On the stability of optimization algorithms given by discretizations of the Euler-Lagrange ODE

Rachel Walker[1] Emily Zhang[2]

[1]Central Washington University [2]Massachusetts Institute of Technology

Young Mathematicians Conference, 2019

Background: Problem Setting

Problem Setting

We consider the optimization problem

$$x^* = \arg\min_{x \in \mathbb{R}^d} f(x), \tag{1}$$

where f is a d dimensional strongly convex quadratic function and $\nabla f(x^*) = \vec{0}$.

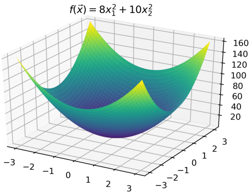


FIGURE – Example strongly convex objective function where $x^* = \vec{0}$

Background: Discrete Gradient-Based Algorithms

Some examples of discrete time algorithms which optimize a convex L-smooth objective function f :

	Discrete Algorithm	Convergence Rate [1]
Gradient Descent	$x_{k+1} = x_k + \delta \nabla f(x_k)$	$O\left(\frac{1}{k}\right)$
Heavy-Ball	$y_{k+1} = x_k + \delta \nabla f(x_k)$ $x_{k+1} = y_{k+1} - \alpha(x_k - x_{k-1})$	$O\left(\frac{1}{k}\right)$
Nesterov's Accelerated Gradient Descent	$y_{k+1} = x_k + \delta \nabla f(x_k)$ $x_{k+1} = y_{k+1} + \frac{k}{k+3}(y_{k+1} - y_k)$	$O\left(\frac{1}{k^2}\right)$

1. Global convergence rate

Background: Modified Equations

Continuous time limits of discrete optimization algorithms for convex functions helps analyze the algorithms. Note that Heavy-Ball assumes an μ-strongly convex f.

	Modified Equation
Gradient Flow	$\dot{X} - \nabla f(X) = 0$
Heavy-Ball	$\ddot{X} + 2\sqrt{\mu}\dot{X} + \nabla f(X) = 0$
Nesterov's Accelerated Gradient Descent	$\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0$

Background: Example

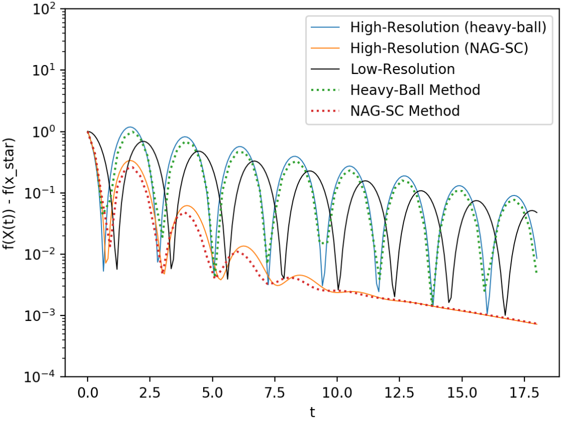


FIGURE – A comparison of discrete optimization methods and their limiting ODEs for $f(x_1, x_2) = 5 \cdot 10^{-3} x_1^2 + x_2^2$

Background: Recent work

Deriving ODEs to describe discrete-time optimization methods:

1. Su et. al. [2016] derive the modified equations and use continuous time Lyapunov function to prove convergence
2. Wibisono et. al. [2016] derived the following ODE from a Lagranian Flow with a parameterized convergence rate

Euler-Lagrange ODE

The Euler-Lagrange ODE

$$\ddot{X}_t + \frac{p+1}{t}\dot{X}_t + Cp^2 t^{p-2}\nabla f(X_t) = 0 \tag{2}$$

has a continuous time convergence rate

$$f(X_t) - f(X^*) \leq O\left(\frac{1}{t^p}\right). \tag{3}$$

Background: Recent work

Wibisono et. al. presented a naive discretization of the Euler-Lagrange ODE :

Naive Discretization (Explicit-Implicit Euler)

$$z_k = z_{k-1} - Cp\delta^p k^{p-1} \nabla f(x_k)$$
$$x_{k+1} = \frac{p}{k} z_k + \frac{k-p}{k} x_k$$

The goal of discretizing the Euler-Lagrange ODE is to achieve the $O\left(\frac{1}{t^p}\right)$ convergence rate, however this does not occur for the naive discretization.

Background: Recent work

Problem

The discrete algorithm oscillates towards the minimize then eventually shoots to infinity, and the reason for this is unclear.

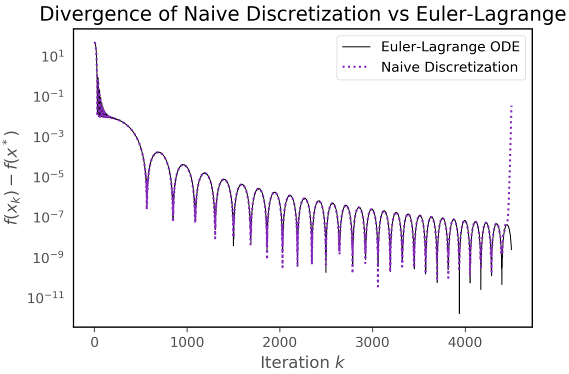


FIGURE – Discrete solution eventually shoots to infinity

Background: Recent work

Recently, work has been done on analyzing discretizations of ODEs as optimization algorithms:

1. Zhang et. al. [2018] show that a direct Runge-Kutta discretization scheme on the Euler-Lagrange ODE achieves acceleration when f is sufficiently smooth
2. Shi et. al. [2019] explore discretization schemes of ODEs as optimization methods

Our Goal

Determine why and when the naive method eventually diverges and attempt to derive an expression to determine when divergence occurs.

Our Approach

We are primarily interested in determining whether a system of update equations given by a certain discretization scheme has converging, diverging, or stable long-term behavior. A system of update equations given by a discretization method is

1. **converging** to the minimizer if the *upper bound* on $|x_k - x^*|$ is decreasing as k increases, [2]
2. **diverging** from the minimizer if the *upper bound* on $|x_k - x^*|$ is increasing as k increases, and
3. **stable** if, for sufficiently large N, $|x_k - x^*| = |x_{k+1} - x^*|$ for all $k > N$.

2. Note that $|x_k - x^*|$ does not have to be a monotonically decreasing sequence in order to be converging.

Our Approach: One-Dimensional Case

We rewrite $f(x)$, a general objective function where x is d-dimensional and A is symmetric, as follows :

$$\begin{aligned} f(x) &= \frac{1}{2}(x - x^*)^T A (x - x^*) \\ &= \frac{1}{2}(x - x^*)^T PDP^T (x - x^*) \\ &= \frac{1}{2}(P^T(x - x^*))^T DP^T (x - x^*) \\ &= \frac{1}{2}\tilde{x}^T D \tilde{x} \end{aligned}$$

where $\tilde{x} := P^T(x - x^*)$, P is the matrix of eigenvectors of A, and D is the diagonal matrix of eigenvalues of A.
Since all dimensions of $\tilde{x}$ update independently of each other, the case where $\tilde{x}$ and x are one-dimensional is without loss of generality.

Our Approach: Stability Function

We consider discretizations of the Euler-Lagrange ODE of the form
$\begin{pmatrix} \tilde{x}_{k+1} \\ z_{k+1} \end{pmatrix} = M_k \begin{pmatrix} \tilde{x}_k \\ z_k \end{pmatrix}$.
We define $R(M_k) := |\lambda_{k,\max}|$ where $\lambda_{k,\max}$ is the eigenvalue of M_k with the largest magnitude, and $R(M_\infty) = \lim_{k\to\infty} R(M_k)$.

Proposition

An optimization algorithm will be

1. converging to the minimizer when $R(M_\infty) < 1$.
2. stable when $R(M_\infty) = 1$.

Proof Idea. We let $u_i := \begin{pmatrix} \tilde{x}_i \\ z_i \end{pmatrix}$. Computing u_k from u_0, we have $u_k = M_{k-1}M_{k-2}\ldots M_1 M_0 u_0$. When all the eigenvalues of M_i have magnitude less than 1, then $\|u_i\| < \|u_{i-1}\|$, and since $\|\tilde{x}_i\| \leq \|u_i\|$, then the upper bound on $\|\tilde{x}_i\|$ is also strictly decreasing.

Our Approach

1. Write the discretization of the Euler-Lagrange ODE in the form

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = M_k \begin{bmatrix} x_k \\ z_k \end{bmatrix}.$$

2. Determine $R(M_\infty)$.
3. Analyze stability conditions for the method.
   - If $R(M_\infty) < 1$, the iterations will be converging to the minimizer.
   - If $R(M_\infty) = 1$, the iterations will be stable.
   - If $R(M_\infty) > 1$, then we determine the largest k for which $R(k) < 1$ in terms of parameters A, p, and δ in order to get a bound on when the iterations exhibit stable behavior.

Euler Methods

Three different Euler discretization schemes are defined as follows for any system of two continuous variables X_t and Z_t such that $\dot{X}_t = f_1(X_t, Z_t)$ and $\dot{Z}_t = f_2(X_t, Z_t)$.
Let δ be the step size and let x_0, z_0 be initialized to the initial value of the ODE that we are trying to discretize.

1. Explicit Euler Method

$$x_{k+1} = x_k + \delta f_1(x_k, z_k)$$
$$z_{k+1} = z_k + \delta f_2(x_k, z_k)$$

2. Implicit Euler Method

$$x_{k+1} = x_k + \delta f_1(x_{k+1}, z_{k+1})$$
$$z_{k+1} = z_k + \delta f_2(x_{k+1}, z_{k+1})$$

3. Explicit-Implicit Euler Method

$$x_{k+1} = x_k + \delta f_1(x_k, z_k)$$
$$z_{k+1} = z_k + \delta f_2(x_{k+1}, z_{k+1})$$

Explicit-Implicit Euler Discretization

The update equations given by the discretization of

$$\dot{X}_t = f_1(X_t, Z_t) = \frac{p}{t}(Z_t - X_t)$$
$$\dot{Z}_t = f_2(X_t, Z_t) = -Cpt^{p-1}\nabla f(X_t).$$

using the explicit-implicit method and the identification $t = \delta k$ are as follows :

$$\begin{aligned} \frac{x_{k+1} - x_k}{\delta} &= \frac{p}{t}(z_k - x_k) \\ \frac{z_k - z_{k-1}}{\delta} &= -Cpt^{p-1}\nabla f(x_k). \end{aligned} \tag{4}$$

This set of update equations eventually diverges after approaching and oscillating around the minimizer, yet it is unknown why this occurs.

Explicit-Implicit Euler Method: Results

Theorem

Let $f(x) : \mathbb{R}^d \to \mathbb{R}$ be an L-smooth function defined as $f(x) = \frac{1}{2}(x - x^*)^T A(x - x^*)$ where $x^* \in \mathbb{R}^d$ is the unique minimizer with $\nabla f(x^*) = \vec{0}$ and A is a positive definite, symmetric $d \times d$ matrix. Let $\delta < \frac{1}{L}$ and $\epsilon = \delta^p$. Then, after we go out enough iterations in the system of update equations given by the naive discretization of the Euler-Lagrange System such that $k > p$ and take $C < \frac{1}{\epsilon L}$, we have the following properties :

1. If $p = 2$, the naive method exhibits stable end behavior.
2. If $p > 2$, the naive method will exhibit stable behavior when

$$k < \left(\frac{4}{CLp^2\epsilon}\right)^{\frac{1}{p-2}}.$$

Explicit-Implicit Euler Method: Proof

Proof Outline.
Step 1. Rewrite the update equations in matrix form :

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} (1-\frac{p}{k})I & \frac{p}{k}I \\ -Cp\epsilon(k+1)^{p-1}(\frac{k-p}{k})A & I - Cp\epsilon(k+1)^{p-1}(\frac{p}{k})A \end{bmatrix}}_{M_k} \begin{bmatrix} x_k \\ z_k \end{bmatrix}. \tag{5}$$

Step 2. Next we determine that

$$R(M_k) = -\frac{-a_k b_k - a_k + 2 - \sqrt{(a_k b_k + a_k - 2)^2 - 4(1-a_k)}}{2} \tag{6}$$

where $a_k = \frac{p}{k}$ and let $b_k = Cp\epsilon(k+1)^{p-1}A$.
Using this, we find the stability function, $R(M_\infty) = \lim_{k\to\infty} R(M_k)$.
Step 3. By analyzing $R(M_\infty)$, we get the result stated in part (a) of the theorem. By simplifying the inequality $R(M_k) \leq 1$, we get the results stated in part (b) of the theorem.

Numerical Results: Explicit Euler

As expected, an explicit Euler discretization becomes unstable quickly

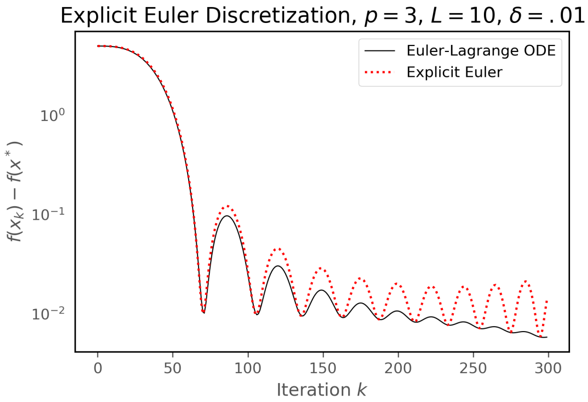


FIGURE – Explicit Euler discretization of the Euler Lagrange quickly diverts from the ODE

Numerical Results: Implicit Euler

As expected, Implicit Euler maintains convergence. Implicit Euler is most useful in the special case where the objective function is in the form $f(\vec{x}) = A\vec{x}$ and A is a positive semi-definite matrix.

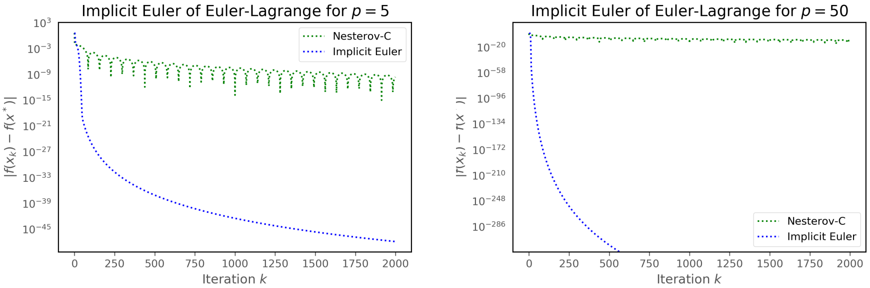


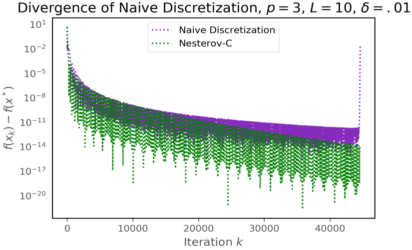
FIGURE – Implicit Euler compared to Nesterov-C

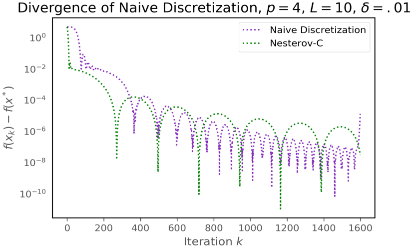
Numerical Results : Explicit-Implicit Euler

We see that the iteration of which we predict the algorithm to converge is accurate.

L	δ	k
10	.01	44,445
10	.001	44,444,445
100	.01	4,445
100	.001	4,444,445

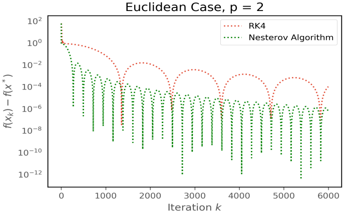
L	δ	k
10	.01	1,582
10	.001	158,113
100	.01	500
100	.001	50,000

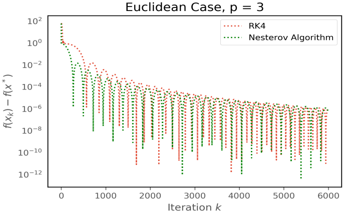


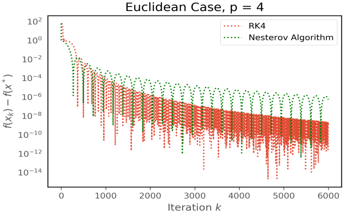


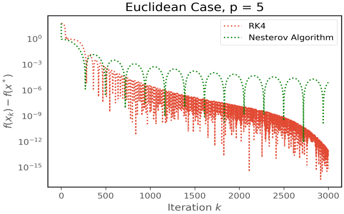
Numerical Results Runge-Kutta

Fourth order explicit Runge-Kutta discretization of the Euler-Lagrange ODE with $L = 10, \delta = .01$.









Discussion and Future Research

1. We showed that the naive method is stable until a certain iteration, however we did not show that it achieve the $O\left(\frac{1}{(\delta k)^p}\right)$ convergence rate. Finding a way to show the convergence rate would be of interest.
2. Runge-Kutta seems to be stable for more iterations than the naive method. It would be of interest to expand our approach to determine when a nth order Runge-Kutta discretization diverges.
3. It would be interesting to apply this method to general convex functions by using linear gradient approximations

Acknowledgements

This research was conducted as part of the 2019 REU program at Georgia Tech and was supported by NSF grant DMS1851843. This project was advised by Professor Rachel A. Kuske and Dr. Andre Wibisono.

References

[1] Armin Eftekhari, Bart Vandereycken, Gilles Vilmart, and Konstantinos C Zygalakis. Explicit stabilised gradient descent for faster strongly convex optimisation. *arXiv preprint arXiv:1805.07199*, 2018.

[2] Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *arXiv preprint arXiv:1810.08907*, 2018.

[3] Bin Shi, Simon S Du, Weijie J Su, and Michael I Jordan. Acceleration via symplectic discretization of high-resolution differential equations. *arXiv preprint arXiv:1902.03694*, 2019.

[4] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[5] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[6] Jingzhao Zhang, Aryan Mokhtari, Suvrit Sra, and Ali Jadbabaie. Direct runge-kutta discretization achieves acceleration. In *Advances in Neural Information Processing Systems*, pages 3900–3909, 2018.