WOLFE CONDITIONS

What is it?

Unconstrained Optimization Algorithms

- Direct Search
- Line Search
  - Initial point
  - Search direction
  - Distance to move along search direction
  - When to stop
- Trust Region

lets us know if we've done a "good enough" line search

- Set of inequalities for performing inexact line search (esp. in quasi-Newton methods)
- As we conduct our line search, we are looking for an α^* that is "good enough" (i.e., what is a "good enough" estimate of an optimum along a line)
  - ↳ The Wolfe conditions give us a quantitative meaning to "good enough"

α^* : optimum distance along a line that approximates the minimum

- Give us "allowable" step lengths (α) in a line search that reduces the objective function "sufficiently"
  - ↑ rather than minimizing the objective function over $\alpha \in \mathbb{R}^+$

What do we need to know? (assumptions)

- There are two Wolfe conditions
- There are two Strong Wolfe conditions

} the first condition (Amijo rule) is the same for both the Wolfe and Strong Wolfe Conditions

Advantages?

- If you apply these rules across subsequent line searches, then the overall optimization will converge well ← proofs exist to show this

Disadvantages?

- Do not necessarily measure convergence of the overall algorithm

Main Points:

The Wolfe Conditions tell us when to stop our line search (when we have performed a "good enough" line search).

Goal of Wolfe conditions:

Tell us when our α is a "good enough" representation of the minimum along the line

WOLFE CONDITIONS

Armijo Rule:

Armijo rule is the first Wolfe condition: ← also called "sufficient decrease"

$$f(x_{k-1} + \alpha s_k) \leq f(x_{k-1}) + c_1 \alpha s_k^T \nabla f(x_{k-1}) \qquad \varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0)$$

can also be written as: ↔ same thing

$$f(x_k + \alpha s_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T s_k$$

where c_1 is a constant ($c_1 \in (0,1)$; typically $c_1 = 10^{-3}$ to 10^{-4}) ← quite small (used to dampen out effect of gradient)

What does it mean?

- Stipulates that α should give a sufficient decrease in the objective function f
- The reduction in f should be proportional to both the step length (α) and the directional derivative ($\nabla f_k^T s_k$)

Breakdown:

$f(x_{k-1} + \alpha s_k)$ ← line search equation

$s_k^T \nabla f(x_{k-1}) = \left.\frac{df}{d\alpha}\right|_{x_{k-1}}$ ← projection of the gradient on to the search direction s_k
↑ dot product ← derivative of f with respect to α along the search direction

Slope of line
$f(x_{k-1}) + c_1 \alpha s_k^T \nabla f(x_{k-1})$ ← 1st order Taylor series expansion of the function about the initial point
equation of line linear in α ← tangent line to the function at the initial point
← linear approximation of the function at the initial point

$$f(x_{k-1} + \alpha s_k) \leq f(x_{k-1}) + c_1 \alpha s_k^T \nabla f(x_{k-1})$$

function value f evaluated along the line at the α I'm interested in — needs to be $\leq$ — the function value at the starting point plus a constant multiplier (c_1)

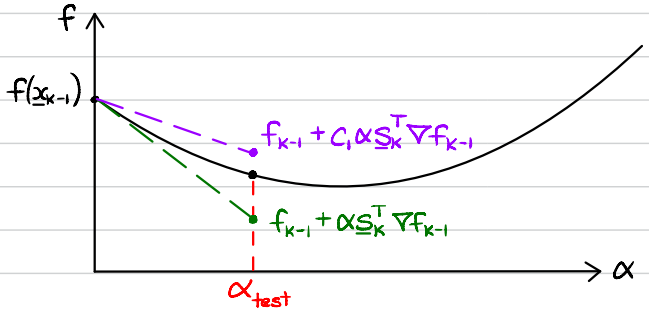
Main Points:

The functional at a given x needs to be less than the initial point plus some small downward slope.
We are searching in a descent direction (where the function decreases as we move in $+\alpha$) and we want the function to decrease "enough"

WOLFE CONDITIONS

Visual examples:

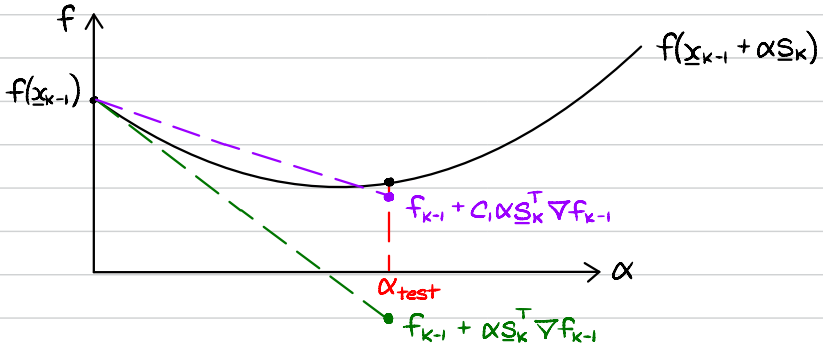
An α that satisfies the first Wolfe condition (Armijo rule):

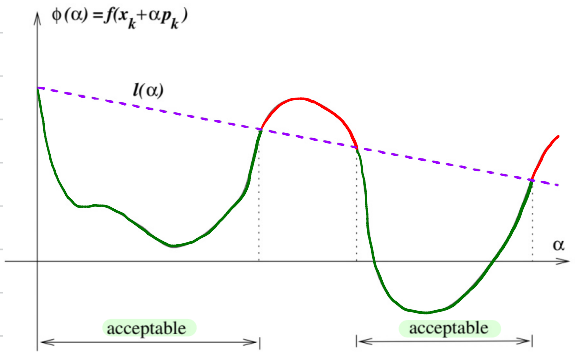


c_1 offsets the slope of the purple line relative to the green line

- we've taken a step "far enough" to get a sufficient decrease in the function

An α that does not satisfy the first Wolfe condition (Armijo rule):





← Lines in green satisfy the Armijo rule

← Lines in red do not satisfy the Armijo rule

← needs to be less than the initial point plus some small downward slope

So, what are we doing?

Our goal is to be to the left of where the minimum is located

- we don't know what the minimum is, we just know an estimate of the gradient of the function at some initial condition or point
- we're trying to make sure we don't overshoot the minimum

The first condition says, "Let's not go too far to the right."

Main Points:

The first Wolfe condition ensures that we don't overshoot the minimum:

1. "Don't go too far!"

but we need another condition to avoid taking too small steps (too small α):

2. "You need to go far enough!"

WOLFE CONDITIONS

Why can't we stop here?

If we stopped after satisfying only the first condition, we may not go far enough!

↳ any value that is "small enough" will satisfy this condition

=> we need another condition to avoid tediously small steps in α

Curvature condition:

The curvature condition is the second Wolfe condition:

$$\underline{s}_k^T \nabla f(\underline{x}_{k-1} + \alpha \underline{s}_k) \geq c_2 \underline{s}_k^T \nabla f(\underline{x}_{k-1}) \qquad \varphi'(\alpha) \geq c_2 \varphi'(0)$$

Can also be written as: ↔ same thing

$$\nabla f(\underline{x}_k + \alpha_k \underline{s}_k)^T \underline{s}_k \geq c_2 \nabla f_k^T \underline{s}_k$$

where c_2 is a constant ($c_2 \in (c_1, 1)$; typically $c_1 = 0.1$ to 0.9)

$c_2 = 0.9$ when s_k is chosen by a Newton or quasi-Newton method

$c_2 = 0.1$ when s_k is obtained from a nonlinear conjugate gradient method

} typical values

What does it mean?

- Stipulates that α should give a sufficient increase in the slope of the function f along the search line

slope of the function starts off negative in the search direction and we're trying to make it zero

Breakdown:

$\underline{s}_k^T \nabla f(\underline{x}_{k-1} + \alpha \underline{s}_k)$ ← slope at any given point along the line
← dot product (looking at the derivative $\frac{df}{d\alpha}$ measured anywhere along the line)

$\underline{s}_k^T \nabla f(\underline{x}_{k-1})$ ← slope at the initial point

$$\underline{s}_k^T \nabla f(\underline{x}_{k-1} + \alpha \underline{s}_k) \geq c_2 \underline{s}_k^T \nabla f(\underline{x}_{k-1})$$

the slope of the function evaluated along the line at the α I'm interested in — needs to be $\geq$ — c_2 times the slope of the function at the initial point

Main Points:

The second Wolfe condition ensures that:

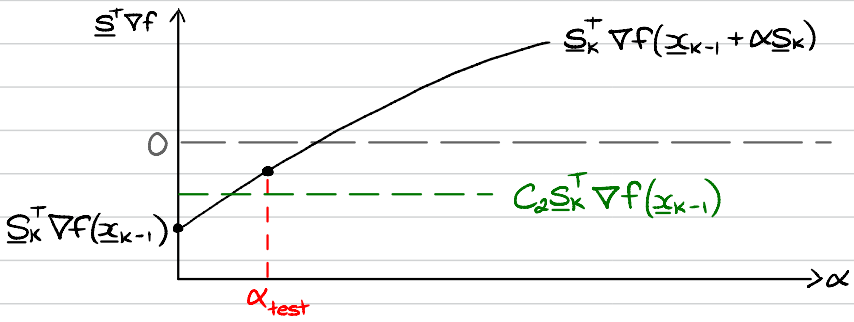
- we avoid taking tediously small steps in α
- α gives a sufficient increase in the slope of the function along the search line

WOLFE CONDITIONS

Visual examples:

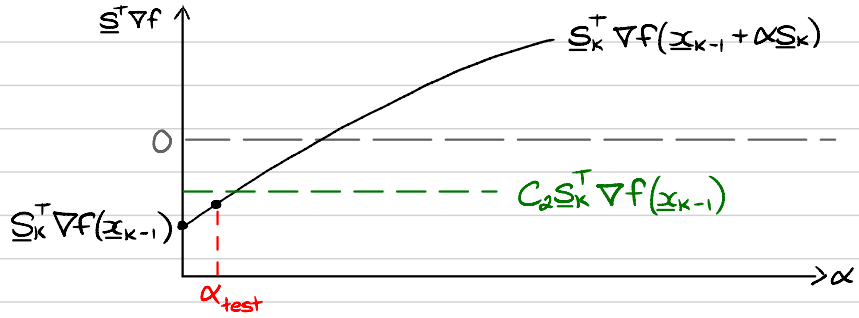
Let's look at a plot not of the function, but of the derivative along that line (dot product of search direction and gradient: $\underline{s}_k^T \nabla f$)

An α that satisfies the second Wolfe condition (curvature condition):



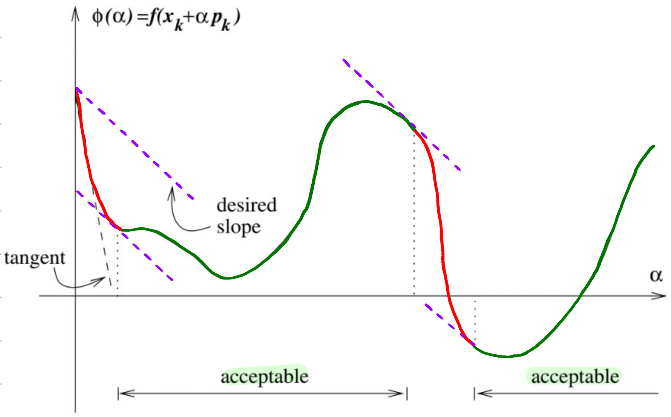
← want to be greater than the green line (greater than some multiple of the initial slope)

An α that does not satisfy the second Wolfe condition (curvature condition):



← did not increase the slope enough to meet the second Wolfe condition (did not take a large enough step)

Looking at a plot of the function:



← Lines in green satisfy the curvature condition

← Lines in red do not satisfy the curvature condition

← slope needs to be greater than the slope of the function at the initial point times some constant

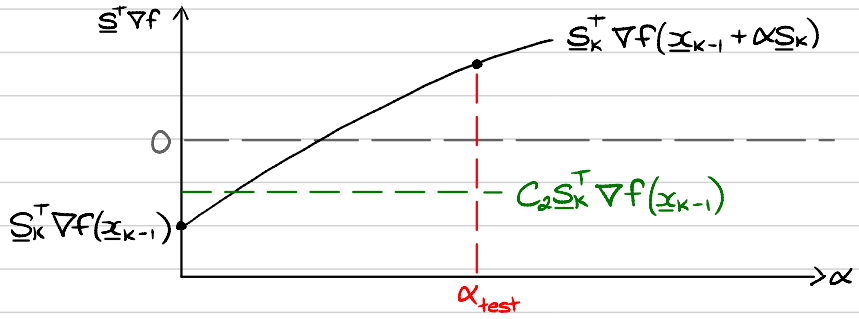
Main Points:

Any sufficiently small step will satisfy the first Wolfe condition
Any sufficiently large step will satisfy the second Wolfe condition

WOLFE CONDITIONS

Visual examples:

Let's take another look at the plot of the derivative along the line (dot product of search direction and gradient: $\underline{s}_k^T \nabla f$). This α also satisfies the second Wolfe condition (curvature condition):



It can be argued that this is not a very good approximation of the optimal α

↳ the magnitude of the derivative in the search direction is rather large at this α

=> we need to modify this condition to avoid steps in α that are too large

Strong Wolfe conditions:

- The first Wolfe condition (Armijo rule / sufficient decrease condition) does not change

$$f(\underline{x}_{k-1} + \alpha \underline{s}_k) \leq f(\underline{x}_{k-1}) + c_1 \alpha \underline{s}_k^T \nabla f(\underline{x}_{k-1}) \qquad \varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0)$$

Can also be written as: (same thing)

$$f(\underline{x}_k + \alpha \underline{s}_k) \leq f(\underline{x}_k) + c_1 \alpha \nabla f_k^T \underline{s}_k$$

Main Points:

Any sufficiently small step will satisfy the first Wolfe condition, but we don't want steps that are too small => second Wolfe condition

Any sufficiently large step will satisfy the second Wolfe condition, but we don't want steps that are too large => second strong Wolfe condition

WOLFE CONDITIONS

Strong Wolfe conditions:

- The second Wolfe condition (curvature condition) is changed to make it a statement on the magnitude of the derivative in the search direction:

$$\left|\underline{s}_k^T \nabla f(\underline{x}_{k-1} + \alpha \underline{s}_k)\right| \leq c_2 \left|\underline{s}_k^T \nabla f(\underline{x}_{k-1})\right| \qquad |\varphi'(\alpha)| \leq c_2 |\varphi'(0)|$$

same thing

can also be written as:

$$\left|\nabla f(\underline{x}_{k-1} + \alpha_k \underline{s}_k)^T \underline{s}_k\right| \leq c_2 \left|\nabla f_k^T \underline{s}_k\right|$$

→ As we go along the function in the descent direction, the slope of the function is increasing
  ↳ descent => negative slope
  ↳ minimum => zero slope

→ As we go along the function in the descent direction, the magnitude of the slope of the function is decreasing
  ↳ descent => negative slope => positive magnitude
  ↳ minimum => zero slope

$$\left|\underline{s}_k^T \nabla f(\underline{x}_{k-1} + \alpha \underline{s}_k)\right| \leq c_2 \left|\underline{s}_k^T \nabla f(\underline{x}_{k-1})\right|$$

the magnitude of the slope of the function evaluated along the line at the α I'm interested in — needs to be $\leq$ — c_2 times the magnitude of the slope of the function at the initial point

↳ the absolute value of the slope of the function at any point on the line must be less than or equal to c_2 times the absolute value of the slope at the initial point
  => now we get a range of α's that are close to the minimum because the magnitude of the slope is small

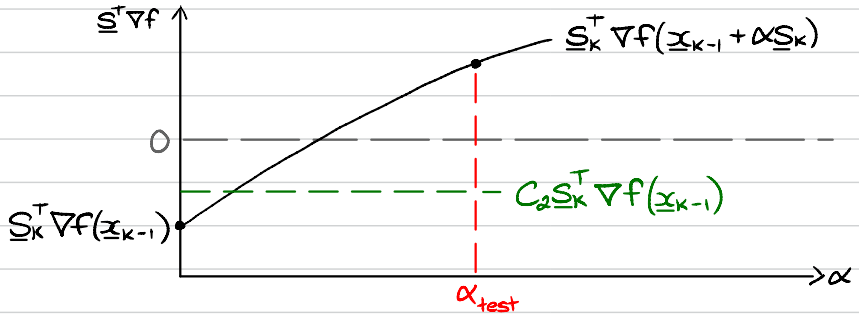
Main Points:

The first strong Wolfe condition still makes sure that the function decreases "enough" but we don't overshoot the minimum.
The second strong Wolfe condition still makes sure that we avoid tediously small steps in α, but now it also precludes values of α that increase the magnitude of the derivative.

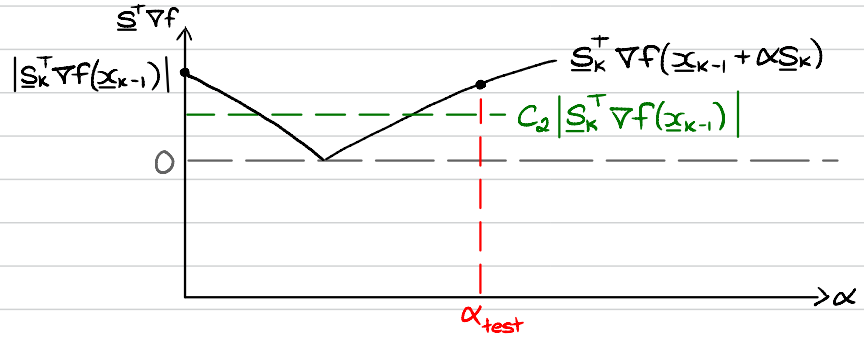
WOLFE CONDITIONS

Visual examples:

Let's take another look at the plot of the derivative along the line (dot product of search direction and gradient: $\underline{s}_k^T \nabla f$) and the second α satisfies the second Wolfe condition (curvature condition):

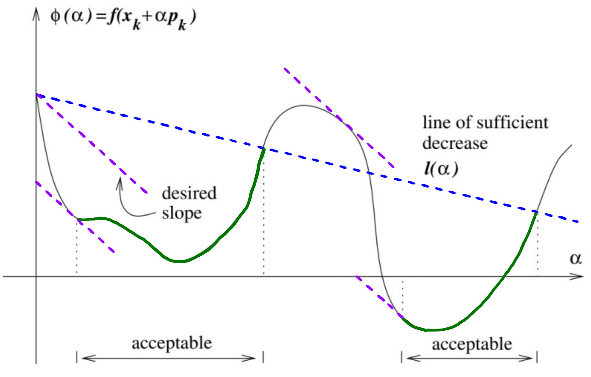


If we change the condition from the second Wolfe condition to the second strong Wolfe condition, we see that this α does not satisfy the second strong Wolfe condition (strong curvature condition):



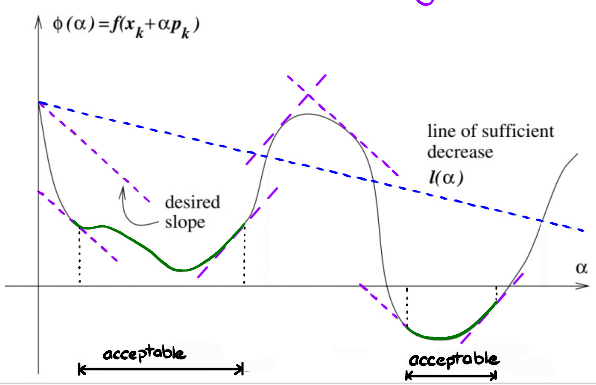
Wolfe Conditions

1st condition 2nd condition



Strong Wolfe Conditions

1st condition 2nd strong condition

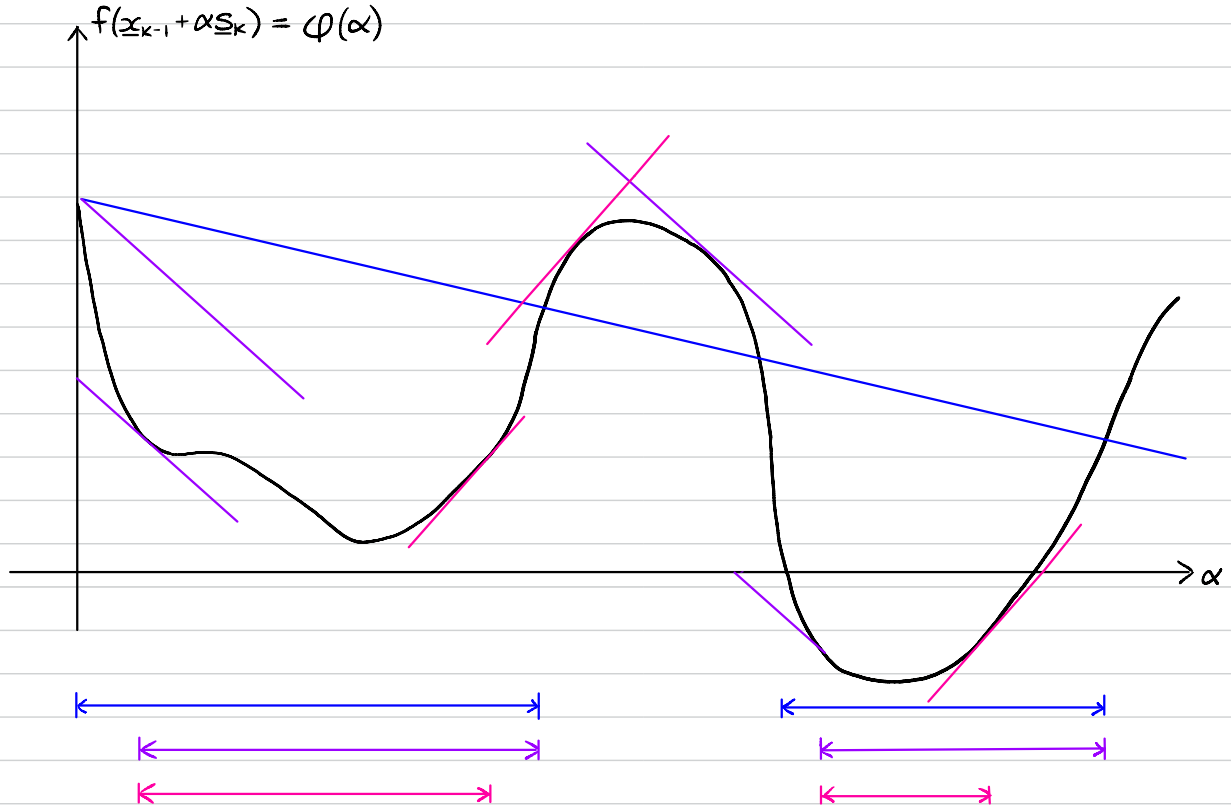


Main Points: Wolfe conditions are effectively just another way to define a bracket, but not just on function values, but on information about the slopes

→ very useful for gradient-based optimization methods

WOLFE CONDITIONS

Visual summary:



function $f(x_{k-1}+\alpha s_k)$ ← line search equation

1st Wolfe condition (Armijo rule / sufficient decrease) — acceptable

$$f(x_{k-1}+\alpha s_k) \leq f(x_{k-1}) + c_1 \alpha s_k^T \nabla f(x_{k-1})$$

$$\varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0)$$

2nd Wolfe condition (curvature condition) — acceptable

$$s_k^T \nabla f(x_{k-1}+\alpha s_k) \geq c_2 s_k^T \nabla f(x_{k-1})$$

$$\varphi'(\alpha) \geq c_2 \varphi'(0)$$

2nd strong Wolfe condition (strong curvature condition) — acceptable

$$|s_k^T \nabla f(x_{k-1}+\alpha s_k)| \leq c_2 |s_k^T \nabla f(x_{k-1})|$$

$$|\varphi'(\alpha)| \leq c_2 |\varphi'(0)|$$

WOLFE CONDITIONS

Backtracking:

- If we begin a line search with a "large" α and then successively decrease it by a constant percentage ← Known as backtracking

- When performing backtracking, it can be shown that it is only necessary to check the 1st Wolfe condition (Armijo rule) ← don't need to check 2nd Wolfe condition (curvature condition)

Backtracking algorithm:

1. Given α_1, $\tau \in (0,1)$, $c_1 \in (0,1)$
2. Set $k = 1$
3. Evaluate $\varphi(\alpha_k)$
   If $\varphi(\alpha_k) \leq \varphi(0) + c_1 \alpha_k \varphi'(0)$, set $\alpha = \alpha_k$, break ← If $\varphi(\alpha_k) > \varphi(0) + c_1 \alpha_k \varphi'(0)$ $\Rightarrow$ α_k violates 1st condition $\Rightarrow$ α_k is too large/long
4. $\alpha_{k+1} = \tau \alpha_k$
5. $k = k+1$, go to step 2

- $\alpha_1 = 1$ in Newton and quasi-Newton methods
  ↳ can have different values in other algorithms (e.g. steepest descent, conjugate gradient)

- An acceptable step length (α_k) will be found after a finite number of trials because α_k will eventually become small enough that the 1st Wolfe condition holds
- In practice, τ can vary at each iteration
  ↳ as long as $\tau \in [\tau_{low}, \tau_{high}]$ for some fixed constants $0 < \tau_{low} < \tau_{high} < 1$

Advantages?

- Only have to use first Wolfe condition (Armijo rule / Sufficient decrease)
  ↳ backtracking ensures that α_k is some fixed value or that it is short enough to satisfy the 1st Wolfe condition, but not too short
- Well-suited for Newton methods

Disadvantages?

- Can be slow
- Does not use information about function evaluations $\varphi(\alpha_k)$
- Not as appropriate for quasi-Newton and conjugate gradient methods

Refined technique:

Interpolate data at $\alpha = 0, \alpha_k$ to obtain estimate of $\min_\alpha \varphi(\alpha)$

Comments: More information on backtracking can be found on pg. 37 of "Numerical Optimization" by Jorge Nocedal and Stephen Wright (2nd Ed)

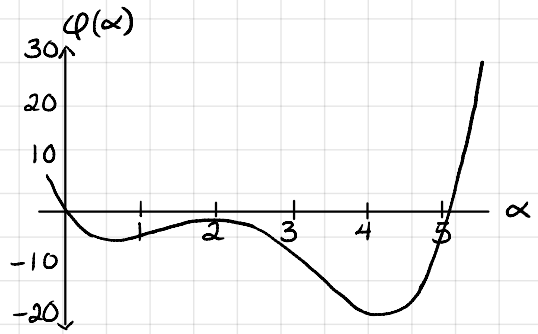
WOLFE CONDITIONS

Example:

Find the intervals for the merit function $\varphi(\alpha)$ satisfying the Wolfe conditions.

$$\varphi(\alpha) = -\alpha + \alpha(\alpha-5)(2-\alpha)^2$$

Solution:



$$\begin{aligned}\varphi(\alpha) &= -\alpha + \alpha(\alpha-5)(4-4\alpha+\alpha^2)\\ &= -\alpha + \alpha(4\alpha - 4\alpha^2 + \alpha^3 - 20 + 20\alpha - 5\alpha^2)\\ &= -\alpha + 4\alpha^2 - 4\alpha^3 + \alpha^4 - 20\alpha + 20\alpha^2 - 5\alpha^3\\ \varphi(\alpha) &= \alpha^4 - 9\alpha^3 + 24\alpha^2 - 21\alpha\end{aligned}$$

First Wolfe condition: $\varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0)$

$\varphi(\alpha) = \alpha^4 - 9\alpha^3 + 24\alpha^2 - 21\alpha \quad \Rightarrow \varphi(0) = 0$

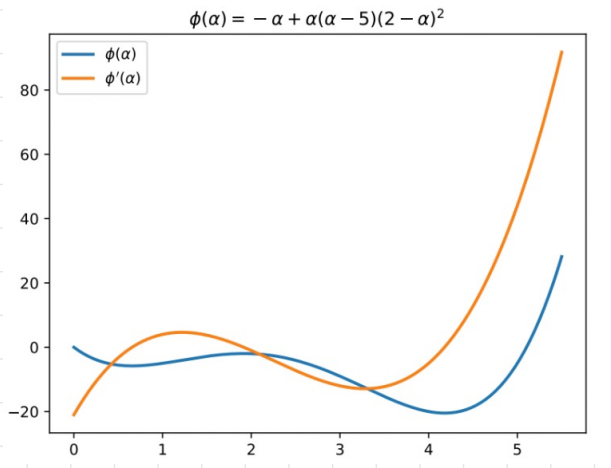
$\varphi'(\alpha) = 4\alpha^3 - 27\alpha^2 + 48\alpha - 21 \quad \Rightarrow \varphi'(0) = -21$

Therefore, the first Wolfe condition is: $\boxed{-\alpha + \alpha(\alpha-5)(2-\alpha)^2 \leq -21 c_1 \alpha}$

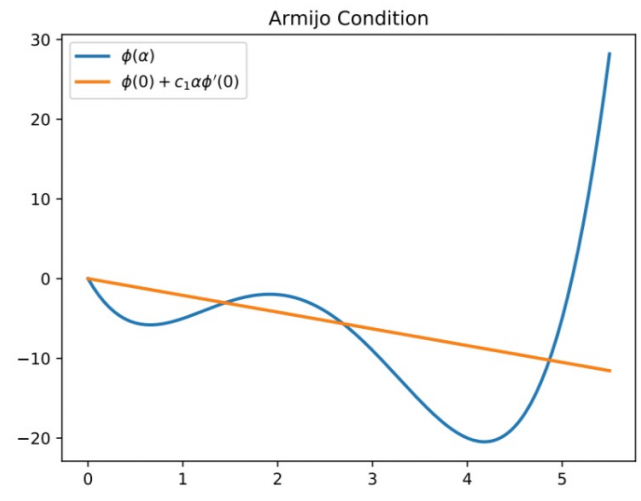
Second Wolfe condition: $\varphi'(\alpha) \geq c_2 \varphi'(0)$

Therefore, the second Wolfe condition is: $\boxed{4\alpha^3 - 27\alpha^2 + 48\alpha - 21 \geq -21 c_2}$

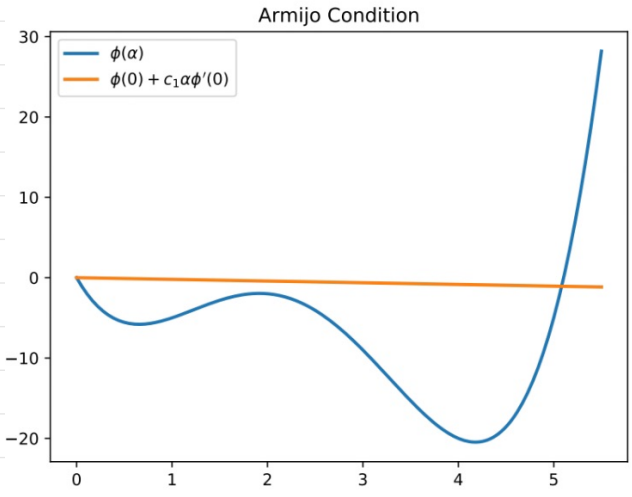
Plot of $\varphi(\alpha)$ and $\varphi'(\alpha)$:



Plot of $\varphi(\alpha)$ and $\varphi(0)+c_1\alpha\varphi'(0)$:

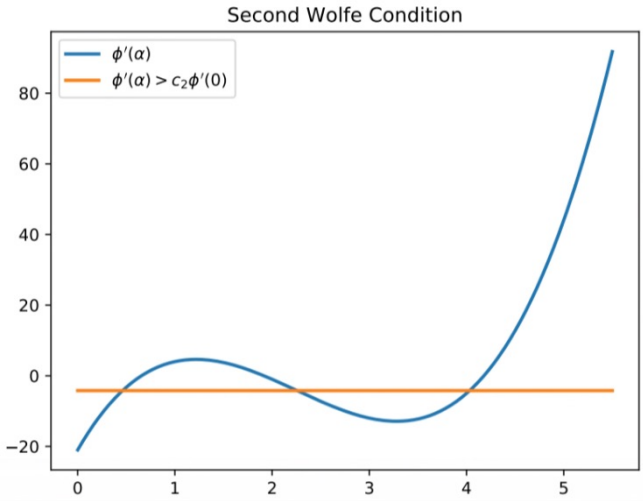


first Wolfe condition
$c_1 = 0.1$

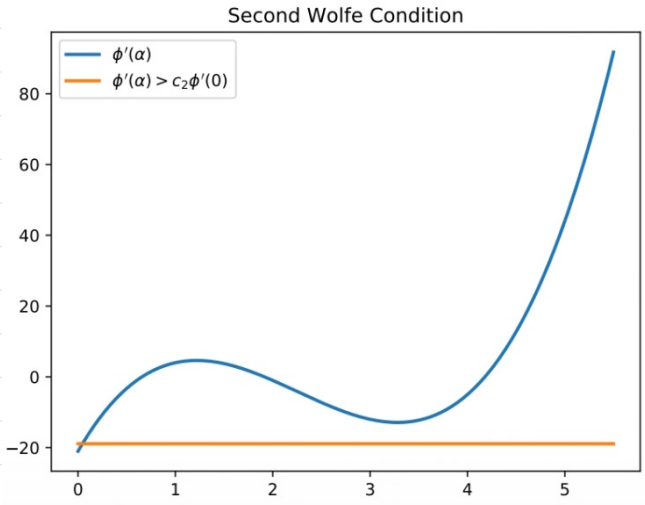


first Wolfe condition
$c_1 = 0.01$

Plot of $\varphi'(\alpha)$ and $\varphi'(\alpha) > c_2\varphi'(0)$:

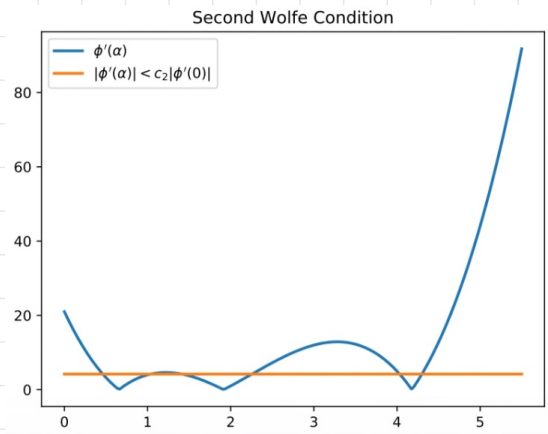


Second Wolfe condition
$c_2 = 0.2$

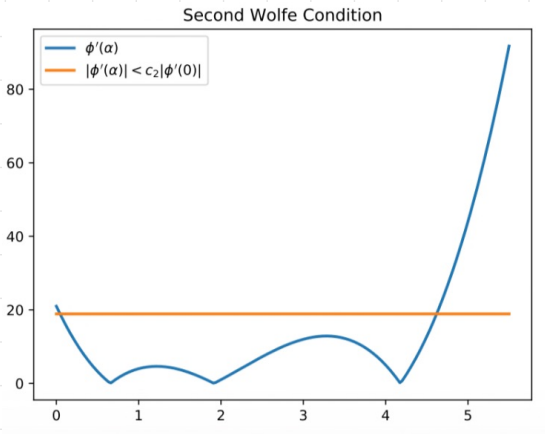


Second Wolfe condition
$c_2 = 0.9$

Plot of $\varphi'(\alpha)$ and $|\varphi'(\alpha)| < c_2|\varphi'(0)|$:



Strong second Wolfe condition
$c_2 = 0.2$



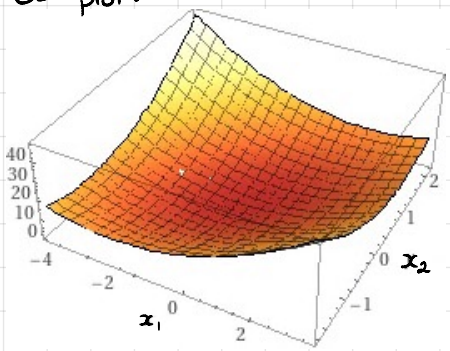
Strong second Wolfe condition
$c_2 = 0.9$

Example:

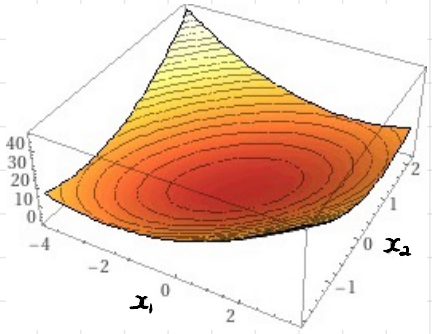
Perform an approximate line search to minimize $f = x_1^2 - 2x_1x_2 + 4x_2^2 + x_1 - 2x_2$ by using the strong Wolfe conditions with $c_1 = 0.0001$ and $c_2 = 0.1$. Start at $[-3, -2]$ and use a search direction of $[1, 2]$. Give a lower and upper limit α that satisfies both strong Wolfe conditions.

Solution:

3D plot:

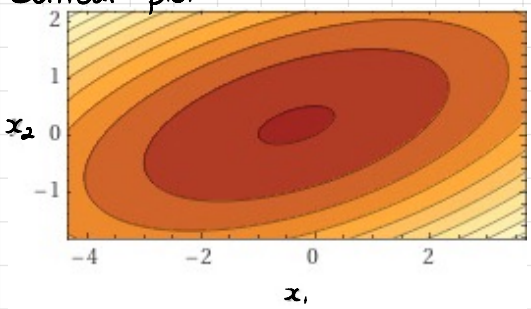


← mesh



← contour lines

Contour plot:



$s_k = [1, 2]$ ← normalize search direction $\Rightarrow$ $s_k = \left[\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\right]$

The line search equation is $\underline{x}_k = \underline{x}_{k-1} + \alpha \underline{s}_k$:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \end{bmatrix} + \alpha \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix} \Rightarrow \begin{aligned} x_1 &= -3 + \frac{\alpha}{\sqrt{5}} \\ x_2 &= -2 + \frac{2\alpha}{\sqrt{5}} \end{aligned}$$

$$\begin{aligned} f(x_1, x_2) &= x_1^2 - 2x_1x_2 + 4x_2^2 + x_1 - 2x_2 \\ &= \left(-3 + \frac{1}{\sqrt{5}}\alpha\right)^2 - 2\left(-3 + \frac{1}{\sqrt{5}}\alpha\right)\left(-2 + \frac{2}{\sqrt{5}}\alpha\right) + 4\left(-2 + \frac{2}{\sqrt{5}}\alpha\right)^2 + \left(-3 + \frac{1}{\sqrt{5}}\alpha\right) - 2\left(-2 + \frac{2}{\sqrt{5}}\alpha\right) \\ &= 9 - \frac{6}{\sqrt{5}}\alpha + \frac{1}{5}\alpha^2 - 2\left(6 - \frac{6}{\sqrt{5}}\alpha - \frac{2}{\sqrt{5}}\alpha + \frac{2}{5}\alpha^2\right) + 4\left(4 - \frac{8}{\sqrt{5}}\alpha + \frac{4}{5}\alpha^2\right) - 3 + \frac{1}{\sqrt{5}}\alpha + 4 - \frac{4}{\sqrt{5}}\alpha \\ &= 9 - \frac{6}{\sqrt{5}}\alpha + \frac{1}{5}\alpha^2 - 12 + \frac{12}{\sqrt{5}}\alpha + \frac{4}{\sqrt{5}}\alpha - \frac{4}{5}\alpha^2 + 16 - \frac{32}{\sqrt{5}}\alpha + \frac{16}{5}\alpha^2 - 3 + \frac{1}{\sqrt{5}}\alpha + 4 - \frac{4}{\sqrt{5}}\alpha \end{aligned}$$

$$\varphi(\alpha) = 14 - 5\sqrt{5}\alpha + \frac{13}{5}\alpha^2$$

1st Wolfe condition (Armijo rule / sufficient decrease):

$$f(\underline{x}_{k-1}+\alpha \underline{s}_k) \leq f(\underline{x}_{k-1}) + c_1 \alpha \underline{s}_k^T \nabla f(\underline{x}_{k-1}) \quad \text{or} \quad \varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0)$$

$\varphi(\alpha) = \frac{13}{5}\alpha^2 - 5\sqrt{5}\alpha + 14 \Rightarrow \varphi(0) = 14$

$\varphi'(\alpha) = \frac{26}{5}\alpha - 5\sqrt{5} \Rightarrow \varphi'(0) = 5\sqrt{5}$

$\varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi'(0) \rightarrow \frac{13}{5}\alpha^2 - 5\sqrt{5}\alpha + 14 \leq 14 + (0.0001)\alpha(5\sqrt{5})$

$\frac{13}{5}\alpha^2 - 11.18\alpha \leq 0$

$\alpha \leq 0, \quad \alpha \leq 4.3$

2nd Strong Wolfe condition (strong curvature condition):

$$|\underline{s}_k^T \nabla f(\underline{x}_{k-1}+\alpha \underline{s}_k)| \leq c_2 |\underline{s}_k^T \nabla f(\underline{x}_{k-1})| \quad \text{or} \quad |\varphi'(\alpha)| \leq c_2 |\varphi'(0)|$$

$|\varphi'(\alpha)| \leq c_2 |\varphi'(0)| \rightarrow \left|\frac{26}{5}\alpha - 5\sqrt{5}\right| \leq 0.1 \left|5\sqrt{5}\right|$

$\left|\frac{26}{5}\alpha - 5\sqrt{5}\right| \leq 1.118$

$\frac{26}{5}\alpha - 5\sqrt{5} \leq 1.118 \qquad \frac{26}{5}\alpha - 5\sqrt{5} \geq -1.118$

$\alpha \leq 2.365, \quad \alpha \geq 1.935$ ← these values are more constraining

$$\boxed{1.935 \leq \alpha \leq 2.365}$$