Numerical Optimization in Robotics

L1--作业提示

主讲人 葫芦娃

前言

L1

- 了解不同数值优化问题需要用什么求解方法
- 可以使用成熟求解器来实现求解

L2

- 满足L1
- 了解不同算法的基本原理与推导过程
- 手动编程实现优化求解器算法的全过程

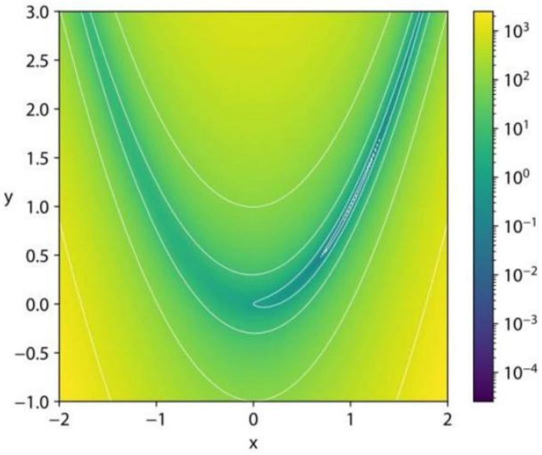
L3

- 满足L2
- 可以手动推导已有不同算法的实现方法
- 针对关键环节给出证明

作业题目回顾

深蓝学院
shenlanxueyuan.com

1. Problem description:Linear-search Steepest Gradient Descent



$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_N) = \sum_{i=1}^{N/2}\left[100\left(x_{2i-1}^2 - x_{2i}\right)^2 + \left(x_{2i-1} - 1\right)^2\right]$$

You have to implement Linear-search Steepest Gradient Descent to optimize the function given above. It is obvious that the global minimum of the function is $(1,1,\ldots,1)_{N\times 1}$, so you can check if the result of your program is admissible.

Unqualified: The results are incorrect, or the assignment is not written in the required format.

Qualified: The results are somewhat different from the standard results, but there is a correct knowledge and understanding of the assignment requirements.

Good: The results are correct. The program converges correctly to the minimum value of the objective function.

OutStanding: There are two chance for you to get an ***'OutStanding'*** :

(1) Your program can optimize a Rosenbrock function for a given N of dimensions.

(2) Your program can visualize a two-dimension Rosenbrock function.

What's more, you must have an in-depth understanding of Rosenbrock function optimization, which should be expressed in your report.

算法实现

算法 1: Steepest Gradient Descent with Armijo Condition

$k := 0, c := c_0, \mathbf{x} := \mathbf{x}_0, found := false$;

while *not found* **do**

  $\mathbf{g} := \triangledown f(\mathbf{x})$;

  if $||\mathbf{g}|| < \epsilon$ **then**

    $found := true$;

  else

    $\alpha := 1, \mathbf{x}_{new} := \mathbf{x} - \alpha\mathbf{g}$;

    while $f(\mathbf{x}_{new}) > f(\mathbf{x}) - c\alpha\mathbf{g}^\top\mathbf{g}$ **do**

      $\alpha := \alpha/2, \mathbf{x}_{new} := \mathbf{x} - \alpha\mathbf{g}$;

    end

    $\mathbf{x} := \mathbf{x}_{new}, k := k + 1$;

  end

end

需要完成工作

1)函数f(x) 与其导数g(x)实现

2)算法实现

算法实现

函数f(x) 与其导数g(x)实现

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_N) = \sum_{i=1}^{N/2}\left[100\left(x_{2i-1}^2 - x_{2i}\right)^2 + \left(x_{2i-1} - 1\right)^2\right]$$

Rosenbrock函数

$f(x) = f(x_1, x_2, \ldots, x_n)$

$= \sum_{i=1}^{N/2}[100(x_{2i-1}^2 - x_{2i})^2 + (x_{2i-1} - 1)^2]$

$= [100 * (x_1 - x_2)^2 + (x_2 - 1)^2] + \ldots + [100 * (x_{N-1} - x_N)^2 + (x_N - 1)^2]$

修改为从0开始索引

$f(x) = f(x_1, x_2, \ldots, x_n)$

$= \sum_{i=1}^{N/2}[100(x_{2i-2}^2 - x_{2i-1})^2 + (x_{2i-2} - 1)^2]$

$= [100 * (x_0^2 - x_1)^2 + (x_0 - 1)^2] + \ldots + [100 * (x_{N-2}^2 - x_{N-1})^2 + (x_{N-2} - 1)^2]$

函数的求导：

$$\frac{\delta f}{\delta x_{2i-2}} = 400 * x_{2i-2} * (x_{2i-2}^2 - x_{2i-1}) + 2 * (x_{2i-1} - 1)$$

$$\frac{\delta f}{\delta x_{2i-1}} = -200 * (x_{2i-2}^2 - x_{2i-1})$$

函数 N=2

$$f(x_0, x_1) = 100 * (x_0^2 - x_1)^2 + (x_0 - 1)^2$$

函数的梯度 N=2

$$\frac{\delta f}{\delta x_0} = 400 * x_0(x_0^2 - x_1) + 2 * (x_0 - 1)$$

$$\frac{\delta f}{\delta x_1} = -200 * (x_0^2 - x_1)$$

建议

1）在学习课程与作业的时候，最好能够结合视频课程、推荐书籍以及网络资源等多方面的资源，可以加强自己的理解

2）学习过程中，鼓励实现不同的算法来对比，甚至使用成熟求解器来做对比

3）函数的实现及求导，需要认真仔细，最好能够自己设计一些可控的测试函数进行自测

4）作业过程中，可以尝试自己调参，看看过程中不同参数对结果有些什么影响

5）作业的提交，建议按照规范

在线问答

Q&A

深蓝学院

shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening