Gradient descent method and conjugate gradient method to solve linear equation

Wang Yitian

* *220171778, email:wyt141@gmail.com*

Abstract: As homework for computational physics in SEU, this article is a simple realization of gradient descent method and conjugate gradient method in solving linear equation $Ax = b$.

Keywords: gradient descent, conjugate gradient, linear equation

1. INTRODUCTION

Gradient method, such as gradient descent and conjugate gradient, is famous for solving constrained and unconstrained optimization problems. Such method is thus applied in vast areas like neural network training, energy minimization problems and so on. However, the mathematics for gradient methods derivation is like bizarre torture inflicted on students and tedious. It takes painful effort to learn the gorgeous and meaningful maths, especially for people like me, who is not good at linear algebra. In contrast, the image for such method is intuitive and interesting, thus this article focuses on solving two-dimensional linear equations to give readers an intuition of the algorithms.

2. WHY LINEAR EQUATION CAN BE SOLVED BY GRADIENT METHOD

This part gives the painless math to understand why we can use gradient method to solve linear equation $Ax = b$, other detailed derivation can easily be found by *Google*, *Bing* or the reference.

Firstly, we can construct a quadratic form function

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

2.1 Symmetric A

Then the gradient of (1) can be solved like:

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b \tag{2}$$

Therefore, if A is symmetric, then $A^T = A$, which means

$$f'(x) = Ax - b \tag{3}$$

Finally, we set (3) to be zero, $Ax = b$ is thus be achieved, which means solving the (1) optimization problem is equal to solving linear equation.

2.2 Positive-definite A

To understand the trick, one can consider the one-dimensional situation (see Fig. 1). It's easy to figure out

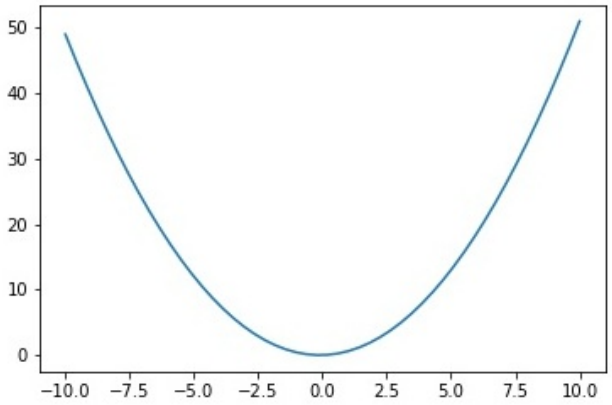

Fig. 1. An example of one-dimensional quadratic form.

that for $f(x) = \frac{1}{2}ax^2 - bx + c$, the solution of $ax = b$ is also the minimum point of f(x).

Of course, in order to make sure the solution is the minimum point, a needs to be positive here, thus A also needs to be positive-definite, which means for every nonzero vector x,

$$x^T Ax > 0 \tag{4}$$

This condition actually makes sure x is the global minimum solution. In addition, negative-definite also works, since the result of negating the negative-definite matrix is positive-definite.

3. RESULTS

This part gives the result of solving a simple two-dimensional quadratic form function (1), where

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix} \quad c = 0 \tag{5}$$

Since A is a symmetric and positive-definite matrix, it can be seen from Fig. 2 and Fig. 3 that the minimum point of f(x) is the global minimum.

3.1 Gradient descent method

One hard thing about gradient descent method is adjusting step size α, a big α can cause the loss of convergence while

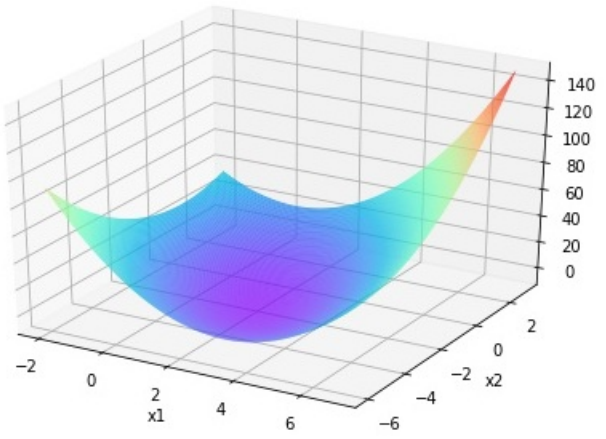

Fig. 2. The example of two-dimensional quadratic form f(x).

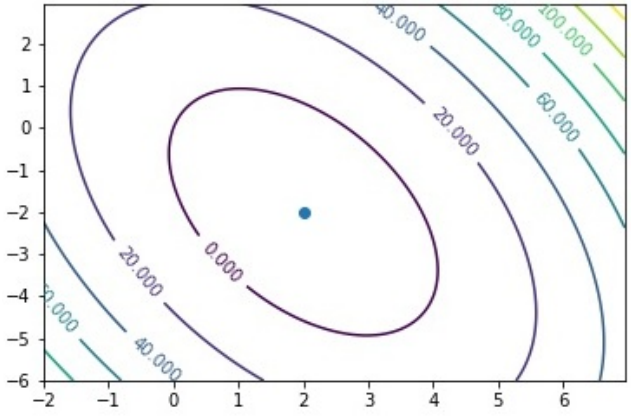

Fig. 3. The corresponding contour graph of Fig. 2, where the minimum x is $[-2-2]^T$.

a small one yields high cost of computational resources. However, when solving linear equation problems, the difficulty can be overcome by simplistic maths, which gives the exact value of α. To convenient the derivation, residual r is defined as

$$r_i = b - Ax_i \tag{6}$$

, which indicates how far x is from the correct value b. Importantly, Equation 6 is also equal to negating Equation 3, meaning a simplistic property that residual r is also the negating gradient. So, we can further get $-f'(x_{i+1}) = r_{i+1}$.

To minimize $f(x_{i+1})$ by choosing α, we should thus solve $\frac{\mathrm{d}}{\mathrm{d}\alpha} f(x_{i+1}) = 0$ from basic calculus:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} f(x_{i+1}) = f'(x_{i+1})^T \frac{d}{d\alpha} x_{i+1} = -r_{i+1}^T r_i = 0 \tag{7}$$

And after tedious maths of solving Equation 7,we can get the annoying α in a simple form:

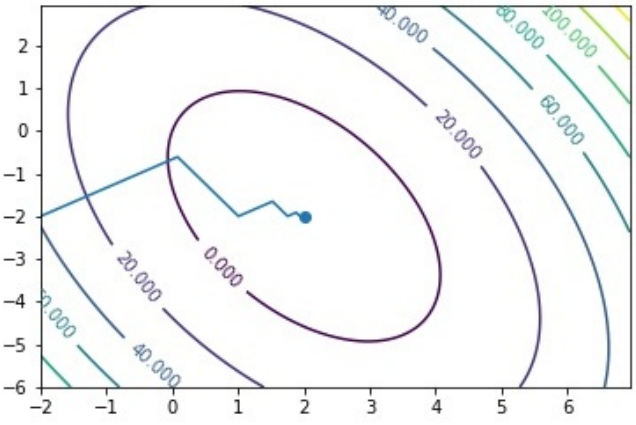

Fig. 4. Recursive trajectory of gradient descent method, where x starts from $[-2-2]^T$

$$\begin{aligned} &r_{i+1}^T r_i = 0 \\ &(b - Ax_{i+1})^T r_i = 0 \\ &(b - A(x_i + \alpha r_i))^T r_i = 0 \\ &r_i^T r_i - \alpha (Ar_i)^T r_i = 0 \\ &\alpha = \frac{r_i^T r_i}{r_i^T A r_i} \end{aligned} \tag{8}$$

Therefore, based on Equation 8, pseudo-code is listed in Algorithm 1 and an example result where x starts from $[-2-2]^T$ can be seen in Fig. 4.

Algorithm 1 Gradient Descent Method

```
initial start point x and negative gradient r = b − Ax
repeat
    q = Ar
    step size α = r^T r / r^T q
    renew point x = x + αr
    renew gradient r = r − αq or r = b − Ax
until (r^T r < epsilon)
```

3.2 Conjugate gradient method

Though gradient descent method is effective, the zigzag trajectory seems rather painful which costs foreseeable computational resources. Therefore, to overcome the iteration barrier, conjugate gradient method is here introduced. Different from descent method, gradient r_i is not used as the search direction. But since gradient has perfect orthogonal property(see Equation 8):

$$r_i r_j = 0, \ i \neq j \tag{9}$$

, gradient r span $\{r_0, r_1, r_2, ..., r_i\}$ is used to construct search directions, where

$$d_i = r_i + \beta_i d_{i-1} \tag{10}$$

and d_i satisfies the property:

$$d_i A d_j = 0, \ i \neq j \tag{11}$$

Then the pseudo-code is presented in algorithm 2, the difference from the algorithm 1 is the special A-orthogonal vector, thus the step size α value should be changed and coefficient β value is introduced for construction. [1]

[1] Detailed mathematics can be found in reference: Shewchuk (1994), this article will not involve the details since the author doesn't think he can do better.

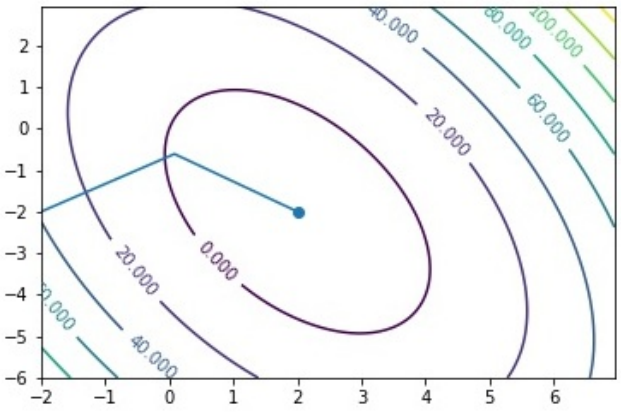

Fig. 5. Recursive trajectory of conjugate gradient method, where x starts from $[-2\ 2]^T$

In conclusion, the results of x starting from $[-2\ -\ 2]^T$ can be seen in Fig. 5. only 2 iterations was conducted, attributed to the improved search direction.

Algorithm 2 Conjugate Gradient Method

initial start point x, negative gradient $r_{old} = b - Ax$ and search direction $d = r_{old}$
repeat
  $q = Ad$
  step size $\alpha = \frac{r_{old}^T r_{old}}{d^T q}$
  renew point $x = x + \alpha d$
  renew residual $r_{new} = r_o ld - \alpha q$ or $r_{new} = b - Ax$
  $\beta = \frac{r_{new}^T r_{new}}{r_{old}^T r_{old}}$
  renew $d = r_{new} + \beta d$
  $r_{old} = r_{new}$
until $(r_{new}^T r_{new} < epsilon)$

3.3 Iteration Comparison Between Gradient Descent and Conjugate Gradient

To further confirm the effectiveness of A-orthogonal search direction d, 18×18 points were sampled for comparison. As shown in Fig. 6, z-axis represents Iteration(GD)-Iteration(CG) and $z(x) \geq 0$, which means for all the sampled points, conjugate gradient is an effective improvement. In addition, one can see that, at some points, conjugate gradient behaves the same as gradient descent. That's because these points only take one iteration to solve the problem and in this situation conjugate gradient is equal to gradient descent since $r_0 = d_0$.

In fact, for a n-dimensional problem, conjugate gradient only takes at most n iterations to solve it theoretically. Though there are round-off error, which may cause the loss of A-orthogonal property, conjugate gradient is generally considered to be superior than gradient descent.

4. DISCUSSION AND CONCLUSION

In conclusion, this article is a simple realization of gradient method solving linear equation $A = bx$, where A should be a symmetric and positive-definite n×n matrix. In this situation, conjugate gradient is superior to gradient descent if we overlook the round-off error.

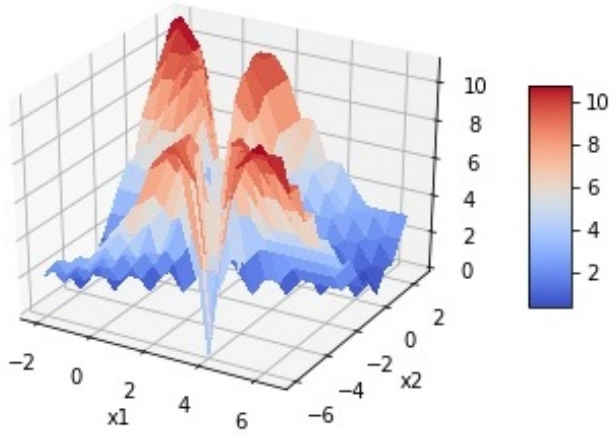

Fig. 6. Iteration Comparison Between Gradient Descent and Conjugate Gradient, the z-axis represents Iteration(GD)-Iteration(CG).

However, cruel reality and marvelous nature always destruct the simplicity of matrix A, which means A could be non-symmetric or indefinite and then a quadratic form function cannot be used. This can be solved by constructing other functions, such as $min|Ax - b|^2$. Moreover, conjugate gradient, like gradient descent, can also be used to minimize any continuous function f(x) for which the gradient f can be computed. The difference can be concluded in three aspects: first, gradient cannot be solved in a recursive way;second, step size α can be hard to compute and third, coefficient β can be chosen by several ways.

ACKNOWLEDGEMENTS

Thanks for Dr.Dong's detailed instruction on computational physics and thank my supervisor Dr.Chen for allowing me to learn something I enjoy.

REFERENCES

Shewchuk, J.R. (1994). *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.