like $m = 5$, turned out to suffice to produce labelings very close to the ground truth labelings, that were generated by the implicit Euler method—see figures 13 and 14. As the runtime of the algorithm increases with growing m, this parameter should not be chosen too large.

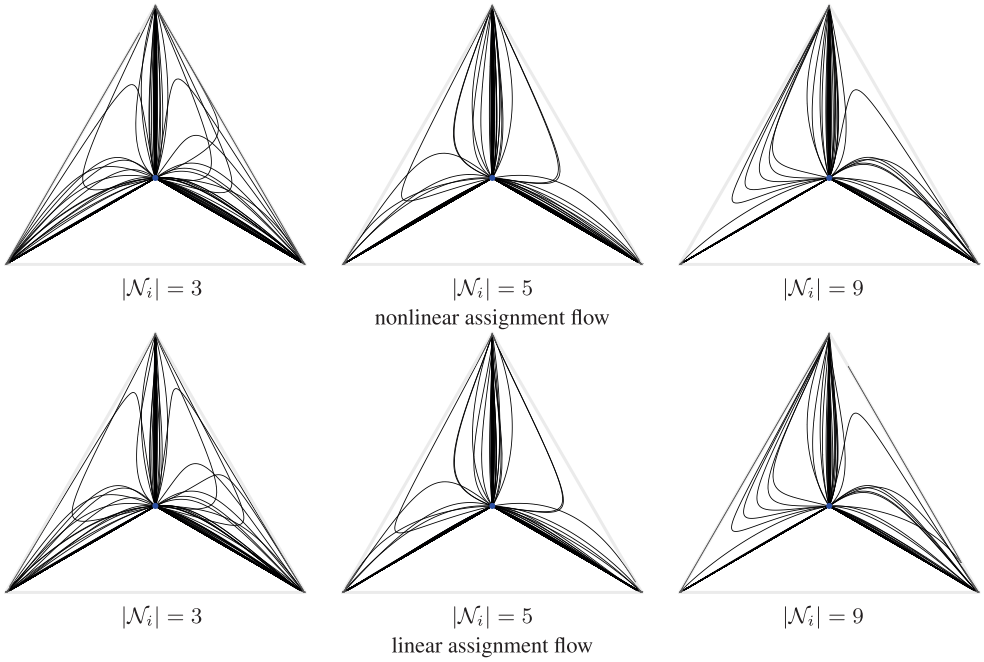


Figure 9. Nonlinear versus linear assignment flow. Comparison of the assignment flow (2.20) (top row) and the linear assignment flow (4.2) (bottom row) in terms of all $|I|$ solution curves $W_i(t)$, $i \in I$, plotted in the 3-label simplex. A major part of the trajectories approaches more or less directly a vertex, whereas another part changes the original direction due to regularization by geometric smoothing. Except for the cases with the minimal neighborhood $|\mathcal{N}_i| = 3$, the similarity of both flows is apparent. This illustrates the reason for very small observed numbers of labeling errors, as listed and depicted by table 1 and figure 8.

Figure 10. Linear assignment flow. Labeling results for the two scenarios of figure 3, using the *linear* assignment flow with a single linearization at the barycenter and the *implicit* Euler scheme for numerical integration. Comparison with the labeling results of the *nonlinear* assignment flow (figure 4(a), figure 5 'ground truth') demonstrates a remarkable approximation property of the linear assignment flow.

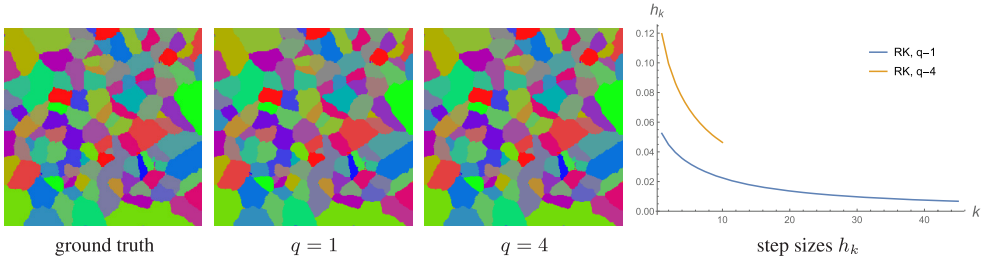


Figure 11. Linear assignment flow, adaptive RK-schemes. Results of the linear assignment flow (4.2) based on the parametrization (4.3), the RK schemes (5.1) of order $q = 1$ (FE) and $q = 4$ (RK4), and adaptive step size selection based on the local error estimate (5.7). The labeling results for $q = 1$, $q = 4$ are almost identical to ground truth from figure 10.

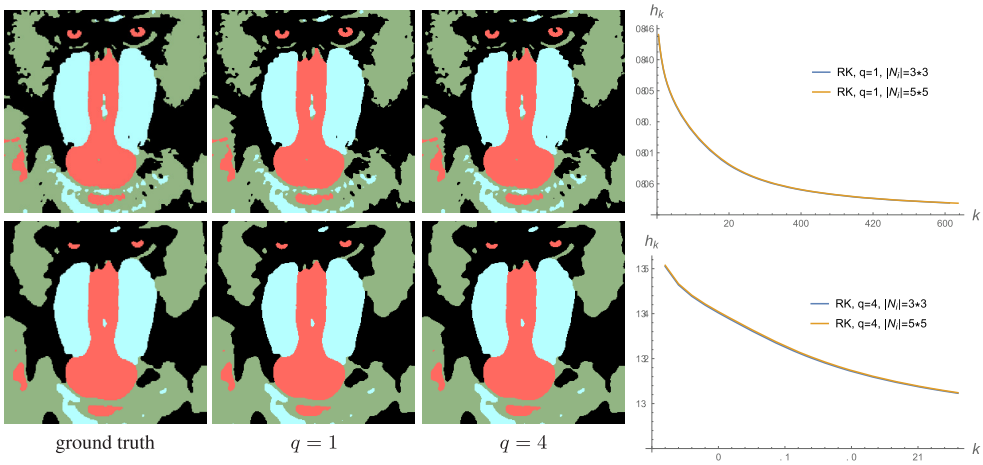
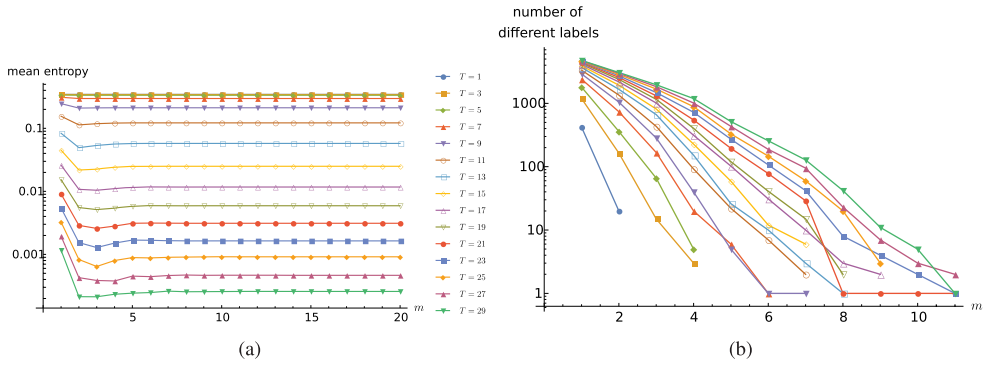


Figure 12. Linear assignment flow, adaptive RK-schemes. Results of the linear assignment flow (4.2) for $|\mathcal{N}_i| = 3 \times 3$ (top row) and $|\mathcal{N}_i| = 5 \times 5$ (bottom row) based on the parametrization (4.3), the RK schemes (5.1) of order $q = 1$ (FE, second column) and $q = 4$ (RK4, third column). The labeling results for $q = 1$, $q = 4$ are almost identical to the ground truth from figure 10. The rightmost column shows the adaptive step size selection based on the local error estimate (5.7). For fixed q, increasing the neighborhood size $|\mathcal{N}_i|$ has almost no effect: the step size sequences agree up to the second digit.

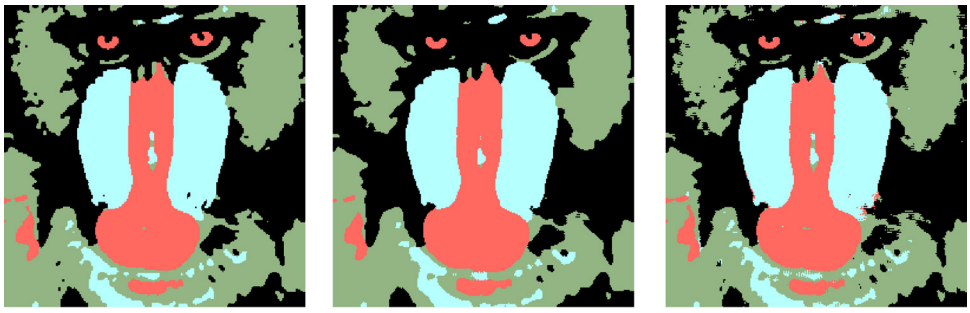
Scaling A and a in (4.18) affects the vector field defining the linear ODE (4.3*b*). Hence, fixing any time point T depends on this scaling factor, too. As a consequence, since A and a depend on the problem data (4.18), the choice of T is problem dependent. On the other hand, the discussion following the proof of theorem 5.2 showed that $\|V(t)\|$ increases with t, and T merely has to chosen large enough such that $W(T)$ defined by (4.3*a*) satisfies the termination criterion (6.2)—see (5.20*c*) for a rough estimate. Choosing T overly large will cause numerical underflow and overflow issues, however.

Almost all runtime is consumed by the Arnoldi iteration producing the subspace basis $\mathcal{V}_m$. Due to the small dimension m, the total runtime is very short, and time required for the subsequent evaluation of the right-hand side of (6.7) is negligible.



(a) (b)

Figure 13. Linear assignment flow, exponential integrators, influence of the Krylov subspace dimension m and the point of evaluation T. The results correspond to the labeling problem shown by figure 14. They demonstrate that m can be chosen quite small. Figure (a) shows that the mean entropy decreases with increasing T but does not decrease for *fixed* T and $m > 3$. Figure (b) shows for $T = 1, \ldots, 29$ (same color code as (a)) and for each m the number of labels that change when the dimension of the Krylov subspace is increased to $m + 1$. All curves decrease with increasing m, and curves that reach 0 label changes just discontinue due to the logarithmic scale. The plot shows that independent of T (i.e. for any T) there is an m such that increasing the dimension of the Krylov space does not change the labeling at all.



(a) (b) (c)

Figure 14. Linear assignment flow, exponential integrators, sample labelings. Figure (a) shows the ground truth labeling as generated by the implicit Euler method. Figure (b) displays the labeling generated by the exponential integrator using the Krylov subspace dimension $m = 5$, which is very close to the ground truth labeling. As demonstrated by figure 13, m cannot be chosen too small, however, since labelings then start to deteriorate rapidly. This is illustrated by figure (c) which shows the labeling for $m = 3$. Comparison with (b) shows that dimensions 4 and 5 'contain' small-scale details of the correct labeling induced by the linear assignment flow.

7. Conclusion

We investigated numerical methods for image labeling by integrating the large system of nonlinear ODEs defining the assignment flow (2.20), which evolves on the assignment manifold. All methods exactly respect the underlying geometry. Specifically, we adapted RKMK methods and showed that *embedded* RKMK-methods work very well for automatically adjusting the step size, at negligible additional costs. Raising the order enables leads to larger step sizes, which is compensated by the higher computational costs per iteration, however. In either case,

each iteration only involves convolution like operations over local neighborhoods together with pixelwise nonlinear functions evaluations.

We derived and introduced the *linear* assignment flow, a nonlinear approximation of the (full) assignment flow that is governed by a large linear system of ODEs on the tangent space. Experiments showed that the approximation is remarkably close, as measured by the number of different label assignments.

We investigated two further families of numerical schemes for integrating the linear assignment flow: established RK schemes with adaptive step size selection based on a local integration error estimate, and exponential integrators for approximately evaluating Duhamel's integral using a Krylov subspace. In the former case, higher-order schemes really pay, unlike for the RKMK schemes and the full assignment flow, as mentioned above. Choosing the classical RK scheme with $q = 4$, for example, few dozens of iterations suffice to reach the termination criterion, with high potential for parallel implementation. The exponential integrators, on the other hand, directly approximate the integral determining $V(T)$ and in this sense are non-iterative. Here, a Krylov subspace basis of low dimension merely has to be computed, using a standard iterative method. Even though this method is differs mathematically from the RK schemes, it has potential for real-time implementation as well.

All methods provide a sound basis for more advanced image analysis tasks that involve labeling by evaluating the assignment flow as a subroutine. Accordingly, our future work concerns an extension of the unsupervised labeling approach [9], where label dictionaries are directly learned from data through label assignment. Furthermore, methods under investigation for learning to adapt regularization parameters of the assignment flow to specific image classes, require advanced discretization and numerical methods based on the results reported in the present paper.

Acknowledgments

This work was supported by the German Research Foundation (DFG), Grant No. GRK 1653.

ORCID iDs

Alexander Zeilmann https://orcid.org/0000-0002-8119-0349
Fabrizio Savarino https://orcid.org/0000-0001-9629-8486
Stefania Petra https://orcid.org/0000-0002-7189-2275
Christoph Schnörr https://orcid.org/0000-0002-8999-2338

References

[1] Åström F, Petra S, Schmitzer B and Schnörr C 2017 Image labeling by assignment *J. Math. Imaging Vis.* **58** 211–38
[2] Amari S-I and Nagaoka H 2000 *Methods of Information Geometry* (Oxford: Oxford University Press) (https://bookstore.ams.org/mmono-191/)
[3] Ay N, Jost J, Lê H V and Schwachhöfer L 2017 *Information Geometry* (Berlin: Springer) (https://doi.org/10.1007/978-3-319-56478-4)
[4] Kappes J H *et al* 2015 A comparative study of modern inference techniques for structured discrete energy minimization problems *Int. J. Comput. Vis.* **115** 155–84

[5] Weinan E 2017 A proposal on machine learning via dynamical systems *Commun. Math. Stat.* **5** 1–11
[6] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. CVPR* (https://doi.org/10.1109/CVPR.2016.90)
[7] Haber E and Ruthotto L 2017 Stable architectures for deep neural networks *Inverse Problems* **34** 014004
[8] Hühnerbein R, Savarino F, Åström F and Schnörr C 2018 Image labeling based on graphical models using wasserstein messages and geometric assignment *SIAM J. Imaging Sci.* **11** 1317–62
[9] Zern A, Zisler M, Åström F, Petra S and Schnörr C 2018 Unsupervised label learning on manifolds by spatially regularized geometric assignment *Proc. GCPR* (https://doi.org/10.1007/978-3-030-12939-2_48)
[10] Iserles A, Munthe-Kaas H Z, Nørsett S P and Zanna A 2005 Lie-group methods *Acta Numer.* **14** 1–148
[11] Munthe-Kaas H 1999 High order Runge–Kutta methods on manifolds *Appl. Numer. Math.* **29** 115–27
[12] Hairer E, Nørsett S P and Wanner G 2008 *Solving Ordinary Differential Equations* I 3rd edn (Berlin: Springer) (https://doi.org/10.1007/978-3-540-78862-1)
[13] Saad Y 1992 Analysis of some Krylov subspace approximations to the matrix exponential operator *SIAM J. Numer. Anal.* **29** 209–28
[14] Hochbruck M and Lubich C 1997 On Krylov subspace approximations to the matrix exponential operator *SIAM J. Numer. Anal.* **34** 1911–25
[15] Hochbruck M and Ostermann A 2010 *Exponential Integrators Acta Numer.* **19** 209–86
[16] Hairer E, Lubich C and Wanner G 2006 *Geometric Numerical Integration* (Berlin: Springer) (https://doi.org/10.1007/3-540-30666-8)
[17] Ay N and Erb I 2005 On a notion of linear replicator equations *J. Dyn. Differ. Equ.* **17** 427–51
[18] Teschl G 2012 *Ordinary Differential Equations and Dynamical Systems* (*Graduate Studies in Mathematics* vol 140) (Providence, RI: American Mathematical Society) (https://doi.org/10.1090/gsm/140)
[19] Higham N J 2008 *Functions of Matrices: Theory and Computation* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9780898717778)
[20] Moler C and Van Loan C 2003 Nineteen Dubious ways to compute the exponential of a matrix, twenty-five years later *SIAM Rev.* **45** 3–49
[21] Niesen J and Wright W M 2012 Algorithm 919: a Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators *ACM Trans. Math. Softw.* **38** 22
[22] Al-Mohy A and Higham N 2011 Computing the action of the matrix exponential, with an application to exponential integrators *SIAM J. Sci. Comput.* **33** 488–511
[23] Sidje R B 1998 Expokit: a software package for computing matrix exponentials *ACM Trans. Math. Softw.* **24** 130–56
[24] Olver F W J, Olde Daalhuis A B, Lozier D W, Schneider B I, Boisvert R F, Clark C W, Miller B R and Saunders B V (ed) *NIST Digital Library of Mathematical Functions* (http://dlmf.nist.gov/), Release 1.0.22 of 2019-03-15
[25] Alzer H 1997 On some inequalities for the incomplete gamma function *Math. Comput.* **218** 771–8
[26] Bergmann R, Fitschen J H, Persch J and Steidl G 2017 Iterative multiplicative filters for data labeling *Int. J. Comput. Vis.* **123** 435–53