Multidimensional optimization

Release 1.0.0

Victor Barbarich

Mar 28, 2022

CONTENTS:

CHAPTER

ONE

CONCEPTUAL MODEL

1.1 Imagine situation:

Factory's management has a function that describes the income from the sale of products. The function depends on many variables, e.g. the salary by each employee, the cost of each product and so on.

The management gave us the function $g(\mathbf{x})$, the running costs, the gradient of the functions in analytical form (Their mathematicians did a good job). But it will better if our solution automatically calculates gradient.

And our goal is to ensure the **best cost distribution**.

* For unifying let's introduce $f(\mathbf{x}) = -g(\mathbf{x})$.

CHAPTER

TWO

MATHEMATICAL MODEL

We have analyzed their problem. The solution is to use the gradient methods.

We have chosen 2 methods:

1. Gradient Descent
2. Nonlinear conjugate gradient method

2.1 Problem statement

1. $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$
2. $\mathbf{x} \in X \subseteq \mathbb{R}^n$
3. $f \longrightarrow \min\limits_{\mathbf{x} \in X}$
4. The f is defined and differentiable on the X
5. Convergence to a local minimum can be guaranteed. When the function f is convex, gradient descent can converge to the global minima.

2.2 Gradient descent

2.2.1 Equations

1. Function argument:

$$x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^\top \tag{2.1}$$

2. Gradient:

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2} \; \dots \; \frac{\partial f}{\partial x_n} \right]^\top \tag{2.2}$$

3. Gradient step:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma_i \cdot \nabla f(\mathbf{x}_i) \tag{2.3}$$

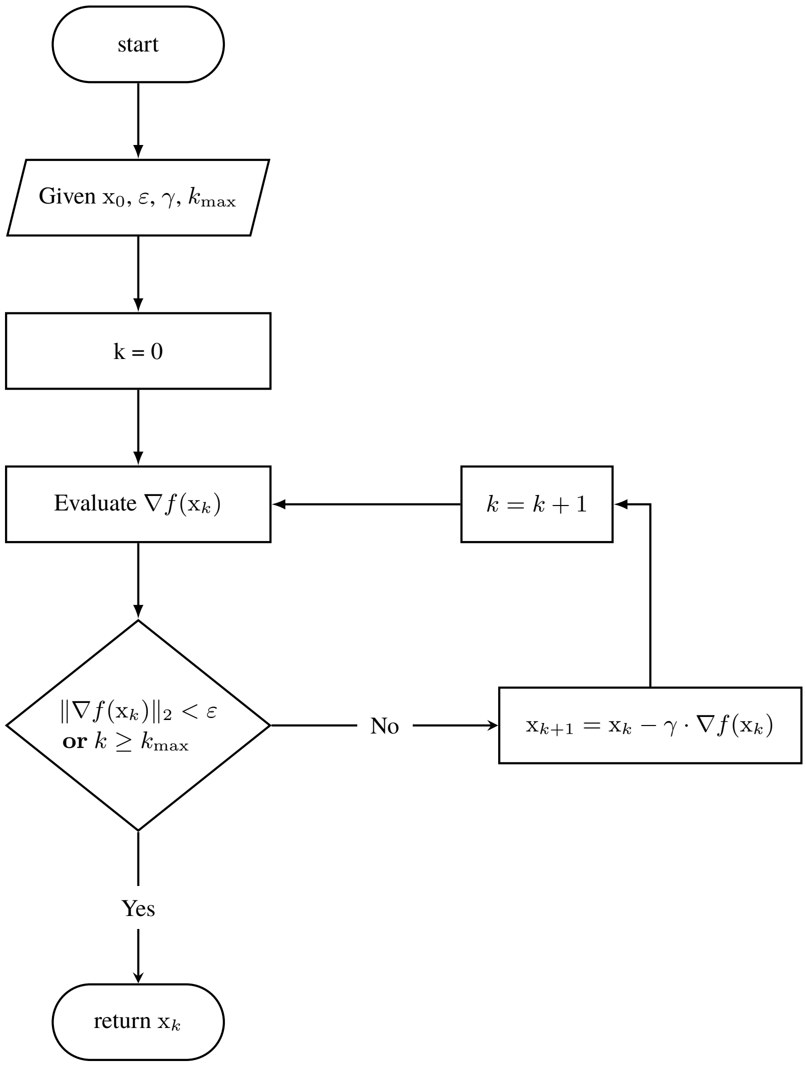
4. Terminate condition:

$$\|\nabla f(\mathbf{x}_i)\|_2 < \varepsilon \tag{2.4}$$

2.2.2 Algorithm with constant step

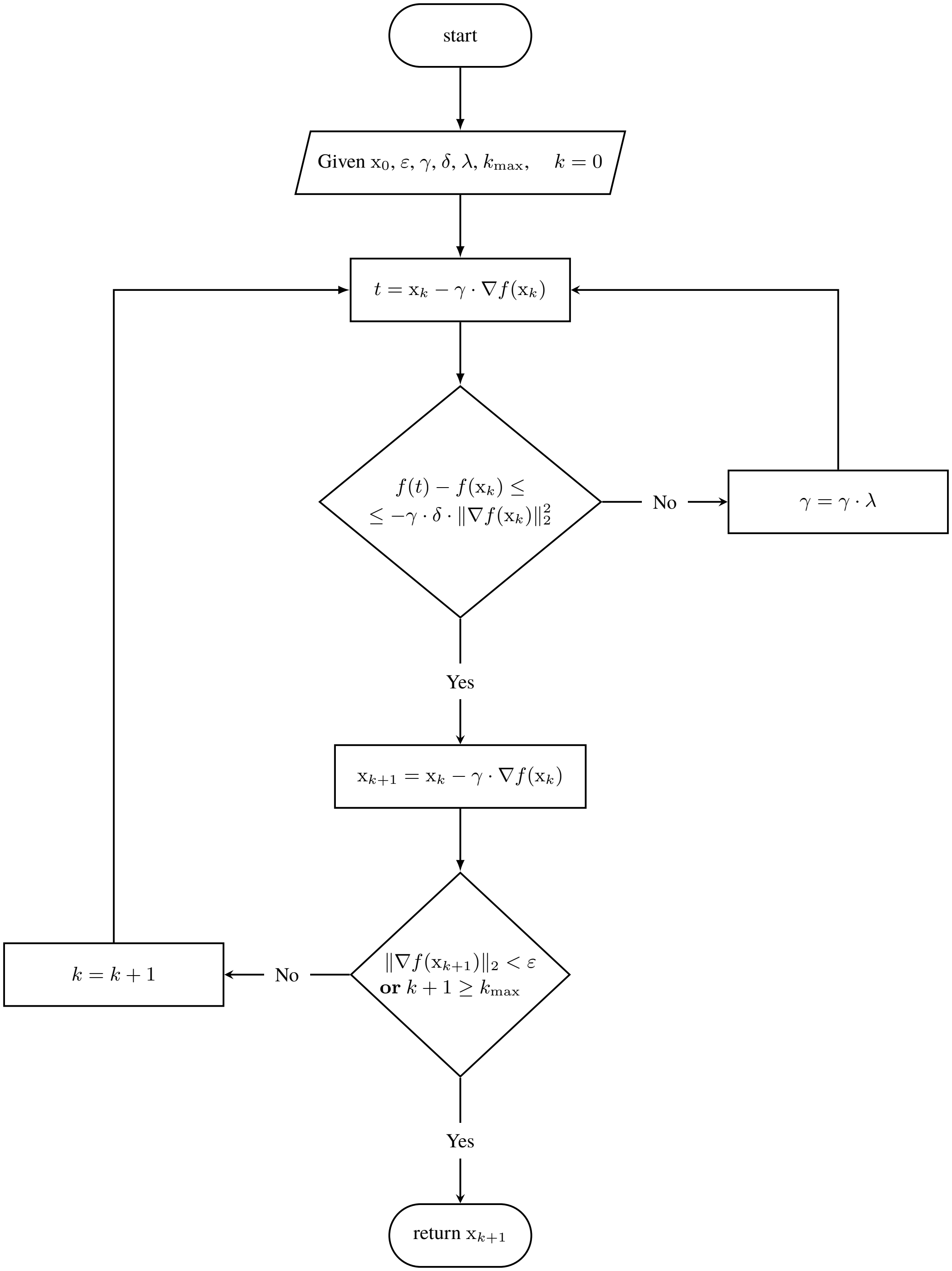
The gradient of the function shows us the direction of increasing the function. The idea is to move in the opposite direction to $\mathbf{x}_{k+1}$ where $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

But, if we add a gradient to $\mathbf{x}_k$ without changes, our method will often diverge. So we need to add a gradient with some weight γ.



2.2.3 Algorithm with descent step

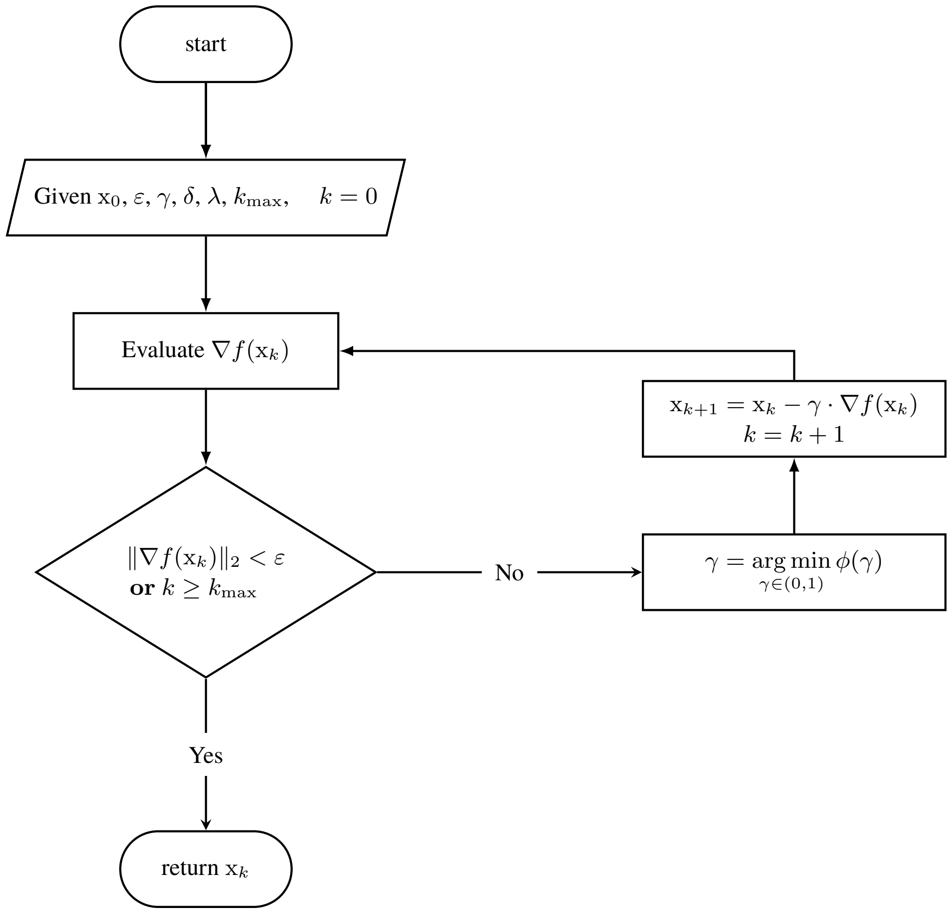
Requirements: $0 < \lambda < 1$ is the step multiplier, $0 < \delta < 1$.



2.2.4 Algorithm with optimal step size

Another good idea is to find a γ that minimizes $\phi(\gamma) = f(\mathrm{x}_k + \gamma \cdot \nabla f(\mathrm{x}_k))$

So we have a task to find the $\gamma_{\min} = \underset{\gamma \in (0,1)}{\arg\min}\, \phi(\gamma)$. We will use Brent's algorithm to search $\gamma_{\min}$.



2.3 Strong Wolfe conditions

The conditions necessary to minimize $\phi(\gamma) = f(\mathrm{x}_k + \gamma p_k)$ and find $\gamma_k = \underset{\gamma}{\arg\min}\, \phi$

$$f(\mathrm{x}_k + \gamma_k p_k) \leq f(\mathrm{x}_k) + c_1 \gamma_k \nabla f_k^\top p_k \tag{2.5}$$

$$|\nabla f(\mathrm{x}_k + \gamma_k p_k)^\top p_k| \leq -c_2 \nabla f_k^\top p_k \tag{2.6}$$

2.4 Nonlinear conjugate gradient method

The Fletcher–Reeves method. p_k is the direction to evaluate x_{k+1}.

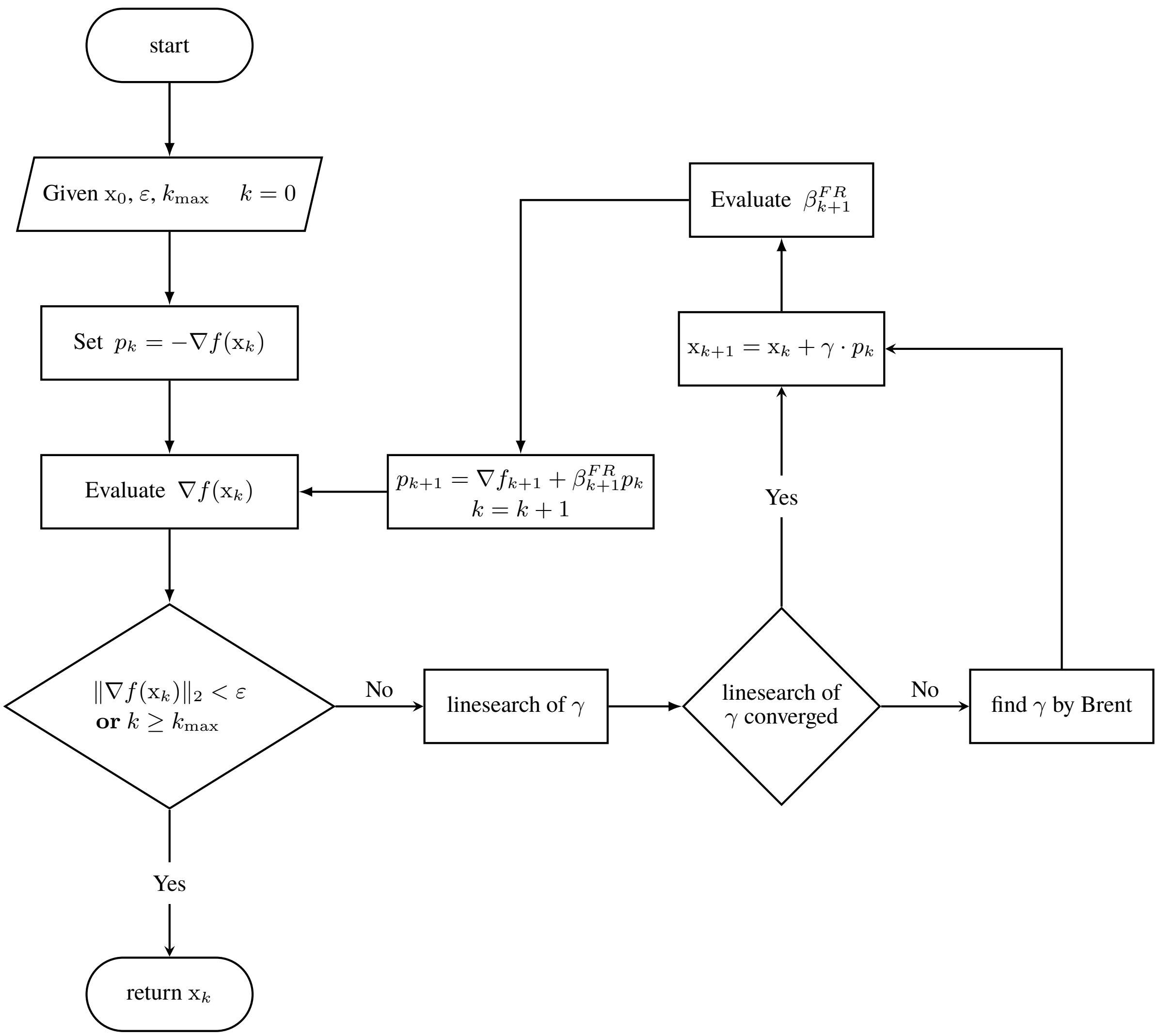
1. $p_0 = -\nabla f_0$
2. $p_{k+1} = \nabla f_{k+1} + \beta^{FR}_{k+1} p_k$

In the RF method, γ is searched using Line Search (Nocedal, Wright (2006) *Numerical Optimization* pp.60-61)

Our modification is that if Line Search does not converge, use Brent's algorithm to search for $\gamma_{\min} = \arg\min_{\gamma} \phi$

$$\beta^{FR}_{k+1} = \frac{\nabla f^{\top}_{k+1} \nabla f_{k+1}}{\nabla f^{\top}_{k} \nabla f_{k}} \tag{2.7}$$

2.4.1 Algorithm Nonlinear conjugate gradient method



CHAPTER

THREE

REQUIREMENTS

3.1 Required input fields

1. Type of algorithm. (Gradient with constant step, fractional step, optimal step, nonlinear conjugate gradient method and Newton-CG)
2. Field with function input $f(\mathrm{x})$
3. Field with the analytical gradient flag. If analytical gradient flag is True then request the gradient in analytical form. $\nabla f(\mathrm{x})$
4. Field with the start point (The default value is random numbers from a uniform distribution on (-1, 1))
5. Search precision ε
6. Field with a history saving flag
7. Required parameters for each method

3.2 Visualization Required

1. If the function depends on 1 variable, then it is necessary to draw a curve and animate the movement of the point and draw the previous steps.
2. If the function depends on 2 variables, then you need to draw contour lines and animate the movement of the point and draw the previous steps.
3. For functions of larger dimensions, output a graph of the decreasing gradient norm and the function values for each iteration.
4. For 1 and 2 dimensional output a graph of the decreasing gradient norm and the function values for each iteration too.

3.3 Requirements for methods

1. Each method must keep a history of all iterations. Be sure to save every x_k, f_k, $\|\nabla f_k\|$
2. The solution must contain the final point, the value of the function, the number of iterations, and the history.
3. The solution should cause a minimum number of calculations of the function values