Introduction to Accelerated Gradient Method[⋆]

Yuxuan Zhou,

Abstract: Accelerated Gradient Method was first established by Yurii Nesterov in 1983 Nesterov [1983]. It can be seen as a modification of the momentum method based on the gradient descent method, which was proved to obtain the best performance among first-order algorithms applied to minimize smooth convex functions.

1. INTRODUCTION

Accelerated Gradient Method was first established by Yurii Nesterov in 1983 Nesterov [1983]. It can be seen as a modification of the momentum method based on the gradient descent method, which was proved to obtain the best performance among first-order algorithms applied to minimize smooth convex functions. It's widely adopted in applications like machine-learning and signal-processing, due to the need of solving large scale problems, where second-order methods are time-consuming and considered as unsuitable.

1.1 Uper and lower bound

Definition 1(Lipschitz continuity). *A function $f : R^n \longrightarrow R^m$ is Lipschitz continuous with Lipschitz constant L if for any x,y $\in R^n$*

$$||f(y) - f(x)||_2 \leq L||y - x||_2 \tag{1}$$

We will need this property for the gradient of functions. The following proposition shows that the functions that have Lipschitz-continuous gradients are upper bounded by a quadratic function.

Proposition 1*(Quadratic upper bound). If the gradient of a function $f : R^n \longrightarrow R$ is Lipschitz continuous with Lipschitz constant L,*

$$||\nabla f(y) - \nabla f(x)||_2 \leq L||y - x||_2 \tag{2}$$

then for any $x, y \in R^n$

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}||y - x||_2^2 \tag{3}$$

Definition 2*(Strong convexity). A function $f : R^n$ is S-strongly convex if for any $x, y \in R^n$*

$$f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2}||y - x||^2 \leq f(y) \tag{4}$$

This means that the function has a quadratic lower bound at any point.

From these two properties it could be proved Boyd and Vandenberghe [2004]

$$\mu I \leq \nabla^2 f(x) \leq LI \tag{5}$$

And this means

$$\sigma_{min}(\nabla^2 f(x) \geq \mu \tag{6}$$

$$\sigma_{max}(\nabla^2 f(x)) \leq L \tag{7}$$

with σ as the singular value of the Hessian of $f(x)$.

Now the definition of condition number should be introduced let A be a m n matrix, the condition number κ is the ratio of max singular value and min singular value of A

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}} \tag{8}$$

In the simplest case, where A is a two dimensional square matrix with full rank

$$\kappa = \frac{max}{min} \tag{9}$$

And we will see in the next sections that the condition number of the hessian matrix has strong effect on the convergence rate of all the first order methods.

2. *GRADIENT DESCENT METHODS*

2.1 Algorithm

```
given a starting point $x_0 \in$ dom $f$
repeat
step :$\Delta x_k = -\nabla f(x_k)$.
Line search. Choose a step size $t$.
Update. $x_{k+1} = x_k - t\nabla f(x)$
until Stopping criterion is satisfied.
```

2.2 Convergence Analysis

At first, the term eigenvector should be reviewed. They are vectors, whose direction after a matrix transformation doesn't change

$$Bx = \lambda x \tag{10}$$

And if B is symmetric, then there exist n independent eigenvectors, which can choose to be the basis of the whole n dimensional space. Thus any vector can be expressed as a linear combination of the n eigenvectors.

And at each iteration the matrix B is applied to vector x_i. Take a two dimensional vector as an example

[⋆] Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany. *http://www.ist.uni-stuttgart.de*

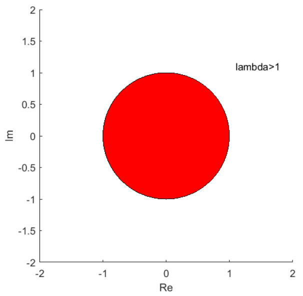

Fig. 1. Stable region of eigenvalues

let v_1, v_2 be the independent eigenvectors of the matrix B, and x be a two dimensional vector

$$x_{k+1} = Bx_k \tag{11}$$

$$x_0 = av_1 + bv_2 \tag{12}$$

if

$$||\lambda_1|| < 1and||\lambda_2|| > 1 \tag{13}$$

after k iteration

$$x_k = B^k x_0 \tag{14}$$

$$B^k = a||\lambda_1||^k v_1 + b||\lambda_2||^k v_2 \tag{15}$$

if $k \longrightarrow \infty$, then the first term converges to zero, and the second term diverges. This illustrates the influence of eigenvalues on the convergence rate of the iteration method roughly.

2.3 Convergence rate

We assume that f is L-Lipschitz continuous and strong convex. There exists a point x^* at which f achieves a finite minimum. If we set the step size of gradient descent to $\alpha_k = \alpha <= 1/L$ for every iteration,

$$||x_{k+1} - x^*||_2 = ||x_k - \alpha\nabla f(x) - (x^* - \alpha\nabla f(x^*))||_2 \tag{16}$$

$$= ||x_k - x^* - \alpha(\nabla f(x_k) - \nabla f(x^*))||_2 \tag{17}$$

By the mean value theorem,

$$\nabla f(x_k) - \nabla f(x^*) = \nabla^2 f(z_k)(x_k - x^*) \tag{18}$$

for some z_k on the line segment between x_k and x^*.

$$||x_{k+1} - x^*||_2 = ||(I - \alpha\nabla^2 f(z))(x_k - x^*)||_2 \tag{19}$$

$$\leq ||I - \alpha\nabla^2 f(z)||_2 ||x_k - x^*||_2 \tag{20}$$

Because $\nabla^2 f(x)$ is between μ and L,

$$||x_{k+1} - x^*||_2 \leq max(|1 - \alpha\mu|, |1 - \alpha L|)||x_k - x^*|| \tag{21}$$

Thus,

$$||x_{k+1} - x^*||_2 \leq max(|1 - \alpha\mu|, |1 - \alpha L|)^k ||x_0 - x^*|| \tag{22}$$

The step size $\frac{2}{L+\mu}$ minimizes the right hand side and using this step size we get,

$$||x_{k+1} - x^*||_2 \leq (\frac{\kappa - 1}{\kappa + 1})^k ||x_0 - x^*|| \tag{23}$$

And $\frac{\kappa-1}{\kappa+1}$ is roughly $1 - 1/\kappa$, when κ is large. Only $\mathcal{O}(\kappa log(1/\epsilon))$ iterations are need to get a ϵ approximation.

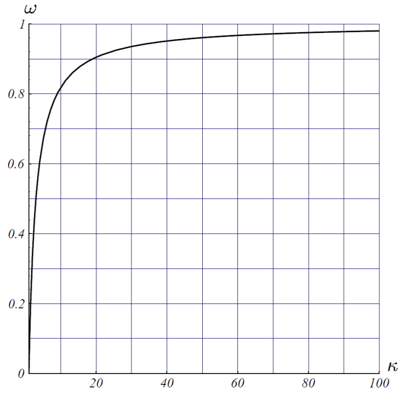

Fig. 2. Convergence rate of Steepest Descent decreases as the condition number increases

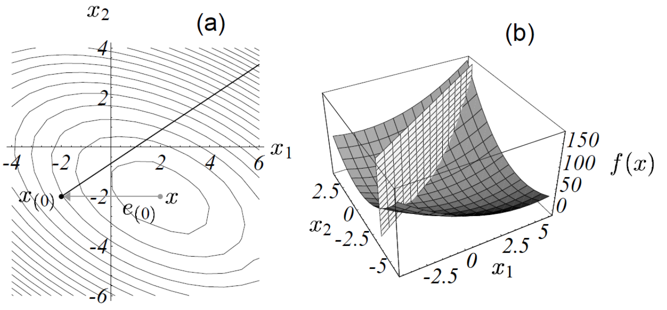

Fig. 3. An example of Positive definite quadratic function and its level contour Shewchuk et al. [1994]

Figure 2 shows the relationship between the convergence of Steepest Descent and the condition number of the matrix, where $w = \frac{\kappa-1}{\kappa+1}$

Taking the quadratic function as an example,

$$f(x) = x^T Ax + bx + c \tag{24}$$

And note that the matrix A has always an equivalent symmetric matrix $S = \frac{1}{2}(A^T + A)$ in the quadratic form,

$$x^T Ax = \frac{1}{2}x^T(A^T + A)x = x^T Sx \tag{25}$$

So we can simply choose a A that is symmetric, then the gradient and hessian are respectively:

$$\nabla f(x) = Ax + b \tag{26}$$

$$\nabla^2 f(x) = A \tag{27}$$

And the level contours of a quadratic function are exactly ellipsoids, whose semi-axes are determined by the eigenvalue and eigenvectors of the hessian matrix A.

Figure 3 illustrates how the gradient descent method proceeds in each step. The choice of step is actually restricted to the intersection of the vertical plane and the paraboloid

And for each step, the direction of the step is always restricted to the line, that is vertical to the tangent of the ellipsoid level contour. Therefor, the convergence rate is determined by the condition number and initial position of the iteration.

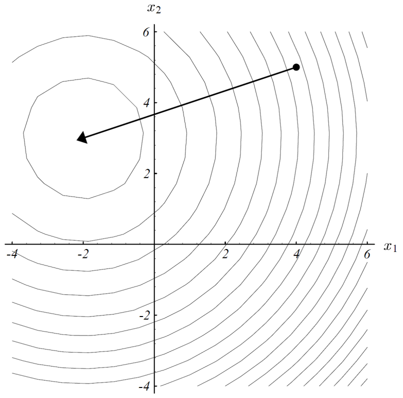

Fig. 4. if all eigenvalues are equal,then the convergence route is just a straight line and it only needs one step by steepest descent gradient method Shewchuk et al. [1994]

Fig. 5. if condition number is large, but the start point is near the axes of the ellipsoid level contour, then it also converges fast Shewchuk et al. [1994]

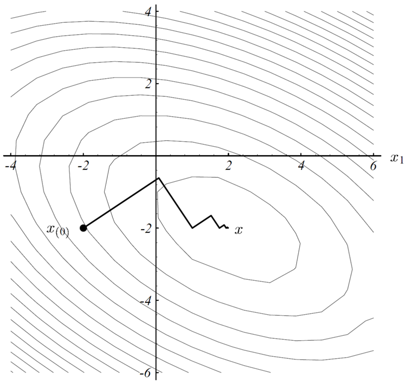

Fig. 6. Zigzag phenomenon at the gradient descent method Shewchuk et al. [1994]

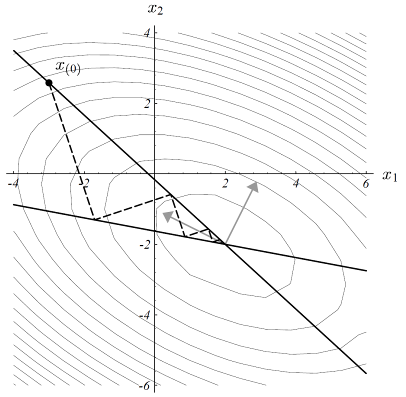

Fig. 7. Ill conditioned case Shewchuk et al. [1994]

In order to minimize f with respect to α, the derivative must equals zero. And there's an intuitive interpretation of this procedure.f will be minimized restricted to the search line only at the point, where the search line is tangent to the smallest ellipsoid level contour. Thus the best α is chosen at the point, where the gradient is orthogonal to the search line.

And this is exactly the reason for the zigzag phenomenon occurring at the gradient descent method and its variants. Figure 3 illustrates this intuitively.

3. *MOMENTUM METHOD*

3.1 convergence rate

In order to reduces the bad performance at ill-conditioned cases(condition number is large), a momentum term has been included. It is widely known as heavy ball method and found out to increase the convergence rate dramatically.Intuitively, the momentum helps to escape from the local slope of the function.

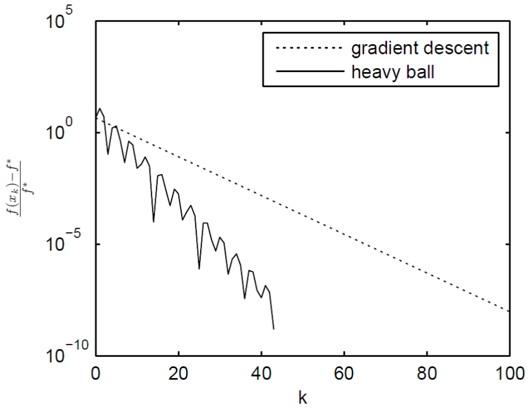

Fig. 8. Convergence rate of Steepest Descent and heavy ball function values on as strongly convex function. The heavy ball method converges faster, though it's not monotonically decreasing.

Unlike gradient descent, the heavy ball method is not a descent method. And it was proved to be only locally instead of globally asymptotically stable.

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \beta_k(x_k - x_{k-1}) \tag{28}$$

The norm of the difference in each step can be transformed in a matrix form:

$$\left\| \begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix} \right\|_2 = \tag{29}$$

$$\left\| \begin{bmatrix} x_k + \beta(x_k - x_{k-1}) - x^* \\ x_k - x^* \end{bmatrix} - \alpha \begin{bmatrix} \nabla f(x) \\ 0 \end{bmatrix} \right\|_2 \tag{30}$$

By the mean value theorem,

$$\nabla f(x_k) - \nabla f(x^*) = \nabla^2 f(z_k)(x_k - x^*) \tag{31}$$

for some z_k on the line segment between x_k and x^*.

$$\left\|\begin{bmatrix} x_{k+1}-x^* \\ x_k - x^*\end{bmatrix}\right\|_2 = \tag{32}$$

$$\left\|\begin{bmatrix}(1+\beta)I & -\beta I\\ I & 0\end{bmatrix}\begin{bmatrix}x_k-x^*\\x_{k-1}-x^*\end{bmatrix} - \alpha\begin{bmatrix}\nabla^2 f(z_k)(x_k-x^*)\\0\end{bmatrix}\right\|_2 \tag{33}$$

$$= \left\|\begin{bmatrix}(1+\beta)I-\alpha\nabla^2 f(z_k) & -\beta I\\ I & 0\end{bmatrix}\begin{bmatrix}x_k-x^*\\x_{k-1}-x^*\end{bmatrix}\right\| \le \tag{34}$$

$$\left\|\begin{bmatrix}(1+\beta)I-\alpha\nabla^2 f(z_k) & -\beta I\\ I & 0\end{bmatrix}\right\| \left\|\begin{bmatrix}x_k-x^*\\x_{k-1}-x^*\end{bmatrix}\right\| \tag{35}$$

3.2 Analogy to second order ODE in continuous time

The relation between the hessian matrix and the convergence rate can be seen from the equations above clearly. But in order to get an intuitive understanding of how the momentum term accelerates the basic gradient method, let's do a comparison with the second order ODE in continuous time.

3.3 referring to second order ODE in continuous time

The equation with a momentum term can be simplified as

$$\Delta x_{k+1} = -\alpha_k \nabla f(x_k) + \beta_k \Delta x_k \tag{36}$$

Let's show this equation is indeed a discretization of the following second order ODE

$$\beta\ddot{x} + a\dot{x} + b\nabla f(x) = 0 \tag{37}$$

First order backward difference quotient:

$$\dot{x} = \frac{x_k - x_{k-1}}{h} \tag{38}$$

Second order central difference quotient:

$$\ddot{x} = \frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} \tag{39}$$

with $h > 0$ and $x = x_k$ in the continuous-time system as approximation. This yields

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} = -b\nabla f(x) - a\frac{x_k - x_{k-1}}{h} \tag{40}$$

and can be transformed into:

$$x_{k+1} = x_k - ah^2\nabla f(x) + (1-bh)(x_k - x_{k-1}) \tag{41}$$

Referring to the conclusion from ODE from Qian's work Qian [1999], there's fact that the critical damping condition allows the fastest damping to its equilibrium position. That means according to the value of the momentum term, the fastest possible convergence can be reached.

But because the choice of optimum momentum term relies on the preknowledge of the condition number(quotient of the upper and lower bound derived from Lipschitz continuouty and strong convexity), and these two constants can only be estimated. So in the application are usually not monotonically decreasing. The maximum convergence rate is $\mathcal{O}(log(\frac{1}{\epsilon}))$.

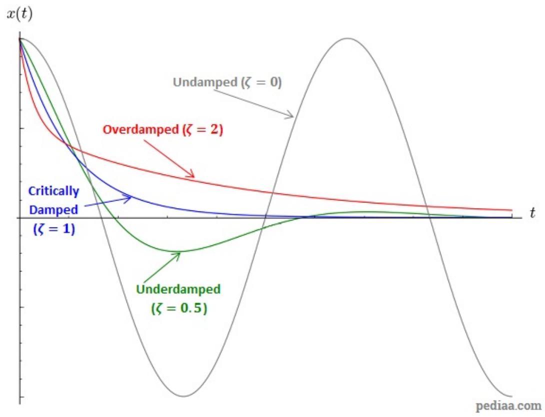

Fig. 9. Fastest damping when critically damped

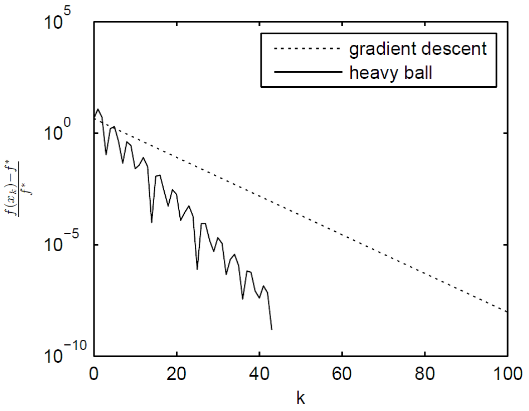

Fig. 10. Convergence rate of Steepest Descent and heavy ball function values on as strongly convex function. The heavy ball method converges faster, though it's not monotonically decreasing.

Problem of this heavy ball method is that the condition number is just a global parameter. There are possibly local regions with either much higher or lower condition number. So despite the optimal choice of momentum value, the oscillating behaviour associated with high momentum emerges still in these local regions.

4. *ACCELERATED GRADIENT METHOD*

It's based on the heavy ball method, but the gradient is evaluated at an intermediate point y_{k+1} instead of x_k.

It was proved to obtain an ϵ - optimal solution in $\mathcal{O}(\sqrt{1/\epsilon})$ iterations. And no method for solving (P) based on accumulated first order information can achieve an iteration -complexity bound of less than $\mathcal{O}(1/\sqrt{\epsilon})$.Nesterov [1983] Moreover, it was proved to be not only locally but also globally asymptotically stable.

$$y_{k+1} = x_k + \beta_k(x_k - x_{k-1}) \tag{42}$$

$$x_{k+1} = y_{k+1} + -t_k\nabla f(y_k) \tag{43}$$

γ_k solves

$$\gamma_k^2 = (1-\gamma_k)\gamma_{k-1}^2 + q\gamma_k \tag{44}$$

$$\beta_k = \gamma_{k-1}(1-\gamma_{k-1})/(\gamma_{k-1}^2+\gamma_k) \tag{45}$$

It was proved that the optimal choice of q equals the reciprocal of the condition number, that is to say, $q^* = u/L$. When we overestimate or underestimate q^* and hence the momentum, the osicillating behaviour or the monotonic but slower convergence would occur.

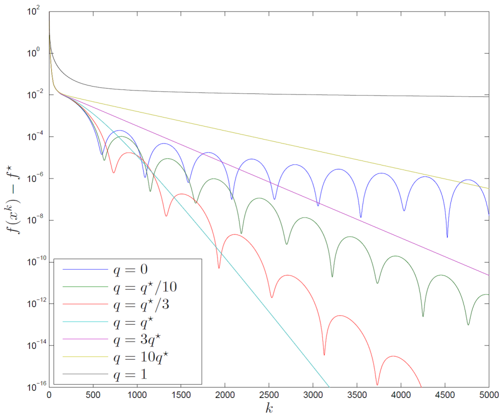

Fig. 11. Convergence of Accelerated descent method with different estimates of q Odonoghue and Candes [2015]

In practice the coefficient q is chosen to be zero, because the it requires a lot of efforts to estimate the condition number. In this case, the equation becomes

$$y_{k+1} = x_k + \beta_k(x_k - x_{k-1}) \tag{46}$$

$$x_{k+1} = y_{k+1} + -t_k\nabla f(y_k) \tag{47}$$

$$\gamma_k = \frac{1}{2}(4\gamma_{k-1}^2 + \gamma_{k-1}^4)^{\frac{1}{2}} - \gamma_{k-1}^2 \tag{48}$$

$$\beta_k =_k (1-\gamma_{k-1}^{-1}) \tag{49}$$

In addition, the momentum term grows from one iteration to the next according to the equations above, whereas it remains unchanged in the heavy ball method.

5. *IMPLEMENT AND COMPARISON*

In order to compare the efficiency and robustness of these three methods, i have implemented them in the Matlab enviroment. Different fixed step sizes were chosen, and the results are as in following figures.

And from figure 13, the effect of Momentum could be explained clearly. It adds an extra inertia to the gradient descent at each step, thus increasing the magnitude of each step and causing the overshooting. Additionally, as the step approaches the optimal solution, this effect degrades, that is to say, a higher momentum is needed to gain the overshooting.

So the superiority of ACG is now also understandable. It just starts with a zero momentum(when β is initialized to be 1), and raises the value as the iteration grows.

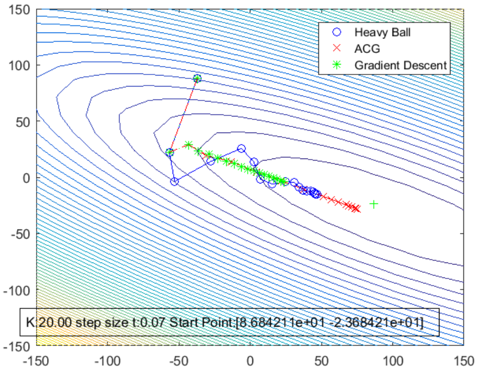

Fig. 12. With a small enough step size, Nesterov's method converges fastest, followed by Heavy Ball method and Gradient Desent.

Notably, ACG is more sensitive to the step size. When the step size is larger, it begins to diverge, whereas the other two methods stays stable.

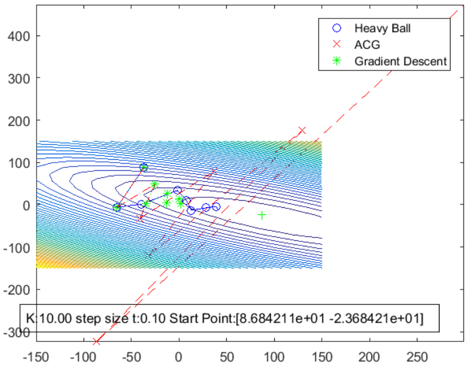

Fig. 13. ACG diverges first when the step size is bigger than some value

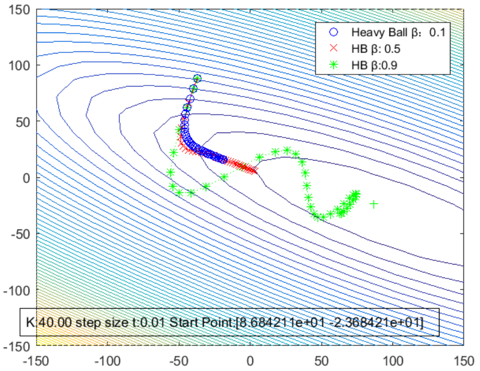

Fig. 14. A higher momentum guarantees a faster convergence when the step size is sufficient small

We study the influence of the momentum term. As the momentum grows, the convergence rate increases, while at the same time the overshooting phenomenon is also enhanced. When the step size becomes larger, a higher momentum value leads to a divergence.

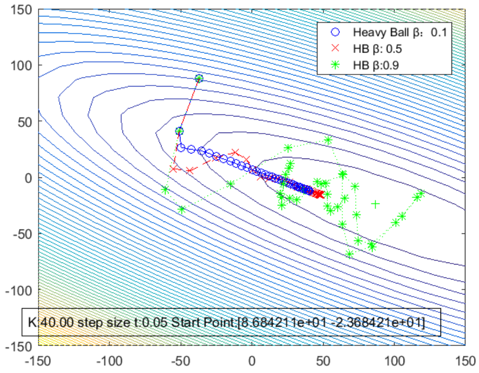

Fig. 15. A lower momentum also supplies a fast convergence rate with larger step sizes

And there's always a trade-off between step size t and momentum β. When β is larger, the step size must be smaller. So it's hard to set the optimal parameter without knowledge of L and μ.

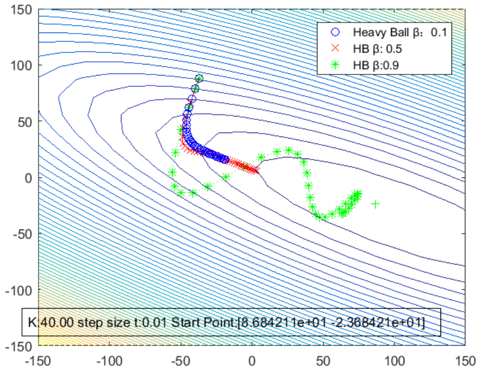

Fig. 16. Heavy ball method with various β

So the sensitivity of ACG could also be explained according to the former two diagrams, because the momentum term in ACG grows dramatically after a few steps, therefore it's easier to overshoot at large step size. Nesterov's method constructs a function to adjust the momentum term automatically.

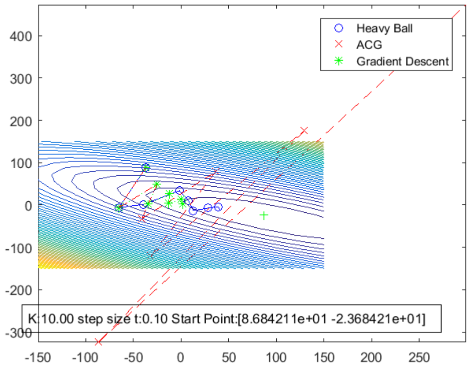

Fig. 17. β is increases from zero and asymptotically to 1

So I suppose that the function to calculate β can be arbitrarily chosen, as long as it increases from zero and is asymptotic to 1. Implementing the simplest function $\beta = 1 - 1/k$, where k is just the number of iteration steps.

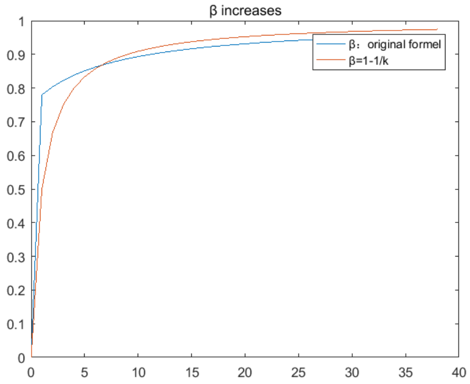

Fig. 18. changed β function

and the result looks good,

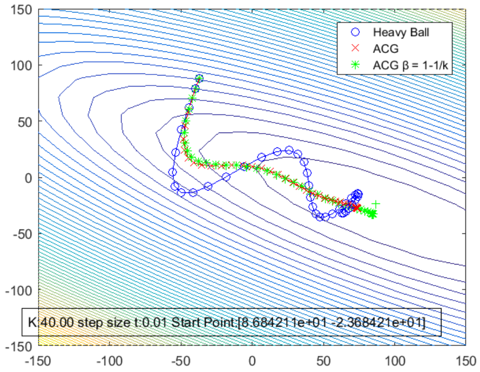

Fig. 19. changed β function

Now we study the influence of calculating the gradient at the intermediate point. It suppresses the overshooting of the heavy ball method and may be the key reason for the convergence.

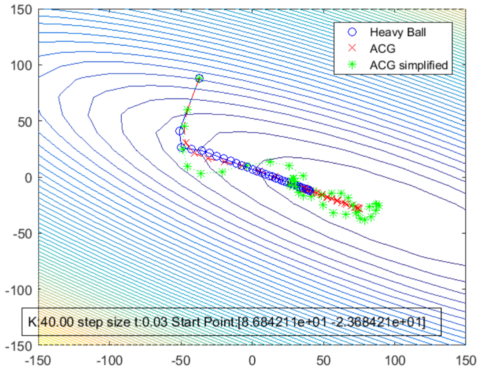

Fig. 20. Calculating the gradient at the intermediate point seems to guarantee the convergence

REFERENCES

Stephen Boyd and Lieven Vandenberghe. Convex optimization. *Cambridge university press, 2004.*

Yurii Nesterov. A method of solving a convex programming problem with convergence rate O (1/k2), *volume 27. 1983.*

Brendan Odonoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics*, 15(3):715–732, 2015.*

Ning Qian. On the momentum term in gradient descent learning algorithms. Neural networks*, 12(1):145–151, 1999.*

Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.