Numerical Optimisation and Inverse Problems Project 1

Edward Small 10786391

February 2021

Contents

1 Well-defined Minimum of a Quadratic

Take the function $\phi(\alpha) : [0, \alpha_0] \to \mathbb{R}$, and assume it is sufficiently smooth in this interval for the following construction. We can then construct a quadratic estimation of this function over the interval $[0, \alpha_0]$ by the following

$$\phi_q(\alpha) = \left(\frac{\phi(\alpha_0) - \phi(0) - \alpha_0 \phi'(0)}{\alpha_0^2} \right) \alpha^2 + \phi'(0)\alpha + \phi(0) \tag{1}$$

where $\alpha \in [0, \alpha_0]$. We would like to find for what conditions does $\phi_q(\alpha)$ have a well-defined minimum α^* such that $\alpha^* \in]0, \alpha_0[$.

1.1 Geometric Interpretation of the problem

This section is just to help my own understanding, more than anything else, so could be skipped over. However, I believe seeing some examples in action can help to think about how to tackle the problem. Code for the plots is included in the appendix.

Taking ϕ to be linear or quadratic allowed ϕ_q to map the function perfectly, as expected. However, mapping some examples of ϕ when it is cubic reveals a nice visualisation of what the problem here is asking.

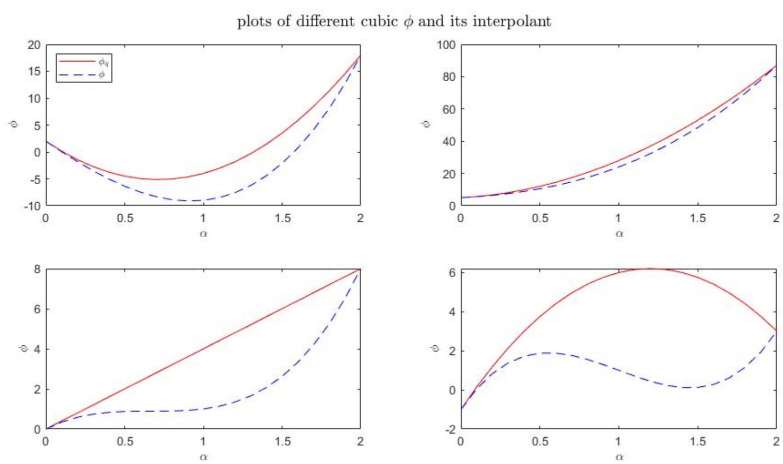


Figure 1: Some of examples of different value for ϕ (blue dotted line) and its corresponding ϕ_q (red solid line) over the interval $[0, 2]$. Top left: $\phi(\alpha) = 5\alpha^3 + 4\alpha^2 - 20\alpha + 2$ Top right: $\phi(\alpha) = 4\alpha^3 + 10\alpha^2 + 5\alpha + 5$ Bottom left: $\phi(\alpha) = 3\alpha^3 - 6\alpha^2 + 4\alpha$ Bottom right: $\phi(\alpha) = 5\alpha^3 - 15\alpha^2 + 12\alpha - 1$

We can see a few different situations here. Clearly ϕ_k does not perfectly match the behaviour of ϕ, but this is expected as it is not as smooth as a cubic function. Visually, some functions have a minimum in the interval, some a maximum, and some it is difficult to tell. So, to re-frame the question, we are asking "What properties does the blue line ϕ need to have (in the given interval) to guarantee that the red line ϕ_q has a minimum (in the given interval?)"

1.2 Quick Observations

Because $\phi_k(\alpha)$ takes the form of a quadratic over the interval $[0, \alpha_0]$, there are a few obvious points that need highlighting.

Firstly, because of the construction of the interpolent ϕ_k it can only have one stationary point (a minimum or a maximum [although it can become linear in special cases, which is shown later]). It should be immediately obvious that

1. $\phi(0) = \phi_k(0)$, the function start point is the same as the interpolant start point
2. $\phi(\alpha_0) = \phi_k(\alpha_0)$, the function end point is the same as the interpolant end point
3. We require $\phi'(0) = \phi_k'(0) < 0$, so negative gradient at the beginning of the interval. In other words, the interpolant should be descending at the earliest point if we have a minimum
4. We require $\phi_k'(\alpha_0) = \frac{2(-\alpha_0\phi'(0)+\phi(\alpha_0)-\phi(0))}{\alpha_0} + \phi'(0) > 0$, the interpolant should be ascending at the latest point. Since $\phi'(0)$ is always negative, we require

$$\frac{2(-\alpha_0\phi'(0) + \phi(\alpha_0) - \phi(0))}{\alpha_0} > |\phi'(0)| \tag{2}$$

5. $\alpha_0 > 0$, so we are defining a positive interval from 0 up to α_0

From here, then, we can already define a condition on two parameters

Condition 1: For a minimum $\alpha^* \in]0, \infty[$ to exist

$$\phi'(0) < 0 \tag{3}$$

This assumption is also given in the question.

Condition 2: For a minimum $\alpha^* > 0$ to exist (assuming condition 1 is met)

$$\alpha_0 > 0 \tag{4}$$

1.3 Behaviour of Quadratic Functions

The behaviour of quadratic functions is pretty well defined, and most of us are used to seeing them in the form of

$$y(x) = ax^2 + bx + c \tag{5}$$

Simply by matching coefficients, we can clearly see that ϕ_q is already in this form, with

- $y(x) = \phi(\alpha)$
- $x = \alpha$
- $a = \frac{\phi(\alpha_0)-\phi(0)-\alpha_0\phi'(0)}{\alpha_0^2}$
- $b = \phi'(0)$
- $c = \phi(0)$

For a generic quadratic y to have a minimum anywhere in $[-\infty, \infty]$ we need $a > 0$. We can confirm this with the Taylor series expansion. If we are at a minimum x^*, then

$$y(x^* + \delta) > y(x^*) \tag{6}$$

for any δ, where

$$y(x + \delta) = y(x) + \delta y'(x) + \frac{\delta^2}{2} y''(x) + ... \tag{7}$$

Basically, any small movement δ in any direction along y should increase the value of y if $x = x^*$. For a generic quadratic, the Taylor series expansion becomes

$$y(x+\delta) = y(x) + \delta(2ax+b) + \frac{\delta^2}{2}2a \tag{8}$$

Clearly, if $a = 0$ then

$$y(x+\delta) = y(x) + \delta b \tag{9}$$

which confirms linearity, with the gradient being dependent on b only (this makes perfect sense if you examine the original equation for y). If $a \neq 0$ then we can examine the point where $y'(x) = 0$, which is when $b = -2ax$ and $x = x^*$ is a turning point.

$$\begin{aligned} y(x^*+\delta) &= y(x^*) + \frac{\delta^2}{2}y''(x^*) \\ &= y(x^*) + \frac{\delta^2}{2}2a \end{aligned} \tag{10}$$

Clearly $\delta^2 > 0$ for $\delta \neq 0$, so whether or not y is strictly increasing/decreasing from the point $y(x^*)$ is dependent on the value of a. For

1. $a < 0$, $y(x^* + \delta) < y(x^*)$, so function is decreasing (has a maximum)
2. $a > 0$, $y(x^* + \delta) > y(x^*)$, so function is increasing (has a minimum)

So, for a minimum to exist at all (even outside of the prescribed limit)

$$\frac{\phi(\alpha_0) - \phi(0) - \alpha_0\phi'(0)}{\alpha_0^2} > 0 \tag{11}$$

Since α_0^2 is always positive (condition 2), we can then simplify this to

$$\phi(\alpha_0) > \phi(0) + \alpha_0\phi'(0) \tag{12}$$

However, this is only a condition to have a minimum anywhere. We need to define a condition that guarantees that, if a minimum does exist, it exists within $]0, \alpha_0[$.

We can find a more specific restriction by turning our attention back to the derivative at $\phi_k'(\alpha_0)$. Recall that

$$\phi_k'(\alpha_0) = \frac{2(-\alpha_0\phi'(0) + \phi(\alpha_0) - \phi(0))}{\alpha_0} + \phi'(0) > 0 \tag{13}$$

Some simple rearranging yields a third condition that relates all the terms to each other

Condition 3: For a minimum $\alpha^* \in]0, \alpha_0[$ to exist, assuming conditions 1 and 2 have been met, we require that

$$\phi(\alpha_0) = \phi_q(\alpha_0) > \phi(0) + \frac{\alpha_0\phi'(0)}{2} \tag{14}$$

Provided that these three conditions are met, the interpolant ϕ_q is guaranteed to have a minimum $\alpha^* \in]0, \alpha_0[$. $\phi(0)$ is almost a free choice, it can technically be any number. However, by choosing $\phi(0)$, $\phi'(0)$ and α_0 (subject to conditions 1 and 2) we automatically choose a lower bound for $\phi(\alpha_0)$ (condition 3). Some code is provided in appendix B where these parameters can be changed to see how they effect one another.

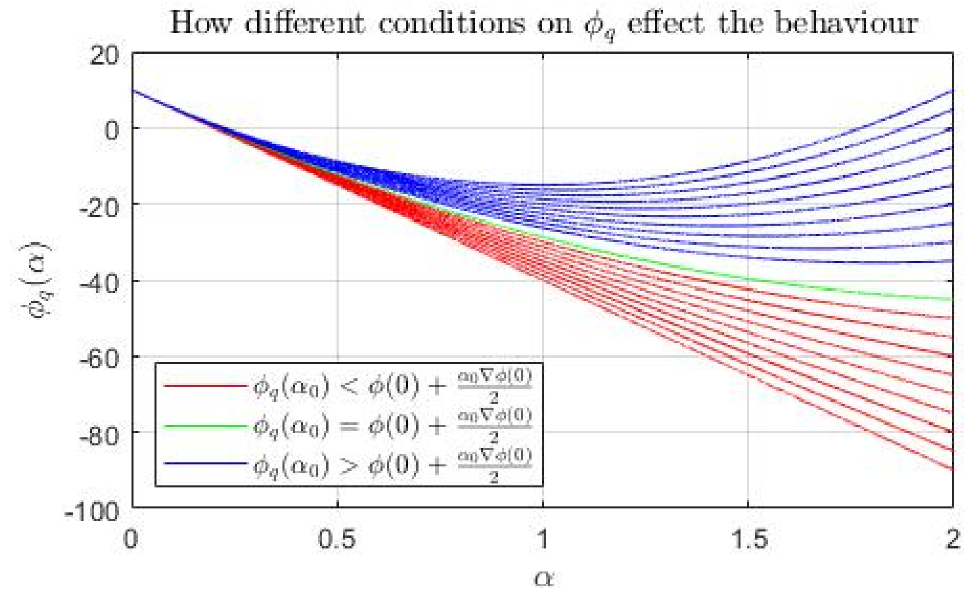


Figure 2: Shows how condition 3 influences the shape of the interpolant if conditions 1 and 2 are met. $\phi(0) = 10$, $\phi'(0) = -50$, $\alpha_0 = 2$

2 Solution to General Constrained Optimisation Problem

Let G be an $n \times n$ symmetric positive-definite matrix, $\boldsymbol{b}$ a given non-zero n-vector, and $\boldsymbol{x} \in \mathbb{R}^n$. By using the first order Karush-Kuhn-Tucker conditions, show that a solution $\boldsymbol{x}$ of the constrained problem

$$\text{Minimize } \frac{1}{2}\boldsymbol{x}^T G\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} \text{ subject to } \boldsymbol{x}^T\boldsymbol{x} = 1 \tag{15}$$

if it exists, needs to satisfy

$$\boldsymbol{x} = -(\mu I + G)^{-1}\boldsymbol{b} \tag{16}$$

for some scalar μ, with I being the $n \times n$ identity matrix

2.1 Considerations

The problem here only has equality constraints. The general form for a quadratic program is

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T G\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c \text{ subject to } A\boldsymbol{x} = \boldsymbol{d} \tag{17}$$

Where our new matrices are defined as A is $m \times n$, $b \in \mathbb{R}^n$, $\boldsymbol{d} \in \mathbb{R}^m$ and $c \in \mathbb{R}$. Using this form, we are then searching for an $\boldsymbol{x}$ where

1. The derivative of the active constraints $\{\nabla c_i(\boldsymbol{x}) | i \in A(\boldsymbol{x})\}$ are linearly independent (i.e. the multiplication of one active constraint by a scalar does not make it equal to another).
2. $\nabla_x L(\boldsymbol{x}, \boldsymbol{\lambda}) = 0$ where
$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i \in A(\boldsymbol{x})} \lambda_i c_i(\boldsymbol{x}) \tag{18}$$
and
$$\nabla_x = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, ..., \frac{\partial}{\partial x_n}\right]^T \tag{19}$$
and $\boldsymbol{\lambda}$ is a vector of Langrange multipliers.
3. $c_i(\boldsymbol{x}) = 0$ for all $i \in E$, so all the constraints are met, or
$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}, \boldsymbol{\lambda}) = 0 \tag{20}$$

To summarise then, we need

$$\nabla L(\boldsymbol{x}, \boldsymbol{\lambda}) = 0 \tag{21}$$

2.2 Applying First Order KKT

For the problem described here, 1. is not a concern, as we only have a single constraint:

$$c_1(\boldsymbol{x}) = x_1x_1 + x_2x_2 + ... + x_nx_n - 1 = \boldsymbol{x}^T\boldsymbol{x} - 1 = 0 \tag{22}$$

Therefore, to match everything up to the general form above: $A = \boldsymbol{x}^T$, $m = 1$, $c = 0$, and $\boldsymbol{d} = 1$. Applying the Langragian then gives the system

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \left(\frac{1}{2}\boldsymbol{x}^T G\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}\right) - \boldsymbol{\lambda}^T(\boldsymbol{x}^T\boldsymbol{x} - 1) \tag{23}$$

and differentiating this expression using the Langragian operators gives

$$\begin{aligned} \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= G\boldsymbol{x} + \boldsymbol{b} - 2\boldsymbol{x}\boldsymbol{\lambda} &&= 0 \\ \nabla_{\boldsymbol{\lambda}} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= \boldsymbol{x}^T\boldsymbol{x} - 1 &&= 0 \end{aligned} \tag{24}$$

It is important to note that $\boldsymbol{\lambda}$ is a scalar (because we only have one active constraint, and we expect a λ_i for each constraint), so take $\boldsymbol{\lambda} = \lambda$. From here, we can define a KKT-system

$$\begin{bmatrix} G & -\boldsymbol{x} \\ \boldsymbol{x}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 2\lambda \end{bmatrix} = \begin{bmatrix} -b \\ 1 \end{bmatrix} \tag{25}$$

From this system we can see that our constraint is still present, as it gives

$$\boldsymbol{x}^T \boldsymbol{x} - 0(-2\lambda) = \boldsymbol{x}^T \boldsymbol{x} = 1 \tag{26}$$

and that

$$G\boldsymbol{x} - 2\lambda\boldsymbol{x} = \boldsymbol{-b} \tag{27}$$

Multiplying by the $n \times n$ identity matrix I gives

$$IG\boldsymbol{x} - 2\lambda I\boldsymbol{x} = -I\boldsymbol{b} \tag{28}$$

Multiplying a matrix by I leaves it unchanged. Take $-2\lambda = \mu$ so

$$G\boldsymbol{x} + \mu I\boldsymbol{x} = \boldsymbol{-b} \tag{29}$$

Rearranging yields

$$\begin{aligned} (\mu I + G)\boldsymbol{x} &= \boldsymbol{-b} \\ \boldsymbol{x} &= -(\mu I + G)^{-1}\boldsymbol{b} \end{aligned} \tag{30}$$

Where μI is an $n \times n$ matrix with $\mu = -2\lambda$ down the leading diagonal, zeroes everywhere else. $\boldsymbol{x}$ must satisfy this expression in order to be a minimum.

3 Minimization in $\mathbb{R}^3$

Minimize

$$f(x_1, x_2, x_3) = f(\mathbf{x}) = -x_1^2 - x_2^2 + x_3^2 \tag{31}$$

such that

$$x_1 + x_2 = -1 \text{ and } x_1^2 + x_2^2 = \frac{1}{2} \tag{32}$$

or

$$\begin{aligned} c_1(\mathbf{x}) &= x_1 + x_2 + 1 &= 0 \\ c_2(\mathbf{x}) &= x_1^2 + x_2^2 - \frac{1}{2} &= 0 \end{aligned} \tag{33}$$

so $E = \{1, 2\}$, $I = \varnothing$

3.1 Unconstrained Optimisation

We can turn the above *constrained problem* into an *unconstrained problem* by using the constraints to remove variables from the function f. For example, from the constraints we know that

$$-x_1^2 - x_2^2 = -\frac{1}{2} \tag{34}$$

and therefore, using the constraint as a substitution, we can define a new function to minimise g such that

$$g(x_3) = -\frac{1}{2} + x_3^2 \tag{35}$$

Since the problem is now unconstrained (as we have no constraint on the value of x_3), we can simply find the point at which $g(x_3)$ can only increase, or in other words when $g'(x_3) = 0$. Since $g'(x_3) = 2x_3$, we can conclude that $x_3 = 0$ at $\boldsymbol{x^*}$. Using this, we can now show that

$$f(x_1, x_2, 0) = -x_1^2 - x_2^2 + 0^2 = -x_1^2 - x_2^2 \tag{36}$$

at $\boldsymbol{x^*}$. Again, this problem is constrained, but it can be unconstrained by using substitution. Since we know that $x_1 + x_2 = -1$ it follows that

$$x_1 = -1 - x_2 \tag{37}$$

Subbing this in for a new function to minimise h yields

$$\begin{aligned} h(x_2) &= -(-1 - x_2)^2 - x_2^2 \\ &= -(1 + 2x_2 + x_2^2) - x_2^2 \\ &= -1 - 2x_2 - 2x_2^2 \end{aligned} \tag{38}$$

Again, because this problem is now unconstrained, the solution of $h'(x_2) = 0$ is satisfactory.

$$h'(x_2) = -2 - 4x_2 = 0 \tag{39}$$

so $x_2 = -\frac{1}{2}$ at $\mathbf{x}^*$. We can then use the relationships from the constraints to find

$$\begin{aligned} x_1 + x_2 &= -1 \\ x_1 &= -1 + \frac{1}{2} \\ &= -\frac{1}{2} \end{aligned} \tag{40}$$

So $\boldsymbol{x}^* = [-\frac{1}{2}, -\frac{1}{2}, 0]^T$, which gives a value of $f(\boldsymbol{x}^*) = -\frac{1}{2}$. This value of $\boldsymbol{x}^*$ fits with the constraints. Another method of solving this would be to use the constraints to find x_1 and x_2 by noting that

$$\begin{aligned} x_1 &= -1 - x_2 \\ x_1^2 &= (-1 - x_2)(-1 - x_2) = 1 + 2x_2 + x_2^2 \end{aligned} \tag{41}$$

which, from the other constraint, leads to

$$\begin{aligned} 1 + 2x_2 + 2x_2^2 - \frac{1}{2} &= 0 \\ (2x_2 + 1)(x_2 + \frac{1}{2}) &= 0 \end{aligned} \tag{42}$$

Showing that $x_2 = -\frac{1}{2}$. We can then use this fact and do the same as above to find that $\mathbf{x}^* = [-\frac{1}{2}, -\frac{1}{2}, 0]^T$.

3.2 Karush-Kuhn-Tucker Optimality Conditions

We will now form a set of optimality conditions for this constrained problem, which can be used to find candidates for $\boldsymbol{x}^*$.

3.2.1 Linear Independence Constrain Qualification

Firstly, the set of gradients of the active constraints at the point $\mathbf{x}^*$ must be linearly independent of each other. We have two constraints in this problem, and they are both equality constraints so are active always. This means that we require that there exists no set $v = \{v_1, v_2\}$ of non-zero scalars where

$$v_1 \nabla c_1(\mathbf{x}^*) + v_2 \nabla c_2(\mathbf{x}^*) = 0 \tag{43}$$

and

$$\nabla c_i(\mathbf{x}^*) \neq 0 \tag{44}$$

Applying

$$\nabla c_1(\mathbf{x}) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \text{ and } \nabla c_2(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \\ 0 \end{bmatrix} \tag{45}$$

3.2.2 KKT conditions

Secondly, if the LICQ condition holds at the minimizer $\boldsymbol{x}^*$ then there exists a vector of Lagrangian multipliers $\boldsymbol{\lambda}^*$ with components λ_i^*, where $i \in E \cup I$ such that

1. $\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0$
2. $c_i(\boldsymbol{x}^*) = 0$ for $i \in E$
3. $c_i(\boldsymbol{x}^*) \geq 0$ for $i \in I$
4. $\lambda_i^* \geq 0$ for $i \in I$
5. $\lambda_i^* c_i(\boldsymbol{x}^*) = 0$ for $i \in E \cup I$ (the active constraints)

Where

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i \in E \cup I} \lambda_i c_i(\boldsymbol{x}) \tag{46}$$

is the Lagrangian function.

We only have two optimality conditions, giving us $\boldsymbol{\lambda} = [\lambda_1, \lambda_2]^T$. Since the constraints for this problem are equality constraints (ie I is empty), we only require conditions 1 and 2 to be met here. We therefore require that

$$c(\boldsymbol{x}^*) = \begin{bmatrix} x_1 + x_2 + 1 \\ x_1^2 + x_2^2 - \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{47}$$

and the KKT optimality conditions $\nabla_{\boldsymbol{x}} L(x_1, x_2, x_3, \lambda_1, \lambda_2) = \nabla f - \lambda_1 \nabla c_1 - \lambda_2 \nabla c_2 = 0$ and $c(x_1, x_2) = 0$ is

$$\begin{bmatrix} -2x_1 - \lambda_1 - 2\lambda_2 x_1 \\ -2x_2 - \lambda_1 - 2\lambda_2 x_2 \\ 2x_3 \end{bmatrix} = \begin{bmatrix} -2x_1(1 + \lambda_2) - \lambda_1 \\ -2x_2(1 + \lambda_2) - \lambda_1 \\ 2x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{48}$$

or

$$\begin{bmatrix} -2x_1 \\ -2x_2 \\ 2x_3 \end{bmatrix} - \lambda_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - \lambda_2 \begin{bmatrix} 2x_1 \\ 2x_2 \\ 0 \end{bmatrix} = 0 \tag{49}$$

3.3 Solving the System

Clearly, the only value of x_3 that satisfies the above equation is $x_3 = 0$. We now turn our attention to first two equations in the system. Rearranging the first two equations gives

$$\begin{aligned} \lambda_1 &= -2x_1(1 + \lambda_2) \\ \lambda_1 &= -2x_2(1 + \lambda_2) \end{aligned} \tag{50}$$

We can then equate these equations to find a relationship between the values of x_1 and x_2

$$\begin{aligned} -2x_1(1 + \lambda_2) &= -2x_2(1 + \lambda_2) \\ x_1 &= x_2 \end{aligned} \tag{51}$$

It is worth noting here that this goes against the LICQ conditions, since, from this relationship, any value of $\boldsymbol{x}$ has a set of scalars v that would violate the linear independence of the $\nabla_{\boldsymbol{x}} c_i(x)$. However, using the constraints and the above relationship would show the original minimum candidate of $\boldsymbol{x}^* = [-\frac{1}{2}, -\frac{1}{2}, 0]$. Observing some plots can help explain this result.

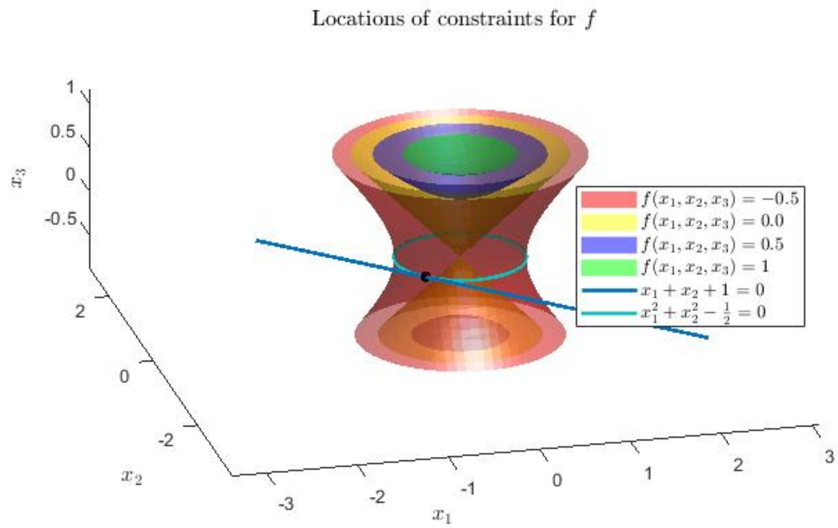


Figure 3: Shows how the constraints relate to each other. The point $\boldsymbol{x}^*$ is marked in black. The circle and line are actually planes that are vertical to the x_3 axis (shown in figure 4). Starting at the black point, moving up this line where the planes touch in any direction increases the value of f, as the inner colours are larger values.

Plotting the function and its constraints reveals that this is, indeed, the minimum. From the constraints, we are free to move in any direction along x_3, but we must be on the plane and touching the cylinder surface (look to figure 4 for context). The red surface is the smallest value of f we can touch whilst satisfying these conditions. Any movement along x_3 moves "further inside" f, which the plot shows increases f. Code to visualise f and its constraints is included in the appendix.

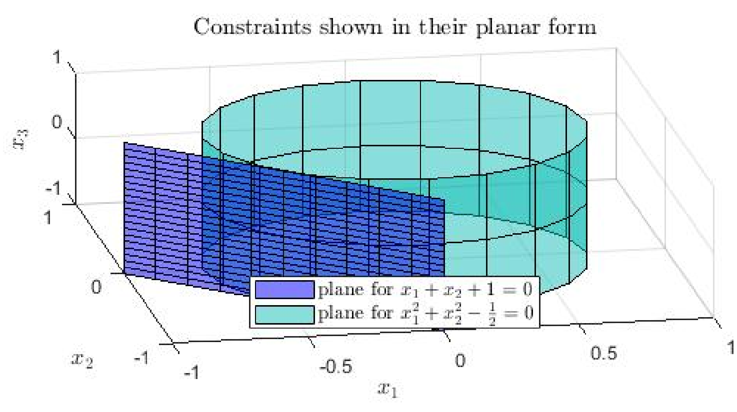


Figure 4: The constraints in figure 3 were restricted to $x_3 = 0$ to clearly show the minimum. However, the constraints are actually planes, and any $\boldsymbol{x}$ that allows for the cylinder and the plane to touch meets the constraints. In other words, satisfying the constrains means $x_1 = -\frac{1}{2}$, $x_2 = -\frac{1}{2}$ and x_3 is a free choice, with $x_3 = 0$ being the minimum for f

References

[1] V. Jeyakumar, A. M. Rubinov, Z. Y. Wu (2007) ‘Non-convex quadratic minimization problems with quadratic constraints: global optimality conditions’, *Springer-Verlag*, pp. 521–540.

[2] Johannes Jahn (2017) ‘Karush–Kuhn–Tucker Conditions in Set Optimization, *Springer Science and Business Media New York*, pp. 712–714.

[3] Monique Laurent, Franz Rendl (2005) ‘Discrete optimization - Semidefinite Programming and Integer Programming (Algorithms), *Elsevier B.V.*, pp. 399–402.

A Matlab Code for Plots from Section 1.1

```
% lower and upper bounds for interval
a = 0;
b = 2;

%cubic example 4
phi4 = @(x) 5 .*x.^3 + -15 .* x.^2 + 12 .* x - 1;
phi4_dash = @(x) 15.*x.^2 -30.*x + 12;

%cubic example 3
phi3 = @(x) 2 -20.*x + 4.*x.^2 + 5.*x.^3;
phi3_dash = @(x)  -20 + 8.*x + 15.*x.^2;

%cubic example 2
phi2 = @(x) 4 .*x.^3 + 10 .* x.^2 + 5 .* x + 5;
phi2_dash = @(x) 12.*x.^2 + 20.*x + 5;

%cubic example 1
phi1 = @(x) 3 .*x.^3 - 6 .* x.^2 + 4 .* x;
phi1_dash = @(x) 9.*x.^2 - 12.*x + 4;

%interpolant
interp = @(phi, phi_dash, x) ...
    ((phi(b) - phi(a) - b * phi_dash(0))./b^2) ...
    * x.^2 + phi_dash(0) .* x + phi(0);

interval = [a:0.1:b];

figure(1)
subplot(2, 2, 1)
plot(interval, interp(phi3, phi3_dash, interval), '-r')
hold on
plot(interval, phi3(interval), '--b')
xlabel('$\alpha$', 'interpreter','latex');
ylabel('$\phi$', 'interpreter','latex');
legend('$\phi_q$', '$\phi$', 'interpreter','latex', ...
    'Location', 'northwest')
subplot(2, 2, 2)
plot(interval, interp(phi2, phi2_dash, interval), '-r')
hold on
plot(interval, phi2(interval), '--b')
xlabel('$\alpha$', 'interpreter','latex');
ylabel('$\phi$', 'interpreter','latex');
subplot(2, 2, 3)
plot(interval, interp(phi1, phi1_dash, interval), '-r')
hold on
plot(interval, phi1(interval), '--b')
xlabel('$\alpha$', 'interpreter','latex');
ylabel('$\phi$', 'interpreter','latex');
subplot(2, 2, 4)
plot(interval, interp(phi4, phi4_dash, interval), '-r')
hold on
plot(interval, phi4(interval), '--b')
xlabel('$\alpha$', 'interpreter','latex');
ylabel('$\phi$', 'interpreter','latex');
sgtitle('plots of different cubic $\phi$ and its interpolant', ...
    'interpreter','latex')
```

A.1 Matlab Code for Quadratic Interpolator Minimum Check

```
%% define the interpolation function
interp = @(phi_0, phi_a, phidash_0, a_0, a) ...
    ((phi_a - phi_0 - a_0 .* phidash_0) ./ a_0^2) .* a.^2 + ...
```

```
    phidash_0 .* a + phi_0;

%% define parameters
a=0; %lower bound of interval
a_0=2; %upper bound of interval (must be > 0)
phi_0=-10; %value of phi at lower bound (unconstrained)
phidash_0=5; %value of gradient at phi(0) (must be < 0)
phi_a_adj = [-1:0.1:1]; %adjuster to observe changes in phi(a)
phi_a=phi_0 + (phidash_0 * a_0)/2 + phi_a_adj; %value of phi at upper bound

%% evaluate the interpolant for different values of phi_a
test = zeros(1, length(phi_a));
step_size = 0.000001;
alpha=[a:step_size:a_0]; %define interval
phi = zeros(length(test), length(alpha));

%% check that we meet constraints to have a minimum inside the interval
disp('For phi_0 + (phidash_0 * a_0)/2 = ' + string(phi_0 + (phidash_0 * a_0)/2))
for i=1:length(phi_a_adj)
    phi(i, :) = interp(phi_0, phi_a(i), phidash_0, a_0, alpha); %get phi over interval
    [a, b] = min(phi(i, :));
    if b < (a_0/step_size) && b > 1
        disp('phi_a=' + string(phi_a(i)) + ' has minimum in interval')
    else
        disp('phi_a=' + string(phi_a(i)) + ' has no minimum in interval')
    end
end
% this breaks down if step size is too large compared to the gradient
% but it gives a good general idea

%% plot the different interpolants
for i=1:(length(phi_a_adj)-1)/2
    test(i) = plot(alpha, phi(i, :), '-r'); %plot the interpolant
    hold on
end
test((length(phi_a_adj)+1)/2) = plot(alpha, phi(i, :), '-g'); %plot the interpolant
hold on
for i=(length(phi_a_adj)+1)/2+1:length(phi_a_adj)
    test(i) = plot(alpha, phi(i, :), '-b'); %plot the interpolant
    hold on
end

legend([test(1), test(11), test(12)], '$\phi_q(\alpha_0) < \phi(0) + \frac{\alpha_0\nabla\phi(0)}{2}$', ...
'$\phi_q(\alpha_0) = \phi(0) + \frac{\alpha_0\nabla\phi(0)}{2}$', ...
'$\phi_q(\alpha_0) > \phi(0) + \frac{\alpha_0\nabla\phi(0)}{2}$', 'interpreter','latex','FontSize',10, ...
'location', 'southwest');
title('How different conditions on $\phi_q$ effect the behaviour','interpreter','latex','FontSize',12)
xlabel('$\alpha$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
ylabel('$\phi_q(\alpha)$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
```

A.2 Some Outputs

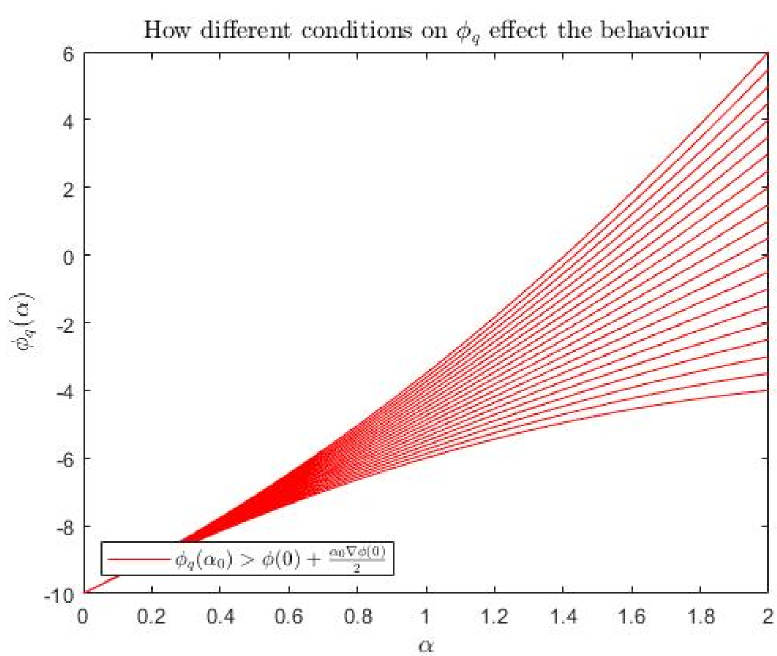
A.2.1 $\alpha_0 = 2$, $\phi(0) = 10$, $\phi'(0) = -50$, $\phi(\alpha_0) \in [-40.1, -39.9]$

```
For phi_0 + (phidash_0 * a_0)/2 = -40
phi_a=-40.1 has no minimum in interval
phi_a=-40.09 has no minimum in interval
phi_a=-40.08 has no minimum in interval
phi_a=-40.07 has no minimum in interval
phi_a=-40.06 has no minimum in interval
phi_a=-40.05 has no minimum in interval
phi_a=-40.04 has no minimum in interval
phi_a=-40.03 has no minimum in interval
phi_a=-40.02 has no minimum in interval
phi_a=-40.01 has no minimum in interval
phi_a=-40 has no minimum in interval
phi_a=-39.99 has minimum in interval
phi_a=-39.98 has minimum in interval
phi_a=-39.97 has minimum in interval
phi_a=-39.96 has minimum in interval
phi_a=-39.95 has minimum in interval
phi_a=-39.94 has minimum in interval
phi_a=-39.93 has minimum in interval
phi_a=-39.92 has minimum in interval
phi_a=-39.91 has minimum in interval
phi_a=-39.9 has minimum in interval
```

Here $\phi(0) + \frac{\phi'(0)*\alpha_0}{2} = -40$, and conditions 1 and 2 are met. For $\phi(\alpha_0) \in [-40.1, -40]$ condition 3 is not met, so there is no minimum detected. However, for $\phi(\alpha_0) \in]-40, -39.9]$ condition 3 is met so a minimum is detected.

A.2.2 $\alpha_0 = 2$, $\phi(0) = -10$, $\phi'(0) = 5$, $\phi(\alpha_0) \in [-4, 6]$

```
For phi_0 + (phidash_0 * a_0)/2 = -5
phi_a=-4 has no minimum in interval
phi_a=-3.5 has no minimum in interval
phi_a=-3 has no minimum in interval
phi_a=-2.5 has no minimum in interval
phi_a=-2 has no minimum in interval
phi_a=-1.5 has no minimum in interval
phi_a=-1 has no minimum in interval
phi_a=-0.5 has no minimum in interval
phi_a=0 has no minimum in interval
phi_a=0.5 has no minimum in interval
phi_a=1 has no minimum in interval
phi_a=1.5 has no minimum in interval
phi_a=2 has no minimum in interval
phi_a=2.5 has no minimum in interval
phi_a=3 has no minimum in interval
phi_a=3.5 has no minimum in interval
phi_a=4 has no minimum in interval
phi_a=4.5 has no minimum in interval
phi_a=5 has no minimum in interval
phi_a=5.5 has no minimum in interval
phi_a=6 has no minimum in interval
```



Here $\phi(0) + \frac{\phi'(0) * \alpha_0}{2} = -5$. So, For $\phi(\alpha_0) \in [-4, 6]$ condition 3 is met. Condition 2 is also met, as we have a positive interval defined by α_0. However, condition 1 is not met, as $\phi'(0) \geq 0$, so we cannot have a mininum anywhere in the interval.

B Matlab Code for Plots from Section 3

B.1 f isosurface plot and 2d constraints

```
%% define functions and get variables
func = @(x,y,z) -x.^2 -y.^2 + z.^2;
[x,y,z] = meshgrid(-2:0.1:2, -2:0.1:2, -2:0.1:2);
v = func(x,y,z);
con1 = @(var) -1 - var;
con2 = @(var) sqrt(1/2 - var.^2);

step = [-2:0.1:2];
con1y = con1(step);
con2y = con2(step);

%% compute isosurfaces (f) and lines (constraints), and min
p1 = patch(isosurface(x,y,z,v,-0.5));
hold on
p2 = patch(isosurface(x,y,z,v,0));
hold on
p3 = patch(isosurface(x,y,z,v,0.5));
hold on
p4 = patch(isosurface(x,y,z,v,1));
hold on

line = plot(step, con1y);
circ = ezplot(@(x,y) (x).^2 + (y).^2 -0.5);
mark = plot(-0.5, -0.5, '.k', 'MarkerSize', 20)

%% plot and colour isonormals at desired levels
isonormals(x,y,z,v,p1);
isonormals(x,y,z,v,p2);
isonormals(x,y,z,v,p3);
isonormals(x,y,z,v,p4);
p1.FaceColor = 'red';
p2.FaceColor = 'yellow';
p3.FaceColor = 'blue';
p4.FaceColor = 'green';
p1.EdgeColor = 'none';
p2.EdgeColor = 'none';
p3.EdgeColor = 'none';
p4.EdgeColor = 'none';

%% fix lighting and transparency
daspect([1,1,1])
view(3); axis tight
camlight
lighting flat
alpha 0.5

%% plot beautify
set(circ,'LineWidth',2);
set(line, 'LineWidth',2);
legend([p1, p2, p3, p4, line, circ], '$f(x_1, x_2, x_3) = -0.5$', ...
    '$f(x_1, x_2, x_3) = 0.0$', ...
    '$f(x_1, x_2, x_3) = 0.5$', ...
    '$f(x_1, x_2, x_3) = 1$', ...
    '$x_1 + x_2 + 1 = 0$', ...
    '$x_1^2 + x_2^2 - \frac{1}{2} = 0$', 'interpreter','latex','FontSize',10, ...
'location', 'east');
title('Locations of constraints for $f$','interpreter','latex','FontSize',12)
xlabel('$x_1$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
ylabel('$x_2$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
zlabel('$x_3$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
```

B.2 Plot of planar constraints

```
%% plane plot
figure(2)
[x z] = meshgrid(-1:0.1:1); % Generate x and y data
y = -1 - x; % Solve for z data
plane = surf(x,y,z); %Plot the surface
set(plane,'FaceColor',[0 0 1],'FaceAlpha',0.5);
hold on
r = sqrt(0.5);
[X,Y,Z] = cylinder(r);
circ1 = surf(X,Y,Z);
circ2 = surf(X,Y,-Z);
alpha 0.5
xlim([-1,1])
ylim([-1,1])
zlim([-1,1])
legend([plane, circ1], 'plane for $x_1 + x_2 + 1 = 0$', ...
    'plane for $x_1^2 + x_2^2 - \frac{1}{2} = 0$', ...
    'interpreter','latex','FontSize',10, ...
    'location', 'south');
title('Constraints shown in their planar form','interpreter','latex','FontSize',12)
xlabel('$x_1$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
ylabel('$x_2$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
zlabel('$x_3$','interpreter','latex','FontSize',12,'FontWeight', 'bold')
```