A Conservation Law Method in Optimization

Bin Shi bshi001@cs.fiu.edu
Florida International University
Tao Li taoli@cs.fiu.edu
Florida International University
Sundaraja S. Iyengar iyengar@cis.fiu.edu
Florida International University

Abstract

We propose some algorithms to find local minima in nonconvex optimization and to obtain global minima in some degree from the Newton Second Law without friction. With the key observation of the velocity observable and controllable in the motion, the algorithms simulate the Newton Second Law without friction based on symplectic Euler scheme. From the intuitive analysis of 1-D nonconvex function, we propose the experiments for strongly convex function, non-strongly convex function and nonconvex function in high-dimension.

1 Introduction

From continuous-time limits, we can view the gradient-based method as ODEs. The gradient method is correspondent to

$$\begin{cases} \dot{x} = -\nabla f(x_k) \\ x(0) = x_0, \end{cases} \tag{1}$$

and the momentum method and Nesterov accelerated gradient method are correspondent to

$$\begin{cases} \ddot{x} + \gamma_t \dot{x} + \nabla f(x) = 0 \\ x(0) = x_0, \ \dot{x}(0) = 0, \end{cases} \tag{2}$$

the difference of which are the setting of the friction parameter γ_t.

We investigate the governing equation in a conservation force field in this paper, shown as below,

$$\begin{cases} \ddot{x} = -\nabla f(x) \\ x(0) = x_0, \ \dot{x}(0) = 0. \end{cases} \tag{3}$$

Based on the concept of phase space, the governing equation (3) can be rewritten as

$$\begin{cases} \dot{x} = v \\ \dot{v} = -\nabla f(x) \\ x(0) = x_0, \ v(0) = 0. \end{cases} \tag{4}$$

In this paper, we implement our discrete strategy on (4) with the utility of the observability and controllability of the velocity, or the kinetic energy for two directions as below,

- **Artifically Dissipating Energy Strategy**: To look for local minima in non-convex function or global minima in convex function, the kinetic energy, or the norm of the velocity, is compared with that in the previous step, it will be re-set to zero until it becomes larger no longer.
- **Energy Conservation Strategy**: To look for global minima in non-convex function, an initial larger velocity $v(0) = v_0$ is implemented at the any initial position $x(0) = x_0$. A ball is implemented with (4), the local maximum of the kinetic energy is recorded to discern how many local minima exists along the trajectory. Then implementing the strategy above to find the minimum of all the local minima.

For implementing our thought in practice, we utilize the scheme in the numerical method for Hamiltonian system, the symplectic Euler method. We remark that a more accuracy version is the Störmer-Verlet method for practice.

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).
The contribution of the second author and the third author are literature modification.

2 Symplectic Scheme and Algorithms

The first-order symplectic Euler scheme for (4) is shown as below

$$\begin{cases} x_{k+1} = x_k + hv_{k+1} \\ v_{k+1} = v_k - h\nabla f(x_k). \end{cases} \tag{5}$$

2.1 The Artifically Dissipating Energy Algorithm

Firstly, the artificially dissipating energy algorithm based on (5) is proposed as below.

Algorithm 1 Artifically Dissipating Energy Algorithm

```
 1: Given a starting point $x_0 \in \mathbf{dom}(f)$
 2: Initialize the step length $h$, maxiter, and the velocity variable $v_0 = 0$
 3: Initialize the iterative variable $v_{iter} = v_0$
 4: while $\|\nabla f(x)\| > \epsilon$ and  $k <$ maxiter do
 5:     Compute $v_{iter}$ from the below equation in (5)
 6:     if $\|v_{iter}\| \leq \|v\|$ then
 7:         $v = 0$
 8:     else
 9:         $v = v_{iter}$
10:     end if
11:     Compute $x$ from the above equation in (5)
12:     $x_k = x$;
13:     $f(x_k) = f(x)$;
14:     $k = k + 1$;
15: end while
```

Remark 2.1 *In the actual algorithm 1, the codes in line* 12 *and* 13 *are not need in the while loop in order to speed up the computation.*

2.2 Energy Conservation Algorithm For Detecting Local Minima

Second, the energy conservation algorithm based on (5) is proposed as below.

Algorithm 2 Energy Conservation Algorithm

```
Given a starting point $x_0 \in \mathbf{dom}(f)$
Initialize the step size $h$ and the maxiter
Initialize the velocity $v_0 > 0$ and compute $f(x_0)$
Compute the velocity $x_1$ and $v_1$ from the equation (5), and compute $f(x_1)$
for $k = 1 : n$ do
    Compute $x_{k+1}$ and $v_{k+1}$ from (5)
    Compute $f(x_{k+1})$
    if $\|v_k\| \geq \|v_{k+1}\|$ and $\|v_k\| \geq \|v_{k-1}\|$  then
        Record the position $x_k$
    end if
end for
```

Remark 2.2 *In the algorithm 2, we can set $v_0 > 0$ such that the total energy large enough to climb up some high peak. Same as the algorithm 1, the function value $f(x)$ is not need in the while loop in order to speed up the computation.*

2.2.1 The Simple Example For Illustration

Here, we use the non-convex function for illustration in figure 1 as below,

$$f(x) = \begin{cases} 2\cos(x), & x \in [0, 2\pi] \\ \cos(x) + 1, & x \in [2\pi, 4\pi] \\ 3\cos(x) - 1, & x \in [4\pi, 6\pi] \end{cases} \tag{6}$$

which is the 2nd-order smooth function but not 3rd-order smooth.

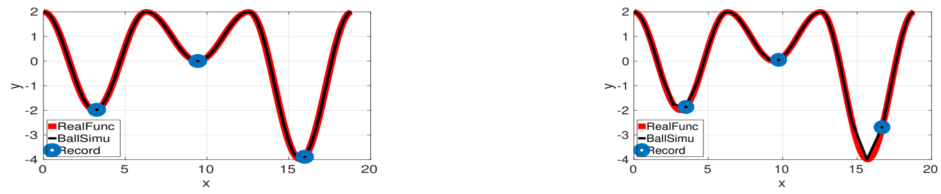

Figure 1: **The Left: the step size $h = 0.1$ with 180 iterative times. The Right: the step size $h = 0.3$ with 61 iterative times.**

3 Experimental Demonstration

In this section, we implement algorithm 1 and algorithm 2 into high-dimension data for comparison with gradient method, momentum method and Nesterov accelerated gradient method.

3.1 Strongly Convex Function

Here, we investigate algorithm 1 on the strongly convex function by the quadratic function as below,

$$f(x) = \frac{1}{2}x^T Ax + b^T x, \tag{7}$$

where A is symmetric and positive-definite matrix. The two cases are shown as below:

(a) The generative matrix A is 500×500 random positive define matrix with eigenvalue from $1e-6$ to 1 with one defined eigenvalue $1e-6$. The generative vector b follows i.i.d. Gaussian distribution with mean 0 and variance 1.

(b) The generative matrix A is the notorious example in Nesterov's book [Nesterov(2013)], i.e.,

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & \end{pmatrix}$$

We implement $\dim(A) = 1000$ and b is zero vector.

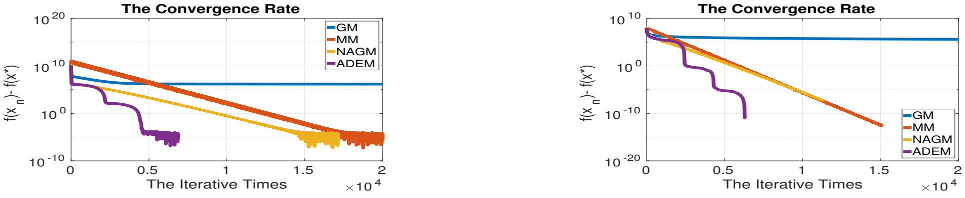

Figure 2: **The Left: the case (a) with the initial point $x_0 = 0$. The Right: the case (b) with the initial point** $x_0 = 1000$

3.2 Non-Strongly Convex Function

Here, we investigate algorithm 1 for the non-strongly convex function by the log-sum-exp function as below,

$$f(x) = \rho \log \left[\sum_{i=1}^{n} \exp \left(\frac{\langle a_i, x\rangle - b_i}{\rho} \right) \right] \tag{8}$$

where A is the $m \times n$ matrix with a_i, $(i = 1, \dots, m)$ the column vector of A and b is the $n \times 1$ vector with component b_i. ρ is the parameter. We show the experiment in (8): the matrix $A = (a_{ij})_{50 \times 200}$ and the vector

$b = (b_i)_{200 \times 1}$ are set by the entry following i.i.d Standard Gaussian distribution for the paramter $\rho = 5$ and $\rho = 10$.

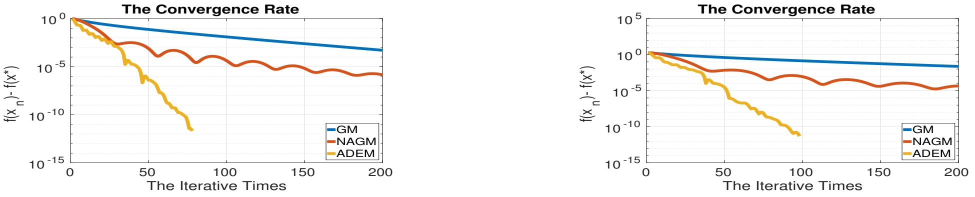

Figure 3: **The convergence rate is shown from the initial point $x_0 = 0$. The Left: $\rho = 5$; The Right: $\rho = 10$.**

3.3 Non-convex Function

For the nonconvex function, we exploit classical test function, Styblinski-Tang function and Shekel function[1], to evaluate characteristics of optimization algorithms from general performance and precision. Firstly, we investigate Styblinski-Tang function, i.e.

$$f(x) = \frac{1}{2} \sum_{i=1}^{d} \left(x_i^4 - 16x_i^2 + 5x_i\right) \tag{9}$$

to demonstrate the general performance to track the number of local minima by algorihtm 2 and then find the local minima by algorithm 1.

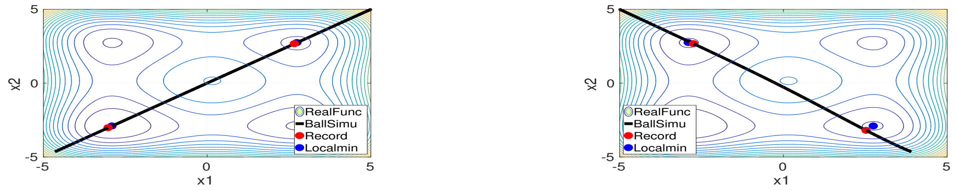

Figure 4: **Detecting the number of the local minima of 2-D Styblinski-Tang function by algorithm 2 with step length $h = 0.01$. The red points are recorded by algorithm 2 and the blue point are the local minima by algorithm 1. The Left: The Initial Position $(5, 5)$; The Right: The Initial Position $(-5, 5)$.**

Secondly, we demonstrate the numerical experiment on more complex Shekel function. Case $m = 10$, the global minima at $x^\star = (4, 4, 4, 4)$ is $f(x^\star) = -10.5364$. From the position $(10, 10, 10, 10)$, the experimental result with the step length $h = 0.01$ and the iterative times 3000 is shown as below

Detect Position (Algorithm 2)

$$\begin{pmatrix} 7.9977 & 5.9827 & 4.0225 & 2.7268 & 6.1849 & 6.2831 & 6.3929 \\ 7.9942 & 6.0007 & 3.8676 & 7.3588 & 6.0601 & 3.2421 & 1.9394 \\ 7.9977 & 5.9827 & 4.0225 & 2.7268 & 6.1849 & 6.2831 & 6.3929 \\ 7.9942 & 6.0007 & 3.8676 & 7.3588 & 6.0601 & 3.2421 & 1.9394 \end{pmatrix}$$

Detect value

$$\begin{pmatrix} -5.1741 & -2.8676 & -7.9230 & -1.5442 & -2.4650 & -1.3703 & -1.7895 \end{pmatrix}$$

Final position (Algorithm 1)

$$\begin{pmatrix} 7.9995 & 5.9990 & 4.0007 & 3.0009 & 5.9990 & 6.8999 & 5.9919 \\ 7.9994 & 5.9965 & 3.9995 & 7.0004 & 5.9965 & 3.4916 & 2.0224 \\ 7.9995 & 5.9990 & 4.0007 & 3.0009 & 5.9990 & 6.8999 & 5.9919 \\ 7.9994 & 5.9965 & 3.9995 & 7.0004 & 5.9965 & 3.4916 & 2.0224 \end{pmatrix}$$

Final value

$$\begin{pmatrix} -5.1756 & -2.8712 & -10.5364 & -2.7903 & -2.8712 & -2.3697 & -2.6085 \end{pmatrix}$$

[1] https://www.sfu.ca/ ssurjano/index.html

References

[Anandkumar and Ge(2016)] A. Anandkumar and R. Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on Learning Theory*, pages 81–102, 2016.

[Beck and Teboulle(2009)] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[Bubeck et al.(2015)] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Ge et al.(2015)Ge, Huang, Jin, and Yuan] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points?online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

[Hardt et al.(2016)Hardt, Ma, and Recht] M. Hardt, T. Ma, and B. Recht. Gradient descent learns linear dynamical systems. *arXiv preprint arXiv:1609.05191*, 2016.

[Lee et al.(2016)Lee, Simchowitz, Jordan, and Recht] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pages 1246–1257, 2016.

[Lessard et al.(2016)Lessard, Recht, and Packard] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[Nesterov(1983)] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[Nesterov(2013)] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Nesterov and Nemirovsky(1988)] Y. Nesterov and A. Nemirovsky. A general approach to polynomial-time algorithms design for convex programming. Technical report, Technical report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR, 1988.

[O'donoghue and Candes(2015)] B. O'donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

[Polyak(1964)] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[Rumelhart et al.(1988)Rumelhart, Hinton, Williams, et al.] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[Su et al.(2014)Su, Boyd, and Candes] W. Su, S. Boyd, and E. 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.

[Sutskever et al.(2013)Sutskever, Martens, Dahl, and Hinton] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[Wibisono et al.(2016)Wibisono, Wilson, and Jordan] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, page 201614734, 2016.

[Wilson et al.(2016)Wilson, Recht, and Jordan] A. C. Wilson, B. Recht, and M. I. Jordan. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

4 Related Work

The history of gradient method for convex optimization can be back to the time of Euler and Lagrange. However, since it is relatively cheaper to only calculate for first-order information, this simplest and earliest method is still active in machine learning and nonconvex optimization, such as the recent work [Ge et al.(2015)Ge, Huang, Jin, and Yuan, Anandkumar and Ge(2016), Lee et al.(2016)Lee, Simchowitz, Jordan, and Recht, Hardt et al.(2016)Hardt, Ma, and Recht]. The natural speedup algorithms are the momentum method first proposed in [Polyak(1964)] and Nesterov accelerated gradient method first proposed in [Nesterov(1983)] and an improved version [Nesterov and Nemirovsky(1988)]. A acceleration algorithm similar as Nesterov accelerated gradient method, named as FISTA, is designed to solve composition problems [Beck and Teboulle(2009)]. A related comprehensive work is proposed in [Bubeck et al.(2015)].

The original momentum method, named as Polyak heavy ball method, is from the view of ODE in [Polyak(1964)], which contains extremely rich physical intuitive ideas and mathematical theory. An extremely important work in application on machine learning is the backpropagation learning with momentum [Rumelhart et al.(1988)Rumelhart, Hinton, Williams, et al.]. Based on the thought of ODE, a lot of understanding and application on the momentum method and Nesterov accelerated gradient methods have been proposed. In [Sutskever et al.(2013)Sutskever, Martens, Dahl, and Hinton], a well-designed random initialization with momentum parameter algorithm is proposed to train both DNNs and RNNs. A seminal deep insight from ODE to understand the intuition behind Nesterov scheme is proposed in [Su et al.(2014)Su, Boyd, and Candes]. The understanding for momentum method based on the variation perspective is proposed on [Wibisono et al.(2016)Wibisono, Wilson, and Jordan], and the understanding from Lyaponuv analysis is proposed in [Wilson et al.(2016)Wilson, Recht, and Jordan]. From the stability theorem of ODE, the gradient method always converges to local minima in the sense of almost everywhere is proposed in [Lee et al.(2016)Lee, Simchowitz, Jordan, and Recht]. Analyzing and designing iterative optimization algorithms built on integral quadratic constraints from robust control theory is proposed in [Lessard et al.(2016)Lessard, Recht, and Packard].

Actually the “high momentum” phenomenon has been firstly observed in [O’donoghue and Candes(2015)] for a restarting adaptive accelerating algorithm, and also the restarting scheme is proposed by [Su et al.(2014)Su, Boyd, and Candes]. However, both works above utilize restarting scheme for an auxiliary tool to accelerate the algorithm based on friction. With the concept of phase space in mechanics, we observe that the kinetic energy, or velocity, is controllable and utilizable parameter to find the local minima. Without friction term, we can still find the local minima only by the velocity parameter. Based on this view, the algorithm is proposed very easy to practice and propose the theoretical analysis. Meanwhile, the thought can be generalized to nonconvex optimization to detect local minima along the trajectory of the particle.