CONTRACTIVE AND TRANSITIONAL DISCRETIZATION OF GRADIENT FLOWS

MOODY T. CHU* AND LI-ZHI LIAO†

Abstract. Gradient adaption is ubiquitous in nature. The notion has also been employed over a wide range of applications. It is often the case that each application has been given consideration separately for its best efficiency. Understanding the mathematical foundations of effective algorithms for general gradient dynamics should be of practical importance across diverse fields. Contractivity and transitionality are two such characteristics investigated in this paper. While standard, general-purpose numerical integrators are readily applicable, they are sophisticatedly designed for the purpose of closely tracking exact integral curves, which is slow, expensive, and unnecessary for gradient flows. In contrast, this work discusses some nontraditional ways of discretization that exploit the gradient structure, namely, they allow the possibility to relax the precision, hence the speedup; the ability to guide the path, hence the robustness; and the contractibility to correct the error, hence the precision. These high-order iterative methods, capable of transforming themselves into fast converging Newton-like methods when closing in a desired equilibrium point, can be combined with suitable starting procedure in the initial phase to further improve both speed and precision.

AMS subject classifications. 15A18, 15A51, 15A69, 60J99

Key words. gradient flows, dissipative vector fields, contractive methods, inexact Newton methods, transitional methods, hybrid methods

1. Introduction. Given a differentiable scalar function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, the basic fact that its gradient vector $\nabla f(\mathbf{x})$ points in the steepest ascent direction with the maximum rate of change $\|\nabla f(\mathbf{x})\|_2$ for the function value at the point $\mathbf{x}$ is well known. Generalizing to a differentiable functional f over a Hilbert space, the gradient should be interpreted as the Riesz representation of the Fréchet derivative of the functional f, whose true denotation depends on, of course, the underlying inner product. It is quite natural to expect that a system adapts itself continuously in the gradient direction. Simply put, a dynamical system in the form

$$\frac{d\mathbf{x}}{dt} = -\nabla f(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{1.1}$$

where f is a second-order differentiable functional over an appropriately defined Hilbert space H, is referred to as a (negative) gradient flow. The gradient dynamics was recognized and viewed challenging by Courant and others in earlier days [8, Section 1.1]. Since then, it has long attracted research attention across both fields of pure and applied mathematics. Far from being complete, we mention a few representative works on the general theory [43, 47, 61, 75, 87, 92, 94, 99].

Gradient adaption plays a significant role in nature and in applications. Thermal conduction along the negative temperature gradient of the isothermal surfaces and osmosis down the concentration gradient across the cell membrane typify the gradient adaption. Mathematical models for the phase separation of materials in iron alloys [18, 77], the segmentation or edge detection in image processing or computer vision [67, 90], the surface evolution in differential geometry [12, 28, 86], the flow of an ideal gas in porous medium [98], and the ground state in quantum systems [5] all employ this notion of gradient dynamics. Numerous other systems that evolve in time can also be interpreted in this way. See, for example, applications in the game theory [31, 78], probability [33], Markov chain [64], economics [32, 36], financial markets [32], quantum field theory [25], network communication [81], circuit theory [88], mechanism design [84], low rank approximation [59, 62], and the general surveys discussed in [42, 45, 47]. We give four specific examples below to motivate the importance of gradient dynamics, followed by an outline of possible approaches to a gradient dynamical system.

[*] Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA. (chu@math.ncsu.edu.) This research was supported in part by the National Science Foundation under grant DMS-1316779.

[†] Department of Mathematics Hong Kong Baptist University Kowloon Tong, Hong Kong. (liliao@hkbu.edu.hk) This research was supported in part by the Hong Kong General Research Fund and FRG of Hong Kong Baptist University.

1.1. Examples of gradient flows. To solve each of the following four problems in the most efficacious way probably requires specific and individually tailored method. However, together they demonstrate the wide scope of applications of gradient dynamics. Along the outline, we also raise a few curious questions.

Example 1. Consider a heat equation of the form

$$\frac{\partial u}{\partial t} = \nabla^2 u + p(u), \tag{1.2}$$

for $u = u(\mathbf{v}, t) \in L^2(U)$ with the boundary conditions $u|_{\partial U} = 0$. Then it is not difficult to see that the right-hand side is precisely the negative gradient of the functional

$$f(u) := \int_U \left(\frac{1}{2} \|\nabla u\|^2 - P(u) \right) \, d\mathbf{v},$$

where P is such that $P' = p$ and guarantees the existence of a minimizer for f.

Indeed, it is known that "a surprisingly large number of well known diffusive partial differential equations have the structure of a gradient flow" [76] with respect to some appropriately chosen energies and dissipative mechanisms [68]. The dissipative mechanism can be quite general even if an inner product is not explicitly known. Under the Wasserstein metric, for example, the Fokker-Planck equation is a Wasserstein gradient flow of a specific functional [58, 76]. Mean curvature flow, as another example, is the gradient flow for the area of hypersurfaces [22]. We understand that a PDE-based gradient dynamics can be approximated numerically by an ODE system of the form (1.1) through proper discretization in the space variable. The techniques proposed in this paper therefore should be applicable to PDE-based gradient dynamics as well.

Example 2. A major theme advocated and extensively demonstrated in the two books [94, 95] is that "gradient dynamical systems cover many iterative formulas of numerical analysis". We mention from our own experiences [19, 20] the Toda lattice

$$\frac{dX}{dt} = [X, \Pi_0(X)] \tag{1.3}$$

for symmetric tridiagonal matrices $X(t) \in \mathbb{R}^{n \times n}$, where $\Pi_0(X) := X^- - X^{-\top}$ with X^- denoting the strictly lower triangular portion of X and $[\cdot, \cdot]$ stands for the Lie bracket. On one hand, it is known that $X(t)$ is a Hamiltonian flow maintaining the spectrum of $X(0)$ for all t. On the other hand, it is also known that the dynamical system (1.3) is precisely the (projected) gradient flow for the objective functional [11]

$$f(Q) := \frac{1}{2} \|Q^\top X_0 Q - N\|_F^2, \tag{1.4}$$

subject to the constraint $Q \in \mathcal{O}(n)$, where $\mathcal{O}(n)$ is the orthogonal group and $N = \operatorname{diag}\{n, n-1, \ldots, 2, 1\}$. That is, $X(t)$ is a gradient flow moving to reduce the off-diagonal entries of $X(t) = Q(t)^\top X_0 Q(t)$ while aligning its diagonal entries in the same ordering as those in N.

Most interestingly, the sequence $\{X(k)\}$ obtained by sampling the solution flow $X(t)$ at integer times corresponds to exactly the sequence generated by the QR algorithm for eigenvalue computation [23, 93], attesting that the ever-important QR algorithm is a gradient adaption. Though the popular shifted QR algorithm is already fast, is it possible to develop a new iterative scheme that moves along the "gradient trajectory" of (1.3) to find its equilibrium point with significantly larger step sizes[1] and, hence, the speed?

Example 3. In the same vein, many important numerical algorithms can also be interpreted via systems and control theory as descent flows [6, 29, 37, 45, 49, 85, 92]. Consider the basic model

$$\frac{d\mathbf{x}(t)}{dt} = \phi(\mathbf{x}, \mathbf{r}), \tag{1.5}$$

[1] The Toda lattice (1.3) is an ODE system constrained to the isospectral manifold. Any reasonable discretization of the solution $X(t)$ should respect this manifold constraint which is eigenvalue preserving. Geometric integration techniques are particularly suitable for isospectral flows characterized by a Lie struture. See [16, 52, 53, 54, 55, 72], and the book [39]. However, so far as we know, these Lie-structure preserving methods have not exploited the gradient structure yet [57].

where the state variable $\mathbf{x}(t)$ is controlled by some properly selected function $\phi(\mathbf{x},\mathbf{r})$ which depends on a certain output feedback $\mathbf{r}$. Different choices of ϕ can be used to manipulate the flow $\mathbf{x}(t)$ and, hence, lead to various algorithms. It is convenient that the choice of the control strategy ϕ depends on optimizing a certain cost function $V(\mathbf{x}(t))$ which, in turn, plays the role as a Lyapunov function for the dynamical system. In this case, $\phi(\mathbf{x},\mathbf{r})$ is not necessarily the gradient of $V(\mathbf{x})$, but still points to a descent direction [91].

Take the classical problem of finding the root(s) of a given differentiable function $\mathbf{g}:\mathbb{R}^n\rightarrow\mathbb{R}^n$ as an example. A reasonable choice of the feedback for monitoring the progress made by any preferred algorithm is the residue function

$$\mathbf{r}(t) := -\mathbf{g}(\mathbf{x}(t)). \tag{1.6}$$

Following [9] and [20], we summarize in Table 1.1 a few possible choices for the control $\phi(\mathbf{x},\mathbf{r})$.

$\phi(\mathbf{x},\mathbf{r})$	$\frac{dV}{dt}$	$\frac{d\mathbf{x}}{dt}$
$\mathbf{g}'(\mathbf{x})^\top\mathbf{r}$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathbf{r}\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^\top\mathbf{g}(\mathbf{x})$
$\mathbf{g}'(\mathbf{x})^{-1}\mathbf{r}$	$-\Vert\mathbf{r}\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^{-1}\mathbf{g}(\mathbf{x})$
$\mathbf{g}'(\mathbf{x})^{-1}\mathrm{sgn}(\mathbf{r})$	$-\Vert\mathbf{r}\Vert_1$	$-\mathbf{g}'(\mathbf{x})^{-1}\mathrm{sgn}(\mathbf{g}(\mathbf{x}))$
$\mathrm{sgn}(\mathbf{g}'(\mathbf{x})^\top\mathbf{r})$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathbf{r}\Vert_1$	$-\mathrm{sgn}(\mathbf{g}'(\mathbf{x})^\top\mathbf{g}(\mathbf{r}))$
$\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{r})$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{r})\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{g}(\mathbf{x}))$

TABLE 1.1
Control strategies and the associated dynamical systems

It is not difficult to verify that $V(\mathbf{x})=\frac{1}{2}\Vert\mathbf{g}(\mathbf{x})\Vert_2^2$ and $V(\mathbf{x})=\Vert\mathbf{g}(\mathbf{x})\Vert_1$ can be used, respectively, as the cost functions in the first four cases and in the last case. The first case corresponds exactly to the gradient flow (1.1) with respect to $V(\mathbf{x})$. The second case is the well-known continuous Newton method [48, 89] which is not the gradient flow with respect to the Euclidean norm. However, because $\langle\mathbf{g}'(\mathbf{x})^{-1}\mathbf{r},\mathbf{g}'(\mathbf{x})^\top\mathbf{r}\rangle=\langle\mathbf{r},\mathbf{r}\rangle\geq 0$, the Newton direction forms an acute angle with the steepest descent direction. So the continuous Newton flow is still a descent flow. Assuming that $\mathbf{g}'(\mathbf{x})$ remains nonsingular, then the flow stops only when $\mathbf{r}(\mathbf{x})=0$. Similar decent properties hold for other cases, as is seen in the second column about $\frac{dV}{dt}$ above. A descent flow such as (1.5) for a suitable cost function $V(\mathbf{x})$ therefor generalizes the notion of the gradient flow. *Is it possible to integrate (1.5) more effectively than just the conventional continuation methods* [48, 80]?

Example 4. In the setting of a Kreĭn space [13, 27, 35], we can even consider a flow of this form

$$\left\{\begin{array}{rcl}\frac{d\mathbf{x}}{dt} & = & -\nabla_{\mathbf{x}}u(\mathbf{x},\mathbf{y}),\\ \frac{d\mathbf{y}}{dt} & = & \nabla_{\mathbf{y}}u(\mathbf{x},\mathbf{y}),\end{array}\right. \tag{1.7}$$

where $u:\mathbb{R}^n\times\mathbb{R}^m\rightarrow\mathbb{R}$ is sufficiently smooth and $\nabla_{\mathbf{x}}$ represents the partial gradient with respect to the variable $\mathbf{x}$ only, as a gradient flow [3, 10, 30]. What happens is that the inner product is now interpreted under different "metric". While the flow (1.1) can be used to find stable equilibria of a system, the flow (1.7) can be used to detect rare but perceptible transition events between long lived metastable states in a complex reaction system. The presence of transition events, namely, saddle points, and the knowledge of their location in the configuration space provide critical information for important systems such as phase transitions in nucleation, conformational changes in macromolecules, and transition states in chemical reactions [44]. The saddle-point flow (1.7) also arises naturally when solving a complex-valued differential system [7, 51]

$$\frac{d\mathbf{z}}{dt} = -g'(\mathbf{z}), \tag{1.8}$$

where $g:\mathbb{C}^n\rightarrow\mathbb{C}$ is analytic. If we identify $\mathbf{z}=\mathbf{x}+\imath\mathbf{y}$ with $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$ and $g(\mathbf{z})=u(\mathbf{x},\mathbf{y})+\imath v(\mathbf{x},\mathbf{y})$, then $g'(\mathbf{z})$ must satisfy the Cauchy-Riemann equations which translate exactly into (1.7).

In many ways, the saddle-point flows share properties similar to the gradient flows that we are about to study [21]. In particular, the techniques developed through this study might be useful for complex-valued differential systems.

1.2. Goals in gradient integrators. It should be self-evident from the above-mentioned examples, other widespread applications and references that having an effective gradient integrator in hand is of tremendous value and practical importance. There are already many prominent advocates of this subject [8, 47, 92, 94]. As is typical the case, while the common element of the larger number of applications is gradient dynamics, each application probably deserves and has been given consideration separately for the best efficiency and effectiveness. Among the various algorithms, it is critical to search for the common ground and ask what the essential ingredients must be in order to make an algorithm effective. At this point, there are two major views about a desirable gradient integrator. The difference is at exploiting two related but different properties of the gradient flow, which leads to two related but different treatments.

Maintaining monotonicity. One universal property of a gradient flow is that the potential $f(\mathbf{x}(t))$ decreases monotonically along the solution trajectory $\mathbf{x}(t)$, i.e.,

$$f(\mathbf{x}(t)) \leq f(\mathbf{x}(s)) \quad \text{for } t > s,$$

with strict inequality except at stationary points of f. Correspondingly, we would like a numerical method to preserve the monotonicity at least locally, i.e., if the method starts from $\mathbf{x}_n = \mathbf{x}(t_n)$ exactly with no past errors, then the approximate solution $\mathbf{x}_{n+1}$ at $t_{n+1} = t_n + h$ produced by the numerical method for a step size h would ensure that

$$f(\mathbf{x}_{n+1}) \leq f(\mathbf{x}_n).$$

Gradient integrators for this purpose include algebraically stable Runge-Kutta methods [41, 50], discrete-gradient methods [65, 73, 74], and average vector field collocation methods [38, 65]. More specifically, within the class of algebraically stable Runge-Kutta methods, the Radau IIA methods under a mild step size restriction enjoy good energy reduction and strong damping property for stiff gradient systems, but the Gauss methods do not have the damping property at all. On the other hand, the discrete-gradient methods and the average vector field methods can diminish the energy without any step size restriction, but also have no damping property for stiff gradient systems. Details can be found in a recent survey article [62].

Utilizing contractivity. The emphasis in this paper is on exploiting the second property of a gradient flow — the contractivity. The rationale why this study is important is because it might help achieve the ultimate goal of a gradient flow (1.1) (or a general descent flow (1.5)), namely, finding its limit point $\mathbf{x}^* = \lim_{t\to\infty} \mathbf{x}(t)$, more effectively. At first glance, it seems that we could simply tackle the first-order optimality condition $\nabla f(\mathbf{x}) = 0$ (or the nonlinear system $\mathbf{g}(\mathbf{x}) = 0$) directly by using some general-purpose Newton-like iterative methods. Such a view, however, fails to recognize why the gradient flow comes to play in the first place. We mention two reasons for following a gradient trajectory:

1. First, the critical points of $f(\mathbf{x})$ might not be isolated or unique [1]. Merely satisfying the optimality condition may give rise to a point far away from a desired solution $\mathbf{x}^*$. For instance, the matrix equation $[X, \Pi_0(X)] = 0$ for the Toda lattice has infinitely many solutions, but we look for the diagonal matrix that is orthogonally similar to the initial matrix X_0 for its eigenvalues. The integral curve $X(t)$ guides the way there and we have to follow the curve.
2. Second, most iteration methods suffer from the inherent limitation of local convergence. The gradient flow guarantees convergence of $f(\mathbf{x}(t))$.

On the other hand, it also seems that employing existing numerical integrators to carefully trace the trajectory $\mathbf{x}(t)$ could serve as a means for finding $\mathbf{x}^*$. As reliable as this approach might be, nonetheless, it usually spends expensive computation at the transient state, which would be a waste if only the limit point $\mathbf{x}^*$ is needed. An ideal gradient integrator should follow the true trajectory approximately without striving for precision and should be able to eventually rise to fast convergence to the limit point with high precision [4, 75]. We think that the property of contractivity provides us with several possible avenues for this pursuit. More rigorous technical details will be presented in a subsequent discussion.

For the ease of discussion, we limit our consideration to the ODE setting of gradient dynamics. For PDEs of gradient nature, a suitable discretization of the space variable such as by the method of lines can often reduce the problem to an ODE system in the form (1.1), which will not be elaborated in this paper. Thus, this paper is organized as follows. In Section 2 we bring forth the essential notion of contractivity which has been studied extensively in the literature. The gradient dynamics is contractive. Instead of employing a G-stable linear multi-step method or an algebraically stable Runge-Kutta method which is known for producing contractive numerical solutions, we argue that any A_0-stable method is sufficient for maintaining contractivity for gradient dynamics. We exploit this property by proposing some new iterative schemes in Section 3. Being of higher orders, these methods have the dual capabilities of initially tracking the true trajectory more precisely by using larger step sizes and ultimately transforming themselves into fast converging algorithms. We offer a mathematical justification on why these characteristics are inherited. Some experimental results are reported in Section 4, which eventually suggests that a 2-stage hybrid method in a spirit similar to those proposed in [14, 15] might be the most effective approach for tackling gradient dynamics.

2. Contractivity. For gradient flows, the monotonic decreasing of the f values along one single solution trajectory $\mathbf{x}(t)$ is obvious, but it does not tell how $\mathbf{x}(t)$ is influenced at the presence of perturbations. Relative to neighboring trajectories, perhaps the most conspicuous property of the gradient dynamics is the contractivity which we describe below.

2.1. Contractive vector fields. A vector field[2]

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}) \tag{2.1}$$

with $\mathbf{f} : D \subset [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ is said to satisfy a one-sided Lipschitz condition[3] if there exists a scalar function $\nu(t)$ such that the inequality

$$\langle \mathbf{f}(t, \mathbf{y}_1) - \mathbf{f}(t, \mathbf{y}_2), \mathbf{y}_1 - \mathbf{y}_2 \rangle \leq \nu(t) \|\mathbf{y}_1 - \mathbf{y}_2\|_2^2 \tag{2.2}$$

holds for all $\mathbf{y}_1, \mathbf{y}_2$ in the set $\{\mathbf{y} \in \mathbb{R}^n | (t, \mathbf{y}) \in D\}$. The one-sided Lipschitz function $\nu(t)$ in (2.2) can be negative. Suppose that $\mathbf{y}(t)$ and $\mathbf{z}(t)$ are two solutions of (2.1). Then it can be argued that [60, Section 7.3] (see also [41, Lemma 12.1])

$$\|\mathbf{y}(t_2) - \mathbf{z}(t_2)\|_2 \leq e^{\int_{t_1}^{t_2} \nu(\xi) d\xi} \|\mathbf{y}(t_1) - \mathbf{z}(t_1)\|_2, \quad a \leq t_1 \leq t_2 \leq b. \tag{2.3}$$

A contraction happens (segmentally) over the interval $[a, b]$ when $e^{\int_{t_1}^{t_2} \nu(\xi) d\xi} < 1$ for $a \leq t_1 \leq t_2 \leq b$. This is the essence of the so called nonlinear stability theory [60, Chapter 7]. For our application, we shall limit ourselves throughout this paper to the autonomous case where

$$\mathbf{f}(t, \mathbf{y}) = \mathbf{f}(\mathbf{y}) := -\nabla f(\mathbf{y}). \tag{2.4}$$

The following result showing local contraction follows from the general notion of asymptotic stability.

LEMMA 2.1. *Suppose that $\mathbf{x}^*$ is an isolated stationary point of f and that $\nabla^2 f(\mathbf{x}^*)$ is positive definite*[4]*. Then there is a closed ball $\mathcal{B}$ centered at $\mathbf{x}^*$ and a positive number $\lambda_{\mathcal{B}}$ such that the gradient flow is contractive in the sense that*

$$\|\mathbf{y}(t_2) - \mathbf{z}(t_2)\|_2 \leq e^{-(t_2 - t_1)\lambda_{\mathcal{B}}} \|\mathbf{y}(t_1) - \mathbf{z}(t_1)\|_2, \tag{2.5}$$

for any two gradient flows $\mathbf{y}(t)$ and $\mathbf{z}(t)$ starting from within $\mathcal{B}$ and $t_1 \leq t_2$.

[2] To avoid confusion we shall reserve the boldfaced $\mathbf{x}$ for the state variable in a gradient flow and $\mathbf{x}_k$ for the related iterates, while we shall use $\mathbf{y}$ as the dependent variable of a general ODE system and $\mathbf{y}_n$ for the discrete approximation at $\mathbf{y}(t_n)$.

[3] A conventional Lipschitz condition necessarily implies a one-sided Lipschitz condition, but not the converse.

[4] In the case that $\nabla^2 f(\mathbf{x}^*)$ is only positive semi-definite, the eigenvectors corresponding to the zero eigenvalue form a center manifold. A similar argument can be made by using the center manifold theory [17, 100]. So as to focus on the main ideas, we assume the generic case of positive definiteness in this paper.

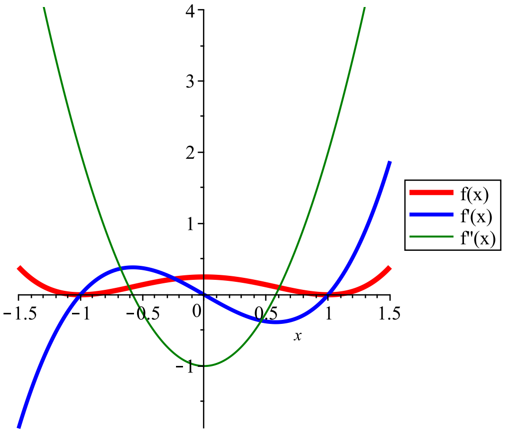

FIGURE 2.1. *Global convergence and local contraction in Example 5.*

The real question is to ask how large the basin of contraction can be for a gradient dynamics and what to do if outside the basin of contraction. Since the following 1-D example can be worked out explicitly, we use it to demonstrate our point [91].

Example 5. The differential equation

$$\frac{dx}{dt} = x - x^3$$

is a gradient system with the potential function $f(x) = \frac{(1-x^2)^2}{4}$. There are three equilibria $\{-1, 0, 1\}$ at which $f''(0) = -1$ and $f''(\pm 1) = 2 > 0$. So, $x = \pm 1$ are asymptotically stable. Indeed, the exact solution is given by

$$x(t) = \frac{x_0}{\sqrt{x_0^2 - x_0^2 e^{-2t} + e^{-2t}}}.$$

So, $x(t)$ converges to ± 1 from everywhere except the unstable equilibrium point $x = 0$. The basin of attraction for the stationary point $x = 1$, for example, is $(0, \infty)$. On the other hand, the contraction (2.5) does not hold everywhere. The ball referred to in Lemma 2.1 around $x = 1$ can be the interval $[1-\delta, 1+\delta]$ with $\delta < 1 - \frac{1}{\sqrt{3}}$. A simple analysis shows that the contraction for flows converging to $x = 1$ does occur whenever flows enter the domain $(\frac{1}{\sqrt{3}}, \infty)$. It is thus illustrated that contraction may not happen currently as convergence. At the initial stage, it is important to rely on the convergence property to steer a gradient flow into the basin of contraction. Once there, we can rely on the contraction property to find an equilibrium without the need of following any particular trajectory precisely.

For a general gradient system, the ball $\mathcal{B}$ referred to in Lemma 2.1 can be as large as any compact convex subset containing $\mathbf{x}^*$ as an interior point, whereas Hessian $\nabla^2 f(\mathbf{x})$ maintains positive definiteness in $\mathcal{B}$. Denote the positive minimal value

$$\lambda_{\mathcal{B}} := \min_{\mathbf{y} \in \mathcal{B}} \lambda^{[1]}(\nabla^2 f(\mathbf{y})), \tag{2.6}$$

where $\lambda^{[1]}(M)$ denotes the smallest eigenvalue of a symmetric matrix M. Then, using the convexity, it can be argued that

$$\langle -\nabla f(\mathbf{y}_1) + \nabla f(\mathbf{y}_2), \mathbf{y}_1 - \mathbf{y}_2 \rangle \leq -\lambda_B \langle \mathbf{y}_1 - \mathbf{y}_2, \mathbf{y}_1 - \mathbf{y}_2 \rangle \tag{2.7}$$

for any $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{B}$. The actual basin of contraction (nearby $\mathbf{x}^*$) might be larger than $\mathcal{B}$, but is difficult to specify in general. Note, however, that if $\mathcal{B}_1 \subset \mathcal{B}_2$, then $\lambda_{\mathcal{B}_1} \geq \lambda_{\mathcal{B}_2}$, implying that the contraction gets exponentially stronger when $\mathbf{x}(t)$ gets closer to $\mathbf{x}^*$. It is worth noting that the above argument is applicable, not just to the stationary point $\mathbf{x}^*$, but to any point $\mathbf{x}$ at which $\nabla^2 f(\mathbf{x})$ is positive definite. After all, the notion of contractivity, as is pointed out in (2.3), is a local property.

The fact that ultimately the vector field of gradients is contractive is of practical importance because, if one solution inadvertently jumps on to a neighboring trajectory, the contraction will bring the paths back to the same equilibrium point. This property gives us the leeway to follow the gradient trajectory loosely while keeping the equilibrium in sight.

2.2. Contractive and transitional methods. When discretizing a contractive vector field for numerical calculation, we certainly wish that the property of contraction is preserved by the underlying numerical method. Considerable efforts have been taken toward this goal. See the book [41, Sections IV.12, V.6, and V.9] and references contained therein for an in-depth discourse on this subject and a chronicle on the development of its theory. A introductory overview of some main results can also be found in the books [24, 60]. Without repeating the details, we simply mention that a G-stable linear multi-step methods [41, Definition 9.1] and an algebraically stable Runge-Kutta methods [41, Definition 12.5] generate contractive numerical solutions.

We quickly point out that these elegant results are developed for general contractive differential systems. For gradient systems, one property standing out is that the Hessian of any sufficiently smooth objective function $f(\mathbf{x})$ is always symmetric and, hence, has only real eigenvalues. Since no complex eigenvalues comes into play, we do not need the full potency of stability conditions of those general methods. Any numerical ODE method whose region of absolute stability contains the negative x-axis while maintaining contractivity is sufficient for following a gradient flow.

In this work, nevertheless, we do not settle for a contractive ODE integrator as is given! We do not adjust the step sizes in accordance with the conventional way where the main concern is to maintain the precision and the stability while tracking the entire trajectory! Instead, we interpret a numerical ODE scheme both as an integrator and as an iterative process. We adjust the step sizes so that the very same scheme transitions itself from a classical integrator to a fast converging iteration. It is this kind non-traditional combination that give us the edge of effectiveness. We demonstrate our point by the following example.

Example 6. Given the current approximate $\mathbf{y}_k$ to $\mathbf{y}(t_k)$ and a step size h_k, suppose that $\mathbf{y}_{k+1}$ is to be approximated by the implicit Euler method. Instead of iterating to convergence as we usually do to obtain $\mathbf{y}_{k+1}$, we perform only one Newton iteration, immediately accept the outcome, and continue to the next step. We thus yield the iterative scheme

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k; \epsilon_k), \tag{2.8}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$\phi(\mathbf{x}; \epsilon) := \mathbf{x} - \left(\frac{1}{\epsilon} I_n + \nabla^2 f(\mathbf{x})\right)^{-1} \nabla f(\mathbf{x}) \tag{2.9}$$

and ϵ is regarded as a parameter. In the meantime, we vary the step size ϵ_k, not based on an error estimator, but according to the so-called "switched evolution relaxation (SER)" strategy [97]

$$\epsilon_k \|\nabla f(\mathbf{x}_k)\|_2 \equiv c \tag{2.10}$$

for a specified constant c. Clearly, such a strategy (2.10) has the characteristics of being relatively small in the initial phase and becoming large when $\nabla f(\mathbf{x}_{k+1})$ converges to zero in the terminal phase of the process. In fact, it can be argued that $\epsilon_{k+1} \geq 2\epsilon_k$ when k is large enough [56, Lemma 2.1]. Thus, the scheme (2.9) initially acts like an ordinary ODE integrator with small step sizes ϵ_k, but asymptotically transitions itself into a fast converging Newton-like iteration when $\epsilon_k \to \infty$. Such a process, known as the pseudo-transient iteration, is a special implicit upwind method used in the PDE community [70] for computing steady-state solutions. We mention in passing that the SER strategy (2.10) is in an interesting contrast to the "artificial time step" strategies discussed in [4] for explicit Euler steps.

We highlight two intrinsic features that will be common to other methods proposed later in this paper. First, we can rewrite (2.8) in such a way that

$$\|\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}_k + \nabla f(\mathbf{x}_k)\|_2 \leq \eta_k \|\nabla f(\mathbf{x}_k)\|_2, \tag{2.11}$$

with

$$\Delta\mathbf{x}_k := \mathbf{x}_{k+1} - \mathbf{x}_k \tag{2.12}$$

and

$$\eta_k := \| \left(I_n + \epsilon_k \nabla^2 f(\mathbf{x}_k)\right)^{-1} \|_2. \tag{2.13}$$

In this case, it is further known that

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|_2 = O(\eta_k \|\mathbf{x}_k - \mathbf{x}^*\|_2 + \|\mathbf{x}_k - \mathbf{x}^*\|_2^2). \tag{2.14}$$

Therefore $\mathbf{x}_k$ converges at least superlinearly. Note that when ϵ_k goes to infinity, η_k goes to zero and (2.14) behaves like quadratic convergence [26].

Second, we claim that the iterative scheme (2.8) is contractive nearby the stationary point $\mathbf{x}^*$. This is important because the scheme (2.8) is not used as an ODE integrator in this area. It can be deduced that the Jacobian of ϕ at $\mathbf{x}^*$ satisfies

$$\left\| \frac{d\phi(\mathbf{x}^*;\epsilon)}{d\mathbf{x}} \right\|_2 = \frac{1}{1+\epsilon\mu^{[1]}} < 1, \tag{2.15}$$

where $\mu^{[1]}$ denotes the smallest eigenvalue $\nabla^2 f(\mathbf{x}^*)$ which, by assumption, is positive. If $\{\mathbf{y}_k\}$ and $\{\mathbf{z}_k\}$ are two sequences generated by (2.8) using the same sequence $\{\epsilon_k\}$ of step sizes and if $\mathbf{y}_k, \mathbf{z}_k$ are sufficiently close to $\mathbf{x}^*$, then by (2.15) we see that

$$\|\mathbf{y}_{k+1} - \mathbf{z}_{k+1}\|_2 = \|\phi(\mathbf{y}_k;\epsilon_k) - \phi(\mathbf{z}_k;\epsilon_k)\|_2 \leq \nu_k \|\mathbf{y}_k - \mathbf{z}_k\|_2 \tag{2.16}$$

for some $\nu_k < 1$. In fact, ϵ_k is expected to be large at the final phase, so the contraction can become stronger when ν_k becomes smaller . Whether there is contraction at other locations along the iteration is harder to tell. The Fréchet derivative of ϕ on a general vector $\mathbf{d} \in \mathbb{R}^n$ at a given point $\mathbf{x}$ is characterized by

$$\begin{aligned} \phi'(\mathbf{x};\epsilon).\mathbf{d} = &\left(I_n - \left(\frac{(\nabla^2 f(\mathbf{x}))^{-1}}{\epsilon} + I_n \right)^{-1} \right) \mathbf{d} \\ &-\epsilon^2 \left(\left(I_n + \epsilon\nabla^2 f(\mathbf{x})\right)^{-1} \left(\nabla^3 f(\mathbf{x}).\mathbf{d}\right) \left(I_n + \epsilon\nabla^2 f(\mathbf{x})\right)^{-1} \right) \nabla f(\mathbf{x}), \end{aligned} \tag{2.17}$$

where $\nabla^3 f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the Fréchet derivative of $\nabla^2 f(\mathbf{x})$ and is a tensor-to-vector multiplication. At a given point $\mathbf{x}$ where $\nabla^2 f(\mathbf{x})$ is positive definite, it can be argued that there exists a positive number $\Upsilon(\mathbf{x})$ such that if $\epsilon_k \in [0, \Upsilon(\mathbf{x})]$, then the operator norm $\phi'(\mathbf{x};\epsilon_k)$ is bounded by 1 and, hence, a contraction happens. In the pseudo-transient iteration, however, we do not wish to see a bound on ϵ_k. Such a dilemma is somewhat of less concern in practice because before we reach the basin of contraction of $\mathbf{x}^*$ we normally follow the gradient trajectory carefully anyway.

Different from the energy-diminishing approach in [62], the idea of the pseudo-transient method is to start the scheme with a small ϵ_k to let the iteration process mimic an explicit, time-accurate integration. This allows staying reasonably close to the gradient trajectory which leads to the basin of attraction. The step size selection mechanism controls the transition. Diminishing values of $\nabla f(\mathbf{x}_k)$ indicate large ϵ_k can be used, which transforms the method to the Newton iteration and, hence, attains fast convergence. Our interest in this paper is to search for some other gradient integrators of similar nature and test their workability.

3. High-order BDF-based methods. Thus far, we have been speculating the basic idea of an ideal method that can play dual roles in effectively tackling a gradient dynamics. The pseudo-transient method outlined in Example 6 is based on the implicit Euler scheme which, used as a numerical ODE integrator in the crucial initial phase, is of order 1 only. It is generally expected that high-order methods can integrate faster with larger step sizes while maintaining the desirable precision. This should help move along the trajectory more effectively into the final phase for finding the stable equilibrium. The main objective of this paper is to propose some high-order methods. To convey the idea, we concentrate on the backward differential formulas (BDF) in this section. Our contribution is at establishing a mathematical foundation showing that, when properly augmented, these schemes enjoy similar contractive and transitional properties as the first-order pseudo-transient method. Similar arguments can be generalized to other types of schemes such as the numerical differential formulas (NDF), which we will outline without giving as much detail in the appendix.

One important feature of the classical p-step backward differentiation formula, $p = 1, \ldots, 6$,

$$\mathbf{y}_{k+1} = \sum_{j=0}^{p-1} \alpha_j \mathbf{y}_{k-j} + h\beta \mathbf{f}(\mathbf{y}_{k+1}), \tag{3.1}$$

is that its region of absolute stability contains the whole of negative real axis. This is desirable for using large step sizes at the final phase of a gradient dynamics. Nevertheless, we do not employ existing, highly sophisticated BDF-based stiff solvers for gradient flows. Instead, in the same spirit as the pseudo-transient iteration, we start from $\mathbf{x}_k$ with step size ϵ_k and advance to $\mathbf{x}_{k+1}$ by taking only one Newton iteration, which can be expressed as

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \left(\frac{1}{\epsilon_k \beta} I_n + \nabla^2 f(\mathbf{x}_k)\right)^{-1} \left(\frac{1}{\epsilon_k \beta}\left(\sum_{j=0}^{p-1} \alpha_j \mathbf{x}_{k-j} - \mathbf{x}_k\right) - \nabla f(\mathbf{x}_k)\right). \tag{3.2}$$

Two remarks are worth noting. First, in contrast to (2.9), the right-hand side of (3.2) involves multiple steps. Such an iteration, however, is no more expensive than that in (2.8), except for a few extra memories of past values and vector arithmetic operations. Second, we use the same coefficients $\alpha_0, \ldots, \alpha_p$ and β from a constant step size BDF scheme for all iterates, even though the past values $\mathbf{x}_k, \ldots, \mathbf{x}_{k-p+1}$ may have obtained from variable step sizes. From the numerical ODE point of view, such a mechanical plug-in makes no sense and is wrong. Nonetheless, we shall argue for its merits when (3.2) is regarded as a stand-alone iterative scheme. It may seem more reasonable to first convert (3.1) to a one-step scheme via the Nordsieck transform and then to form the iteration scheme from there by taking one Newton step in the same spirit as we do here. We shall prove in the later part of this section that the Nordsieck scheme has poor, if any, transitional property. This irony of doing what is right (as an ODE method) and yet getting worse behavior (in convergence) is an interesting contrast to our approach.

Through the vectorization

$$X_k := \left[\mathbf{x}_k^\top, \ldots, \mathbf{x}_{k-p+1}^\top\right]^\top \in \mathbb{R}^{pn}, \tag{3.3}$$

we may regard the iteration (3.2) as a fixed-point iteration

$$X_{k+1} = \psi(X_k; \epsilon_k), \tag{3.4}$$

with

$$\psi(X;\epsilon) := \begin{bmatrix} \mathbf{x}^{[0]} + \left(\frac{1}{\epsilon\beta} I_n + \nabla^2 f(\mathbf{x}^{[0]})\right)^{-1} \left(\frac{1}{\epsilon\beta}\left(\sum_{j=0}^{p-1} \alpha_j \mathbf{x}^{[j]} - \mathbf{x}^{[0]}\right) - \nabla f(\mathbf{x}^{[0]})\right) \\ \mathbf{x}^{[0]} \\ \vdots \\ \mathbf{x}^{[p-2]} \end{bmatrix} \tag{3.5}$$

where X is partitioned in blocks as $X = \left[\mathbf{x}^{[0]\top}, \ldots, \mathbf{x}^{[p-1]\top}\right]^\top$ with $\mathbf{x}^{[j]} \in \mathbb{R}^n$. The following result shows that, similar to the pseudo-transient method, the more sophisticated scheme has the desirable property of contractivity.

THEOREM 3.1. *Suppose that $\mathbf{x}^*$ is an isolated stationary point of f and that $\nabla^2 f(\mathbf{x}^*)$ is positive definite. Then there exists a neighborhood $\mathcal{B}$ of $\mathbf{x}^*$ such that*

$$\|Y_{k+1} - Z_{k+1}\|_2 = \|\psi(Y_k; \epsilon_k) - \psi(Z_k; \epsilon_k)\|_2 \leq \nu_k \|Y_k - Z_k\|_2 \tag{3.6}$$

with some $\nu_k < 1$, provided components of Y_k and Z_k are from within $\mathcal{B}$.

Proof. The fixed-point of (3.4) is $X^* = [\mathbf{x}^{*\top}, \ldots, \mathbf{x}^{*\top}]^\top$ at which the Jacobian of ψ is given by

$$\frac{d\psi(X^*)}{dX} = \begin{bmatrix} \Omega(\epsilon)\frac{\alpha_0}{\epsilon\beta} & \Omega(\epsilon)\frac{\alpha_1}{\epsilon\beta} & & \ldots & & \Omega(\epsilon)\frac{\alpha_{p-1}}{\epsilon\beta} \\ I_n & 0 & 0 & \ldots & & 0 \\ 0 & I_n & 0 & & & 0 \\ & & & \ddots & & \\ \vdots & & \vdots & & & \vdots \\ 0 & 0 & 0 & \ldots & I_n & 0 \end{bmatrix}, \tag{3.7}$$

where for convenience we introduce the abbreviation

$$\Omega(\epsilon) := \left(\frac{1}{\epsilon\beta} I_n + \nabla^2 f(\mathbf{x}^*)\right)^{-1}. \tag{3.8}$$

The Jacobian (3.7) appears as the structure of a block companion matrix. So eigenvalues of (3.7) are precisely those of the matrix polynomial

$$p(\lambda) := I_n \lambda^p - \Omega(\epsilon)\frac{\alpha_0}{\epsilon\beta}\lambda^{p-1} - \Omega(\epsilon)\frac{\alpha_1}{\epsilon\beta}\lambda^{p-2} - \ldots - \Omega(\epsilon)\frac{\alpha_{p-1}}{\epsilon\beta}. \tag{3.9}$$

Let the eigenvalues of $\nabla^2 f(\mathbf{x}^*)$ be denoted by $\mu^{[1]} \leq \mu^{[2]} \leq \ldots \leq \mu^{[n-1]} \leq \mu^{[n]}$. It suffices to consider the roots of the polynomials

$$q(\lambda; \mu^{[i]}) := \lambda^p - \frac{\alpha_0}{(1 + \epsilon\beta\mu^{[i]})}\lambda^{p-1} - \ldots - \frac{\alpha_{p-1}}{(1 + \epsilon\beta\mu^{[i]})}, \quad i = 1, \ldots, n. \tag{3.10}$$

If the step size ϵ is large enough such that

$$\sum_{j=0}^{p-1} |\alpha_j| < 1 + \epsilon\beta\mu^{[1]}, \tag{3.11}$$

then, using the Rouché theorem, we see that all roots are bounded within the unit disk. □

The above proof does not make use of the values of the coefficients α_j and β of the BDF scheme. These coefficients really come to play in the following lemma asserting the transitional property that the multi-step scheme (3.2) can also be regarded as an inexact Newton method, whence the fast convergence.

THEOREM 3.2. *Define $\zeta_0 = 1$ and*

$$\zeta_j := 1 - \sum_{i=0}^{j-1} \alpha_i, \quad j = 1, \ldots p - 1. \tag{3.12}$$

Then the BDF-based iteration (3.2) is equivalent to

$$\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}_k + \nabla f(\mathbf{x}_k) = -\frac{1}{\epsilon_k}\left(\sum_{j=0}^{p-1}\frac{\zeta_j}{\beta}\Delta\mathbf{x}_{k-j}\right), \tag{3.13}$$

whereas the summation on the right side of (3.13) is an average of the time series $\Delta\mathbf{x}_k, \ldots, \Delta\mathbf{x}_{k-p+1}$.

Proof. It is a known fact that for each fixed p, the coefficients of a BDF in the form (3.1) necessarily satisfy the unique algebraic relationships [41, 60]

$$\sum_{j=0}^{p-1}\alpha_j = 1, \tag{3.14}$$

$$\sum_{j=0}^{p-1}(j+1)\alpha_j = \beta. \tag{3.15}$$

Using (3.14), we can rewrite the summation on the right side of (3.2) recursively as

$$\mathbf{x}_k - \sum_{j=0}^{p-1}\alpha_j\mathbf{x}_{k-j} = \sum_{j=0}^{p-1}\alpha_j(\mathbf{x}_k - \mathbf{x}_{k-j}) = \sum_{j=1}^{p-1}\left(1 - \sum_{i=0}^{j-1}\alpha_i\right)\Delta\mathbf{x}_{k-j}.$$

An rearrangement of terms in (3.2) leads to (3.13). To see the averaging effect, the identity

$$\sum_{j=0}^{p-1}\zeta_j = \beta \tag{3.16}$$

is simply a rearrangement of the summation in (3.15). □

It should be noted that some of the weights $\frac{\zeta_j}{\beta}$ in (3.13) might be negative, but they sum to be 1. The average nature

$$\Delta\overline{\mathbf{x}}_k := \sum_{j=0}^{p-1}\frac{\zeta_j}{\beta}\Delta\mathbf{x}_{k-j} \tag{3.17}$$

serves as a good estimate of the convergence in the following sense which generalizes that in (2.13) for $p > 1$.

COROLLARY 3.3. *Suppose that the SER strategy (2.10) is used. Then the sequence* $\{\Delta\overline{\mathbf{x}}_k\}$ *determines the rate of convergence. In particular, if* $\lim_{k\to\infty}\Delta\overline{\mathbf{x}}_k = 0$*, then* $\mathbf{x}_k$ *converges to a stationary point* $\mathbf{x}^*$ *superlinearly.*

Proof. By Lemma 3.2, the residual of the (Newton) iteration is given by $\mathbf{r}_k := -\frac{1}{\epsilon_k}\Delta\overline{\mathbf{x}}_k$. By the theory of the inexact Newton method, the rate of convergence depends on the forcing sequence [26, Corollary 3.5]

$$\eta_k := \frac{\|\mathbf{r}_k\|_2}{\|\nabla f(\mathbf{x}_k)\|_2} = \frac{1}{c}\Delta\overline{\mathbf{x}}_k, \tag{3.18}$$

where $c := \epsilon_0\|\nabla f(\mathbf{x}_0)\|_2$ is a constant determined by the first step size ϵ_0. □

By now, most of properties known for the pseudo-transient method have been generalized to the class of BDFs. Of particular usefulness is that a higher order BDF has the advantages of allowing larger steps in the initial stage when tracing a gradient trajectory and, when $\mathbf{x}_k$ is close enough to x^*, it transitions itself to a faster Newton method.

In the remaining of this section, we clarify the question of whether the multi-step scheme (3.2) should be first rewritten in the one-step Nordsieck form [41, 60] before employing the SER strategy to change the step sizes. It is known that by defining the $n \times (p+1)$ matrix

$$Z_k := \left[\mathbf{y}_k, h\mathbf{y}_k^{(1)}, \frac{h^2}{2}\mathbf{y}_k^{(2)}, \ldots, \frac{h^p}{p!}\mathbf{y}_k^{(p)}\right], \tag{3.19}$$

where $\mathbf{y}_k^{(j)}$ is an approximation to $\mathbf{y}^{(j)}(\mathbf{x}_k)$, the p-step BDF method is equivalent to [40, Section III.6]

$$Z_{k+1} = Z_k P^\top + \left(h\mathbf{f}(\mathbf{y}_{k+1}) - Z_k[0,1,2,\ldots,p]^\top\right)\ell^\top, \tag{3.20}$$

where $P \in \mathbb{R}^{(p+1)\times(p+1)}$ is Pascal matrix

$$P := \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 2 & 3 & & p \\ 0 & 0 & 1 & 3 & & \\ \vdots & & & & & \\ 0 & 0 & 0 & & \ldots & 1 \end{bmatrix},$$

$\ell = [\beta, 1, \ell_2, \ldots, \ell_p]^\top \in \mathbb{R}^{(p+1)\times 1}$ is a specific vector whose other entries are known [40, Section III.6, Table 6.2]. The first column in the equation (3.20) is a nonlinear system

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \sum_{i=1}^{p}\frac{h^i}{i!}\mathbf{y}_k^{(i)} + \beta\left(h\mathbf{f}(\mathbf{y}_{k+1}) - \sum_{i=1}^{p}\frac{h^i}{(i-1)!}\mathbf{y}_k^{(i)}\right) \tag{3.21}$$

in the unknown $\mathbf{y}_{k+1}$. Once $\mathbf{y}_{k+1}$ is obtained, the other columns are explicitly determined from (3.20).

We work out the Nordsieck-based iteration for the case $p = 2$ as an example.

Example 7. The constant step size 2-step BDF method is

$$\mathbf{y}_{k+1} = \frac{4}{3}\mathbf{y}_k - \frac{1}{3}\mathbf{y}_{k-1} + \frac{2h}{3}\mathbf{f}(\mathbf{y}_{k+1}). \tag{3.22}$$

The corresponding nonlinear system in the Nordsieck form is given by

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h}{3}\mathbf{y}'_k - \frac{h^2}{6}\mathbf{y}''_k + \frac{2h}{3}\mathbf{f}(\mathbf{y}_{k+1}). \tag{3.23}$$

The difference between (3.22) and (3.23) is that the former requires the past value $\mathbf{y}_{k-1}$ with a fixed step size h, while the latter depends solely on the current value $\mathbf{y}_k$ and its derivatives and, hence, h can be arbitrary.

Applying (3.23) to the gradient flow by taking one Newton correction with step size ϵ_k yields an explicit iterative scheme

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \left(\frac{3}{2\epsilon_k}I_n + \nabla^2 f(\mathbf{x}_k)\right)^{-1}\left(\frac{3}{2\epsilon_k}\left(\epsilon_k\mathbf{x}'_k - \frac{1}{3}\frac{\epsilon_k^2}{2}\mathbf{x}''_k\right)\right). \tag{3.24}$$

If the step size is changed to ϵ_{k+1}, then we are ready for the next iteration $\mathbf{x}_{k+2}$ by taking

$$\epsilon_{k+1}\mathbf{x}'_{k+1} := -\epsilon_{k+1}\nabla f(\mathbf{x}_{k+1}), \tag{3.25}$$

$$\frac{\epsilon_{k+1}^2}{2}\mathbf{x}''_{k+1} = -\frac{1}{3}\left(\frac{\epsilon_{k+1}}{\epsilon_k}\right)^2\left(\epsilon_k\mathbf{x}'_k - \frac{\epsilon_k^2}{2}\mathbf{x}''_k + \epsilon_k\nabla f(\mathbf{x}_{k+1})\right). \tag{3.26}$$

To analyze the limiting behavior of (3.24) near $\mathbf{x}^*$, it is necessary to consider the local behavior of $\mathbf{x}_k, \mathbf{x}'_k, \mathbf{x}''_k$ simultaneously. Define the column vector $W_k := [\mathbf{x}_k^\top, {\mathbf{x}'_k}^\top, {\mathbf{x}''_k}^\top]^\top \in \mathbb{R}^{3n}$. We can rewrite (3.24) and the associated derivative information as a fixed-point iteration

$$W_{k+1} = \vartheta(W_k; \epsilon_k), \tag{3.27}$$

where $\vartheta : \mathbb{R}^{3n} \longrightarrow \mathbb{R}^{3n}$ is defined by

$$\vartheta(W;\epsilon) := \begin{bmatrix} \mathbf{w}^{[0]} + \left(I_n + \frac{2\epsilon}{3}\nabla^2 f\left(\mathbf{w}^{[0]}\right)\right)^{-1}\left(\epsilon\mathbf{w}^{[1]} - \frac{\epsilon^2}{6}\mathbf{w}^{[2]}\right) \\ -\nabla f\left(\mathbf{w}^{[0]} + \left(I_n + \frac{2\epsilon}{3}\nabla^2 f\left(\mathbf{w}^{[0]}\right)\right)^{-1}\left(\epsilon\mathbf{w}^{[1]} - \frac{\epsilon^2}{6}\mathbf{w}^{[2]}\right)\right) \\ \frac{2}{3\epsilon}\left(-\nabla f\left(\mathbf{w}^{[0]} + \left(I_n + \frac{2\epsilon}{3}\nabla^2 f\left(\mathbf{w}^{[0]}\right)\right)^{-1}\left(\epsilon\mathbf{w}^{[1]} - \frac{\epsilon^2}{6}\mathbf{w}^{[2]}\right)\right) - \mathbf{w}^{[1]} + \frac{\epsilon}{2}\mathbf{w}^{[2]}\right) \end{bmatrix},$$

and W is partitioned in blocks as $W = \left[\mathbf{w}^{[0]\top}, \mathbf{w}^{[1]\top}, \mathbf{w}^{[2]\top}\right]^\top$ with $\mathbf{w}^{(j)} \in \mathbb{R}^n$. It is easy to check that the fixed point of (3.27) must be of the form $(\mathbf{x}^*, \mathbf{0}, \mathbf{0})$. Through some tedious but straightforward manipulations, it can be shown that the spectrum of the Jacobian matrix of ϑ at the equilibrium point $(\mathbf{x}^*, \mathbf{0}, \mathbf{0})$ consists of eigenvalues of each of the 3×3 matrices

$$\begin{bmatrix} 1 & \frac{\epsilon}{1+\frac{2}{3}\epsilon\mu^{[i]}} & -\frac{\epsilon^2}{6\left(1+\frac{2}{3}\epsilon\mu^{[i]}\right)} \\ -\mu^{[i]} & -\frac{\epsilon\mu^{[i]}}{1+\frac{2}{3}\epsilon\mu^{[i]}} & \frac{\epsilon^2\mu^{[i]}}{6\left(1+\frac{2}{3}\epsilon\mu^{[i]}\right)} \\ -\frac{2}{3\epsilon}\mu^{[i]} & -\frac{2}{3\epsilon}\left(\frac{\epsilon\mu^{[i]}}{\left(1+\frac{2}{3}\epsilon\mu^{[i]}\right)}+1\right) & \frac{2}{3\epsilon}\left(\frac{\epsilon^2\mu^{[i]}}{6\left(1+\frac{2}{3}\epsilon\mu^{[i]}\right)}+\frac{\epsilon}{2}\right) \end{bmatrix}, \quad i = 1, \dots n, \tag{3.28}$$

which are

$$0, \ \frac{2 \pm \sqrt{1 - 2\epsilon\mu^{[i]}}}{3 + 2\epsilon\mu^{[i]}}, \quad i = 1, \dots n, \tag{3.29}$$

respectively. For each $i = 1, \dots n$, all eigenvalues are in the unit disk for any step sizes ϵ and converge to zero when ϵ goes to infinity. Hence, the local convergence by the iteration (3.24) is guaranteed.

Without repeating the details, the above argument can be extended to the general $p = 1, \dots, 6$. We even can still write the scheme derived from the Nordsieck-based methods as an inexact Newton iteration, but it is at this point that the difference between a Nordsieck-based iteration and a BDF-based iteration becomes more obvious as we now explain.

LEMMA 3.4. *For a fixed $p = 1, \dots, 6$, let $\mathbf{x}_{k+1}$ denote the result of one Newton step of the nonlinear equation (3.21) applied to the gradient flow with step size ϵ_k and starting value $\mathbf{x}_k$. Then*

$$\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}_k + \nabla f(\mathbf{x}_k) = \nabla f(\widehat{\mathbf{y}}_{k+1}) + \frac{1}{\beta\epsilon_k}(\widehat{\mathbf{y}}_{k+1} - \mathbf{x}_{k+1}), \tag{3.30}$$

where $\widehat{\mathbf{y}}_{k+1}$, depending on fixed $\mathbf{x}_k$ and ϵ_k, is the exact solution to (3.21).

Proof. One Newton step applied to (3.21) leads to

$$\left(I_n + \beta\epsilon_k\nabla^2 f(\mathbf{x}_k)\right)\Delta\mathbf{x}_k = \sum_{i=1}^{p}\frac{h^i}{i!}\mathbf{x}_k^{(i)} - \beta\sum_{i=1}^{p}\frac{h^i}{(i-1)!}\mathbf{x}_k^{(i)} - \beta\epsilon_k\nabla f(\mathbf{x}_k).$$

A rearrangement shows that

$$\beta\epsilon_k\left(\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}_k + \nabla f(\mathbf{x}_k)\right) = \left(\widehat{\mathbf{y}}_{k+1} - \mathbf{x}_k + \beta\epsilon_k\nabla f(\widehat{\mathbf{y}}_{k+1})\right) - \Delta\mathbf{x}_k,$$

whereas the right hand side follows from (3.21). □

Note that the residual $\mathbf{r}_k := \nabla f(\widehat{\mathbf{y}}_{k+1}) + \frac{1}{\beta\epsilon_k}(\widehat{\mathbf{y}}_{k+1} - \mathbf{x}_{k+1})$ is only implicitly defined because $\widehat{\mathbf{y}}_{k+1}$ is yet to be calculated. If the SER strategy is used in this one-step Nordsieck setting, then the forcing sequence becomes

$$\eta_k = \frac{1}{\beta c}\|\beta\epsilon_k\nabla f(\widehat{\mathbf{y}}_{k+1}) + (\widehat{\mathbf{y}}_{k+1} - \mathbf{x}_{k+1})\|_2 \tag{3.31}$$

which, according the theory of inexact Newton methods [26], determines the rate of convergence. An intuitive way to see why η_k would be small is that if $\{\mathbf{x}_k\}$ ever converges, then the sequence $\{\widehat{\mathbf{y}}_k\}$ should behave like

$$\widehat{\mathbf{y}}_{k+1} \approx \mathbf{x}^* - \beta\epsilon_k\nabla f(\widehat{\mathbf{y}}_{k+1}) \approx \mathbf{x}_{k+1} - \beta\epsilon_k\nabla f(\widehat{\mathbf{y}}_{k+1}). \tag{3.32}$$

As such, there should be a cancelation in (3.31) to make η_k small. In our numerical experiment, however, we find that η_k diminishes slowly in general for a reason not fully understood at present. For now, we conclude that, under the SER strategy, the transitional ability of the Nordsieck-based iterative scheme is poor and do not recommend this conversion.

4. Numerical experiment. Thus far, we have proposed some high-order BDF-based iterative schemes that are contractive and transitional. In this section we carry out a few numerical experiments to demonstrate these concepts. For quick implementation, critical issues such as effective step size selection or other cost reduction tactics are not considered in these experiments. We employ the same SER strategy across the board for all orders, which itself may not be the best policy. We aim at comparing step sizes variation and checking whether the methods themselves will transform at the end stage.

We choose a few interesting application problems with known sources to benchmark the performance. These are the Müller-Brown potential energy surface (MBPES) [71], the Powell badly scaled function (PBSF) [69], the chemical equilibrium system (CES) [66], the stead-state reaction rate equation (SSRRE) [82], and the circuit design problem (CDP) [79], all of which are known to be challenging [2]. These problems are set up as gradient dynamical systems in the form (1.1). Although these are optimization problems whose equilibria can be found directly by existing optimization packages, we stress that there are numerous applications where the process of evolution from the starting point to the stationary point is important to practitioners. Tracking the gradient dynamics is more than just finding its stationary points.

4.1. Test on low-precision low-order ODE integrators. To set a point of reference, we employ standard ODE solvers in MATLAB as the base for evaluating the performance. These solvers are the ode23s which is order 2 and the BDF in ode15s with MaxOrder = 2. Built in these ODE solvers is the step size selection strategy which estimates the local error e_i in the entry y_i at each step and chooses step sizes to ascertain that the criterion

$$|e_i| \leq |y_i|\mathsf{RelTol} + \mathsf{AbsTol} \tag{4.1}$$

is satisfied. This more sophisticated strategy differs from the SER strategy in that the latter takes no local errors into account. We are interested in low precision with the hope that the contractivity of the gradient dynamics will keep the convergence in bay, so we set the local tolerance $\mathsf{AbsTol} = \mathsf{RelTol} = 10^{-2}$.

We do not preset the interval length of integration, but let the process terminate automatically when one of the following events occurs:

$$\|\nabla f(\mathbf{x}_k)\| < \mathrm{TerTol} \quad \text{or} \quad \frac{\|\nabla f(\mathbf{x}_k)\|}{\|\nabla f(\mathbf{x}_0)\|} < \mathrm{TerTol} \quad \text{or} \quad \|\mathbf{x}_k - \mathbf{x}_{k-3}\| < \mathrm{TerTol}, \tag{4.2}$$

where the termination tolerance TerTol is set at 10^{-9}. On the other hand, high precision integration, such as setting $\mathsf{AbsTol} = \mathsf{RelTol} = 10^{-12}$ to follow the flow closely, provides information about how long the analytic gradient trajectory takes to reach its equilibrium point [75].

Involved in the overhead of these methods are time steps, failed steps, function evaluations, Jacobian evaluations, LU decompositions, linear solvers, and so on. Roughly speaking, these parameters depend linearly on the number of steps, so we report only the time steps in Table 4.1. Detailed statistics on this experiment and all other tests reported herein can be furnished upon request.

	ode23 (10^{-2})	BDF (order 2) (10^{-2})	ode15s (full) (10^{-12})	integration length
MBPES	21	51	1441	0.06
PBSF	249	142	1651	4.94e+04
CES	253	526	5133	1.01e+05
SSRRE	66	159	3378	5.39e+05
CDP	86	139	3358	6.01e+02

TABLE 4.1

Time steps and length of integration needed by standard ODE integrators.

It should be clear from this experiment that low-precision and low-order ODE integrators are sufficient for gradient dynamical systems. There is really no need to track the trajectories in high precision. This, of course, is mainly due to the inherent contractivity of the gradient dynamics. It is also worth noting that, even though the integration lengths might be long, the time steps required are only in hundreds by these integrator, suggesting that the step size selection strategy (4.1) is also capable of adopting very large step sizes while satisfying the specified local tolerance condition.

4.2. Test on high order BDF-based pseudo-transient methods. Using the same stopping criteria in the preceding experiment, we now test the BDF-based pseudo-transient methods (3.2) on the same five problems. We choose initial step size $\epsilon_0 = 10^{-2}$. After generating enough starting values via the explicit Euler formula, we vary ϵ_k by the SER strategy. Summarized in Table 4.2 is a comparison on the numbers of steps taken by the best multi-step BDF iterative schemes (3.2) against those by `ode23s` with local error tolerance 10^{-2}. It should quite obvious that the BDF-based pseudo-transient methods generally require much fewer iterations for convergence.

	ode23 (10^{-2})	BDF-based Transient	(order)
MBPES	21	9	(2)
PBSF	249	25	(4)
CES	253	30	(1)
SSRRE	66	27	(2)
CDP	86	36	(2)

TABLE 4.2

Comparison of time steps between `ode23s` as an ODE integraor versus the best (order) transitional BDF scheme (3.2).

What is not clear in Table 4.2 is why the best BDF-based method is not necessarily of the highest order. This might have something to do with our using the SER strategy for all methods. This strategy is proven successful only for the order-1 pseudo-transient method [56, Lemma 2.1], but might not be as effective for higher-order methods. This area is widely open for further research! See [4, 34, 46, 63, 75, 96]. To demonstrate our point, we repeatedly test the CES problem by different starting values. From Table 4.3, we see that the cost of `ode23s` is fairly stable, but the numbers of steps for the BDF-based methods fluctuate.

ode 23s	BDF1	BDF2	BDF3	BDF4	BDF5
253	39	20	31	29	22
231	32	29	26	36	77
224	36	17	45	44	40
229	24	214	24	39	43
227	38	43	33	32	36

TABLE 4.3

Repeated tests of `ode23s` vs. BDF-based iterative schemes (3.2) on CSE with different starting values.

On the other hand, Figure 4.1 typifies the history of gradient reduction and the variation of step sizes based on the SER strategy. The drawings clearly manifest the quadratic convergence of the gradient and the exponential growth of ϵ_k at the final stage of iteration, even though the SER strategy is used as a rough estimator.

5. Conclusion. Gradient dynamics appears in a wide range of disciplines. This work explores some general mathematical characteristics that an effective gradient dynamics algorithm should bear. In particular, we propose some ODE-based iterative methods, but do not follow the usual numerical ODE protocols. Innate to these methods are their transition capability and contractivity property. Preliminary experiments, even by a rudimentary step size selection strategy, evidence the potential of these methods.

Many questions remain open, including proper step size control strategies and other cost reduction tactics [14, 15]. Also important is the generalization to structured gradient dynamics. One such an application with significant consequence is a proper projected gradient method for the Toda lattice (1.3) for eigenvalue computation.

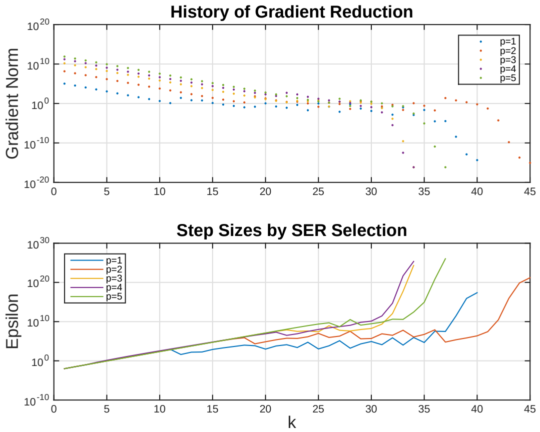

FIGURE 4.1. *Step size variations by the SER strategy and gradient reduction for CES by the BDF-based scheme (3.2).*

6. Appendix. As is often the case in applications, algorithms should be specially designed to handle specific gradient dynamics for its best efficiency. We have exploited the BDF-based iterative schemes as a way to tackle the general gradient dynamics for its contractive and transitional properties. Certainly, there are other possible approaches. Without going into lengthy details, we briefly mention two other possible ODE-based iterative schemes for gradient dynamics.

6.1. NDF-based methods. Implemented in the solver `ode15s` in the MATLAB ODE suite as an effective solver for still ODEs is the so called numerical differentiation formulas (NDF):

$$\mathbf{y}_{k+1} = \sum_{j=0}^{p-1} \alpha_j \mathbf{y}_{k-j} + h\beta \mathbf{f}(\mathbf{y}_{k+1}) + \beta\kappa\gamma_p \left(\mathbf{y}_{k+1} - \widetilde{\mathbf{y}}_{k+1}\right), \tag{6.1}$$

where the quantity

$$\widetilde{\mathbf{y}}_{k+1} := \sum_{m=0}^{p} \triangledown^m \mathbf{y}_k \tag{6.2}$$

is calculable from previous steps via the back difference operator $\triangledown \mathbf{y}_k := \mathbf{y}_k - \mathbf{y}_{k-1}$ and the value $\gamma_p := \sum_{j=1}^{p} \frac{1}{j}$ is fixed. The motivation for the NDFs is that the parameter κ can be adjusted to gain more accuracy without significantly sacrificing the stability [83]. Similar to the BDFs, the NDFs can be made into contractive and transitional iterative schemes for gradient dynamics.

Scheme: One application of the Newton iteration of (6.1) with step size ϵ_k to the gradient flow leads to the iterative scheme:

$$\mathbf{x}_{k+1} := \mathbf{x}_k - \left(\frac{1-\kappa\gamma_p\beta}{\epsilon_k\beta} I_n + \nabla^2 f(\mathbf{x}_k)\right)^{-1} \left(\nabla f(\mathbf{x}_k) - \frac{1}{\epsilon_k\beta}\left(\sum_{j=0}^{p-1} \alpha_j \mathbf{x}_{k-j} - \mathbf{x}_k + \kappa\gamma_p\beta\left(\mathbf{x}_k - \widetilde{\mathbf{x}}_{k+1}\right)\right)\right). \tag{6.3}$$

Transitionality: From the identity that

$$\triangledown^m \mathbf{y}_k = \sum_{j=0}^{m} (-1)^j \binom{m}{j} \mathbf{y}_{k-j},$$

we can write the intermediate vector $\widetilde{\mathbf{x}}_{k+1}$ as

$$\begin{aligned}\widetilde{\mathbf{x}}_{k+1} &= \mathbf{x}_k + \sum_{m=1}^{p} \triangledown^m \mathbf{x}_k = \mathbf{x}_k + \sum_{m=1}^{p} \triangledown^{m-1} \Delta\mathbf{x}_{k-1} = \mathbf{x}_k + \sum_{m=1}^{p}\sum_{j=0}^{m-1} (-1)^j \binom{m-1}{j} \Delta\mathbf{x}_{k-1-j} \\ &= \mathbf{x}_k + \sum_{j=0}^{p-1}\sum_{m=j+1}^{p} (-1)^j \binom{m-1}{j} \Delta\mathbf{x}_{k-1-j} = \mathbf{x}_k + \sum_{j=1}^{p} \eta_j \Delta\mathbf{x}_{k-j},\end{aligned}$$

with the abbreviations

$$\eta_j := \sum_{m=j}^{p} (-1)^{j-1} \binom{m-1}{j-1}, \quad j = 1, \dots p. \tag{6.4}$$

Upon substituting $\widetilde{\mathbf{x}}_{k+1}$ into (6.3) and rearranging terms, and with the additional notation $\eta_0 := -1$, we obtain the equivalence

$$\nabla^2 f(\mathbf{x}_k)\Delta\mathbf{x}_k + \nabla f(\mathbf{x}_k) = -\frac{1}{\epsilon_k}\left(\sum_{j=0}^{p-1}\left(\frac{\zeta_j}{\beta} + \kappa\gamma_p\eta_j\right)\Delta\mathbf{x}_{k-j} + \kappa\gamma_p\eta_p\Delta\mathbf{x}_{k-p}\right). \tag{6.5}$$

The fact that $\sum_{j=0}^{p} \eta_j = 0$, together with (3.16), implies the coefficients of the time series $\Delta\mathbf{x}_k, \dots, \Delta\mathbf{x}_{k-p}$ sum to the unity. So the right side of (6.3) can be regarded as an average. Thus the iterative scheme (6.3) is still an inexact Newton method with averaging. Note that (6.5) requires one extra memory than (3.13).

Contractivity: Since the iteration (6.3) can be regarded as an inexact Newton method, it already suggests contractivity. An argument similar to Theorem 3.1 can establish that the NRF-based iteration (3.13) is indeed contractive.

6.2. Rosenbrock-based methods. Implemented in the solver ode23s in the MATLAB ODE suite is the special Rosenbrock method

$$\left\{\begin{array}{lcl} \mathbf{y}_{n+1} &=& \mathbf{y}_n + h\mathbf{k}_2 \\ W\mathbf{k}_1 &=& \mathbf{f}(\mathbf{y}_n) \\ W\mathbf{k}_2 &=& \mathbf{f}\left(\mathbf{y}_n + \frac{1}{2}h\mathbf{k}_1\right) - hdJ\mathbf{k}_1 \\ W\mathbf{k}_3 &=& \mathbf{f}(\mathbf{y}_n + h\mathbf{k}_2) + (e-2)hdJ\mathbf{k}_1 - ehdJ\mathbf{k}_2, \end{array}\right. \tag{6.6}$$

with $d = \frac{1}{2+\sqrt{2}}$, $e = 6 + \sqrt{2}$, $W := I - hdJ$, and $J := \frac{\partial \mathbf{f}(\mathbf{y}_n)}{\partial y}$. The method is effective at crude tolerances due to its FSAL (first same as last) and L-stability [83, Section 3.1], which makes it a good solver for the gradient trajectory. However, the method does not have the desired transitional property.

One step of Rosenbrock iteration (6.6) is equivalent to the equation

$$\frac{\partial \mathbf{f}(\mathbf{x_k})}{\partial \mathbf{y}}\Delta\mathbf{x}_k + \frac{1}{d}\mathbf{f}\left(\mathbf{x}_k + \frac{1}{2}\Delta\mathbf{x}_k^{[1]}\right) = \frac{1}{hd}\Delta\mathbf{x}_k + \frac{\partial \mathbf{f}(\mathbf{x_k})}{\partial \mathbf{y}}\Delta\mathbf{x}_k^{[1]}$$

for $\Delta\mathbf{x}_k$, where $\Delta\mathbf{x}_k^{[1]} := h\mathbf{k}_1$ is a known quantity. Linearizing the second term yields an approximate system

$$\frac{\partial \mathbf{f}(\mathbf{x_k})}{\partial \mathbf{y}}\Delta\mathbf{x}_k + \frac{1}{d}\mathbf{f}\left(\mathbf{x}_k\right) = \frac{1}{dh}\Delta\mathbf{x}_k + (1 - \frac{1}{2d})\frac{\partial \mathbf{f}(\mathbf{x_k})}{\partial \mathbf{y}}\Delta\mathbf{x}_k^{[1]}. \tag{6.7}$$

This expression looks like the inexact Newton form, but the weights do not add up to one. Therefore, the Rosenbrock-based iteration does not have the ability to transition itself into a fast Newton method. Taking advantage of its high efficiency for crude tolerance, however, the method might be used as the initial integrator for steering the flow into the neighborhood of $\mathbf{x}^*$ before switching to other fast converging method.

REFERENCES

[1] P.-A. Absil, R. Mahony, and B. Andrews, *Convergence of the iterates of descent methods for analytic cost functions*, SIAM J. Optim., 16 (2005), pp. 531–547 (electronic).

[2] N. Andrei, *Gradient flow method for nonlinear least squares minimization*, manuscript, Research Institute for Informatics, 2004. http://camo.ici.ro/neculai/nleast.pdf.

[3] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, With contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II, Stanford University Press, Stanford, Calif., 1958.

[4] U. M. Ascher, K. van den Doel, H. Huang, and B. F. Svaiter, *Gradient descent and fast artificial time integration*, M2AN Math. Model. Numer. Anal., 43 (2009), pp. 689–708.

[5] W. Bao and Q. Du, *Computing the ground state solution of bose–einstein condensates by a normalized gradient flow*, SIAM Journal on Scientific Computing, 25 (2004), pp. 1674–1697.

[6] S. Batterson and J. Smillie, *The dynamics of Rayleigh quotient iteration*, SIAM J. Numer. Anal., 26 (1989), pp. 624–636.

[7] P. R. Beesack, *On an existence theorem for complex-valued differential equations*, The American Mathematical Monthly, 65 (1958), pp. pp. 112–115.

[8] W. Behrman, *An efficient gradient flow method for unconstrained optimization*, PhD thesis, Stanford University, 1998.

[9] A. Bhaya and E. Kaszkurewicz, *Control perspectives on numerical algorithms and matrix problems*, vol. 10 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[10] A. Bloch, R. Brockett, and T. Ratiu, *On the geometry of saddle point algorithms*, in Decision and Control, 1992., Proceedings of the 31st IEEE Conference on, 1992, pp. 1482–1487 vol.2.

[11] A. M. Bloch, R. W. Brockett, and T. S. Ratiu, *Completely integrable gradient flows*, Comm. Math. Phys., 147 (1992), pp. 57–74.

[12] A. I. Bobenko and P. Schröder, *Discrete willmore flow*, in Proceedings of the Third Eurographics Symposium on Geometry Processing, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005, Eurographics Association.

[13] J. Bognár, *Indefinite inner product spaces*, Springer-Verlag, New York-Heidelberg, 1974. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 78.

[14] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[15] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young, *Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation*, SIAM J. Sci. Comput., 19 (1998), pp. 246–265 (electronic). Special issue on iterative methods (Copper Mountain, CO, 1996).

[16] M. P. Calvo, A. Iserles, and A. Zanna, *Numerical solution of isospectral flows*, Math. Comp., 66 (1997), pp. 1461–1486.

[17] J. Carr, *Applications of centre manifold theory*, vol. 35 of Applied Mathematical Sciences, Springer-Verlag, New York-Berlin, 1981.

[18] L.-Q. Chen, *Phase-field models for microstructure evolution*, Annual review of materials research, 32 (2002), pp. 113–140.

[19] M. T. Chu, *On the continuous realization of iterative processes*, SIAM Rev., 30 (1988), pp. 375–387.

[20] ———, *Linear algebra algorithms as dynamical systems*, Acta Numer., 17 (2008), pp. 1–86.

[21] ———, *On the dynamics of maximin flows*, preprint, North Carolina State University, 2016.

[22] T. H. Colding, W. P. Minicozzi, II, and E. K. Pedersen, *Mean curvature flow*, Bull. Amer. Math. Soc. (N.S.), 52 (2015), pp. 297–333.

[23] P. Deift, T. Nanda, and C. Tomei, *Ordinary differential equations and the symmetric eigenvalue problem*, SIAM J. Numer. Anal., 20 (1983), pp. 1–22.

[24] K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, vol. 2 of CWI Monographs, North-Holland Publishing Co., Amsterdam, 1984.

[25] L. Del Debbio, A. Patella, and A. Rago, *Space-time symmetries and the Yang-Mills gradient flow*, J. High Energy Phys., (2013), pp. 212, front matter+24.

[26] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[27] M. A. Dritschel and J. Rovnyak, *Operators on indefinite inner product spaces*, in Lectures on operator theory and its applications (Waterloo, ON, 1994), vol. 3 of Fields Inst. Monogr., Amer. Math. Soc., Providence, RI, 1996, pp. 141–232.

[28] M. Droske and M. Rumpf, *A level set formulation for willmore flow*, Interfaces and free boundaries, 6 (2004), pp. 361–378.

[29] L. Faybusovich, *Hamiltonian structure of dynamical systems which solve linear programming problems*, Phys. D, 53 (1991), pp. 217–232.

[30] D. Feijer and F. Paganini, *Stability of primal-dual gradient dynamics and applications to network optimization*, Automatica J. IFAC, 46 (2010), pp. 1974–1981.

[31] S. D. Flåm, *Equilibrium, evolutionary stability and gradient dynamics*, Int. Game Theory Rev., 4 (2002), pp. 357–370.

[32] D. Friedman and R. Abraham, *Bubbles and crashes: gradient dynamics in financial markets*, J. Econom. Dynam. Control, 33 (2009), pp. 922–937.

[33] J. Fritz, *Gradient dynamics of infinite point systems*, Ann. Probab., 15 (1987), pp. 478–514.

[34] B. M. Garay and K. Lee, *Attractors under discretization with variable stepsize*, Discrete Contin. Dyn. Syst., 13 (2005), pp. 827–841.

[35] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and indefinite scalar products*, vol. 8 of Operator Theory: Advances and Applications, Birkhäuser Verlag, Basel, 1983.

[36] R. M. GOODWIN, *The use of gradient dynamics in linear general disequilibrium theory*, in Macrodynamique et déséquilibres, Économica, Paris, 1987, pp. 11–30. With comments in French by Pierre Malgrange.

[37] J. GUCKENHEIMER, *Numerical analysis of dynamical systems*, in Handbook of dynamical systems, Vol. 2, North-Holland, Amsterdam, 2002, pp. 345–390.

[38] E. HAIRER, *Energy-preserving variant of collocation methods*, J. Numer. Anal. Ind. Appl. Math., 5 (2010), pp. 73–84.

[39] E. HAIRER, C. LUBICH, AND G. WANNER, *Geometric numerical integration*, vol. 31 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 2006. Structure-preserving algorithms for ordinary differential equations.

[40] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving ordinary differential equations. I*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1993. Nonstiff problems.

[41] E. HAIRER AND G. WANNER, *Solving ordinary differential equations. II*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1996. Stiff and differential-algebraic problems.

[42] J. K. HALE, *Asymptotic behavior of dissipative systems*, vol. 25 of Mathematical Surveys and Monographs, American Mathematical Society, Providence, RI, 1988.

[43] ———, *Stability and gradient dynamical systems*, Rev. Mat. Complut., 17 (2004), pp. 7–57.

[44] P. HÄNGGI, P. TALKNER, AND M. BORKOVEC, *Reaction-rate theory: fifty years after kramers*, Rev. Modern Physics, 62 (1990), p. 251.

[45] U. HELMKE AND J. B. MOORE, *Optimization and dynamical systems*, Communications and Control Engineering Series, Springer-Verlag London Ltd., London, 1994.

[46] D. J. HIGHAM, *Trust region algorithms and timestep selection*, SIAM J. Numer. Anal., 37 (1999), pp. 194–210 (electronic).

[47] M. W. HIRSCH AND S. SMALE, *Differential equations, dynamical systems, and linear algebra*, Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1974. Pure and Applied Mathematics, Vol. 60.

[48] ———, *On algorithms for solving $f(x) = 0$*, Comm. Pure Appl. Math., 32 (1979), pp. 281–313.

[49] R. G. HOLT AND I. B. SCHWARTZ, *Newton's method as a dynamical system: global convergence and predictability*, Phys. Lett. A, 105 (1984), pp. 327–333.

[50] A. R. HUMPHRIES AND A. M. STUART, *Runge-Kutta methods for dissipative and gradient dynamical systems*, SIAM J. Numer. Anal., 31 (1994), pp. 1452–1485.

[51] E. L. INCE, *Ordinary Differential Equations*, Dover Publications, New York, 1944.

[52] A. ISERLES, *On the discretization of double-bracket flows*, Found. Comput. Math., 2 (2002), pp. 305–329.

[53] A. ISERLES, H. Z. MUNTHE-KAAS, S. P. NØRSETT, AND A. ZANNA, *Lie-group methods*, in Acta numerica, 2000, vol. 9 of Acta Numer., Cambridge Univ. Press, Cambridge, 2000, pp. 215–365.

[54] M. IWASAKI AND Y. NAKAMURA, *On the convergence of a solution of the discrete Lotka-Volterra system*, Inverse Problems, 18 (2002), pp. 1569–1578.

[55] ———, *An application of the discrete Lotka-Volterra system with variable step-size to singular value computation*, Inverse Problems, 20 (2004), pp. 553–563.

[56] C. T. KELLEY AND D. E. KEYES, *Convergence analysis of pseudo-transient continuation*, SIAM J. Numer. Anal., 35 (1998), pp. 508–523 (electronic).

[57] C. T. KELLEY, L.-Z. LIAO, L. QI, M. T. CHU, J. P. REESE, AND C. WINTON, *Projected pseudotransient continuation*, SIAM J. Numer. Anal., 46 (2008), pp. 3071–3083.

[58] D. KINDERLEHRER, L. MONSAINGEON, AND X. XU, *A Wasserstein gradient flow approach to Poisson-Nernst-Planck equations*, ESAIM Control Optim. Calc. Var., 23 (2017), pp. 137–164.

[59] O. KOCH AND C. LUBICH, *Dynamical low-rank approximation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 434–454.

[60] J. D. LAMBERT, *Numerical methods for ordinary differential systems*, John Wiley & Sons, Ltd., Chichester, 1991. The initial value problem.

[61] J. C. LARSEN, *On gradient dynamical system on semi-Riemannian manifolds*, J. Geom. Phys., 6 (1989), pp. 517–535.

[62] C. LUBICH AND I. V. OSELEDETS, *A projector-splitting integrator for dynamical low-rank approximation*, BIT, 54 (2014), pp. 171–188.

[63] X.-L. LUO, C. T. KELLEY, L.-Z. LIAO, AND H. W. TAM, *Combining trust-region techniques and Rosenbrock methods to compute stationary points*, J. Optim. Theory Appl., 140 (2009), pp. 265–286.

[64] J. MAAS, *Gradient flows of the entropy for finite Markov chains*, J. Funct. Anal., 261 (2011), pp. 2250–2292.

[65] R. I. MCLACHLAN, G. R. W. QUISPEL, AND N. ROBIDOUX, *Geometric integration using discrete gradients*, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 1021–1045.

[66] K. MEINTJES AND A. P. MORGAN, *Chemical-equilibrium systems as numerical test problems*, ACM Trans. Math. Soft, 16 (1990), pp. 143–151.

[67] O. MICHAILOVICH, Y. RATHI, AND A. TANNENBAUM, *Image segmentation using active contours driven by the bhattacharyya gradient flow*, IEEE Transactions on Image Processing, 16 (2007), pp. 2787–2801.

[68] A. MIELKE, *A gradient structure for reaction-diffusion systems and for energy-drift-diffusion systems*, Nonlinearity, 24 (2011), pp. 1329–1346.

[69] J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[70] W. A. MULDER AND B. VAN LEER, *Experiments with implicit upwind methods for the Euler equations*, J. Comput. Phys., 59 (1985), pp. 232–246.

[71] K. MÜLLER AND L. D. BROWN, *Location of saddle points and minimum energy paths by a constrained simplex optimization*

procedure, Theoret. Chim. Acta, 53 (1979), pp. 75–93.

[72] H. MUNTHE-KAAS, *Runge-Kutta methods on Lie groups*, BIT, 38 (1998), pp. 92–111.

[73] R. A. NORTON, D. I. MCLAREN, G. R. W. QUISPEL, A. STERN, AND A. ZANNA, *Projection methods and discrete gradient methods for preserving first integrals of ODEs*, Discrete Contin. Dyn. Syst., 35 (2015), pp. 2079–2098.

[74] R. A. NORTON AND G. R. W. QUISPEL, *Discrete gradient methods for preserving a first integral of an ordinary differential equation*, Discrete Contin. Dyn. Syst., 34 (2014), pp. 1147–1170.

[75] F. OTTO AND M. G. REZNIKOFF, *Slow motion of gradient flows*, J. Differential Equations, 237 (2007), pp. 372–420.

[76] M. A. PELETIER, *Energies, gradient flows, and large deviations: a modelling point of view*, Lecture notes, Technische Universiteit Eindhoven, 2012. http://www.win.tue.nl/~mpeletie/Onderwijs/Pisa2011/PeletierLectureNotesPisa2011.pdf.

[77] P. PENZLER, M. RUMPF, AND B. WIRTH, *A phase-field model for compliance shape optimization in nonlinear elasticity*, ESAIM: Control, Optimisation and Calculus of Variations, 18 (2012), pp. 229–258.

[78] D. PINHEIRO, A. A. PINTO, S. Z. XANTHOPOULOS, AND A. N. YANNACOPOULOS, *A projected gradient dynamical system modelling the dynamics of bargaining*, J. Difference Equ. Appl., 19 (2013), pp. 59–95.

[79] H. RATSCHEK AND J. ROKNE, *Experiments using interval analysis for solving a circuit design problem*, J. Global Optim., 3 (1993), pp. 501–518.

[80] R. RIAZA AND P. J. ZUFIRIA, *Discretization of implicit ODEs for singular root-finding problems*, in Proceedings of the 9th International Congress on Computational and Applied Mathematics (Leuven, 2000), vol. 140, 2002, pp. 695–712.

[81] K. SAKURAMA, S.-I. AZUMA, AND T. SUGIE, *Distributed controllers for multi-agent coordination via gradient-flow approach*, IEEE Trans. Automat. Control, 60 (2015), pp. 1471–1485.

[82] M. SHACHAM, *Numerical solution of constrained nonlinear algebraic equations*, Internat. J. Numer. Methods Engrg., 23 (1986), pp. 1455–1481.

[83] L. F. SHAMPINE AND M. W. REICHELT, *The MATLAB ODE suite*, SIAM J. Sci. Comput., 18 (1997), pp. 1–22.

[84] K. SHI, T. XIAO, AND Z. SHENG, *Incentive mechanism design problem based on gradient dynamics*, Kybernetes, 38 (2009), pp. 481–488.

[85] M. SHUB, *Some remarks on dynamical systems and numerical analysis*, in Dynamical systems and partial differential equations (Caracas, 1984), Univ. Simon Bolivar, Caracas, 1986, pp. 69–91.

[86] L. SIMON, *Asymptotics for a class of nonlinear evolution equations, with applications to geometric problems*, Ann. of Math. (2), 118 (1983), pp. 525–571.

[87] S. SMALE, *On gradient dynamical systems*, Ann. of Math. (2), 74 (1961), pp. 199–206.

[88] S. SMALE, *On the mathematical foundations of electrical circuit theory*, J. Differential Geometry, 7 (1972), pp. 193–210.

[89] S. SMALE, *Convergent process of price adjustment and global Newton methods*, in Frontiers of quantitative economics, Vol. IIIA (Invited papers, Econometric Soc., Third World Congress, Toronto, Ont., 1975), North-Holland Publishing Co., Amsterdam, 1977, pp. 191–205. Contributions to Economic Analysis, Vol. 105.

[90] R. STRZODKA, M. DROSKE, AND M. RUMPF, *Image registration by a regularized gradient flow. a streaming implementation in dx9 graphics hardware*, Computing, 73 (2004), pp. 373–389.

[91] A. M. STUART, *Numerical analysis of dynamical systems*, in Acta numerica, 1994, Acta Numer., Cambridge Univ. Press, Cambridge, 1994, pp. 467–572.

[92] A. M. STUART AND A. R. HUMPHRIES, *Dynamical systems and numerical analysis*, vol. 2 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1996.

[93] W. W. SYMES, *The QR algorithm and scattering for the finite nonperiodic Toda lattice*, Phys. D, 4 (1981/82), pp. 275–280.

[94] Y. Z. TSYPKIN, *Adaptation and learning in automatic systems*, Academic Press, New York, 1971. Translated from the Russian by Z. J. Nikolic, Mathematics in Science and Engineering, Vol. 73.

[95] ———, *Foundations of the theory of learning systems*, Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1973. Translated from the Russian by Z. J. Nikolic, Mathematics in Science and Engineering, Vol. 101.

[96] K. VAN DEN DOEL AND U. ASCHER, *The chaotic nature of faster gradient descent methods*, J. Sci. Comput., 51 (2012), pp. 560–581.

[97] B. VAN LEER AND W. A. MULDER, *Relaxation methods for hyperbolic conservation laws*, in Numerical methods for the Euler equations of fluid dynamics (Rocquencourt, 1983), SIAM, Philadelphia, PA, 1985, pp. 312–333.

[98] J. L. VÁZQUEZ, *The porous medium equation: mathematical theory*, Oxford University Press, 2007.

[99] C. T. C. WALL, *A remark on gradient dynamical systems*, Bull. London Math. Soc., 4 (1972), pp. 165–166.

[100] L. WANG, M. T. CHU, AND B. YU, *A computational framework of gradient flows for general linear matrix equations*, Numer. Algorithms, 68 (2015), pp. 121–141.