PURDUE UNIVERSITY · CS 52000
COMPUTATIONAL OPTIMIZATION

HOMEWORK 1
Mathieu Gaillard
January 21, 2019

Collaborators

1. Wikipedia
2. https://math.stackexchange.com/questions/2880568
3. Yichen Sheng

Problem 1: Some quick, simple theory

Question 1:

What is $1/\log(x)$ when $x = 10^9$?

$$\frac{1}{\log(10^9)} = \frac{1}{9}$$

What is the limit of the sequence $1/\log(x)$ as $x \to \infty$?

$$\lim_{x\to\infty} \frac{1}{\log(x)} = 0$$

Question 2:

Show, using the definition, that the sequence $1 + k^{-k}$ converges superlinearly to 1.

$$X_k = 1 + k^{-k}$$

According to the definition of the superlinear convergence, we want to show that:

$$\lim_{x\to\infty} \frac{\|X_{k+1} - X^*\|}{\|X_k - X^*\|} = 0$$

Let's transform this expression to something easier to deal with.

$$\begin{aligned}\frac{\|X_{k+1} - X^*\|}{\|X_k - X^*\|} &= \frac{\|1 + (k+1)^{-k-1} - 1\|}{\|1 + k^{-k} - 1\|} \\ &= \frac{(k+1)^{-k-1}}{k^{-k}} \quad \text{because} \quad k > 0 \\ &= \frac{1}{k} \times \left(\frac{k+1}{k}\right)^{-k-1}\end{aligned}$$

This form is now suitable for computing the limit. Because:

$$\lim_{x\to\infty} \frac{1}{k} = 0 \quad \text{and} \quad \lim_{x\to\infty} \left(\frac{k+1}{k}\right)^{-k-1} = e^{-1} \quad \text{(proved on page 2)}$$

Therefore:

$$\lim_{x\to\infty} \frac{\|X_{k+1} - X^*\|}{\|X_k - X^*\|} = \lim_{x\to\infty} \frac{1}{k} \times \left(\frac{k+1}{k}\right)^{-k-1} = 0$$

Q.E.D.

Annex to question 2:

Concerning the limit of $\left(\frac{k+1}{k}\right)^{-k-1}$, following is the proof. First, we modify it, so that the limit is easier to compute.

$$\left(\frac{k+1}{k}\right)^{-k-1} = e^{-\ln\left(\frac{k+1}{k}\right)(k+1)}$$

We want to show that the limit of $\ln\left(\frac{k+1}{k}\right)(k+1)$ is 1.

$$\begin{aligned}
\lim_{x\to\infty} \ln\left(\frac{k+1}{k}\right)(k+1) &= \lim_{x\to\infty} k\ln\left(\frac{k+1}{k}\right) + \ln\left(\frac{k+1}{k}\right) \\
&= \lim_{x\to\infty} k\ln\left(\frac{k+1}{k}\right) \quad \text{because} \quad \lim_{x\to\infty}\left(\frac{k+1}{k}\right) = 0 \\
&= \lim_{x\to\infty} \frac{\ln\left(\frac{k+1}{k}\right)}{\frac{1}{k}}
\end{aligned}$$

Then, we use l'Hôpital's rule:

$$\begin{aligned}
\lim_{x\to\infty} \frac{\ln\left(\frac{k+1}{k}\right)}{\frac{1}{k}} &= \lim_{x\to\infty} \frac{\frac{1}{k+1} - \frac{1}{k}}{-\frac{1}{k^2}} \\
&= \lim_{x\to\infty} \frac{k^2}{k} - \frac{k^2}{k+1} \\
&= \lim_{x\to\infty} \frac{k^2(k+1) - k^3}{k(k+1)} \\
&= \lim_{x\to\infty} \frac{k^2}{k^2+k} \\
&= 1
\end{aligned}$$

Therefore:

$$\lim_{x\to\infty}\left(\frac{k+1}{k}\right)^{-k-1} = e^{-1}$$

Question 3:

Suppose we have a sequence $X_{k+1} = \sqrt{X_k}$. Show that this sequence converges for all positive inputs. Show further the rate.

First, we express the recurrence relation as a closed form. Following is the idea for X_2:

$$X_2 = \sqrt{X_1} = \sqrt{\sqrt{X_0}} = (X_0)^{\frac{1}{2}\frac{1}{2}} = (X_0)^{\frac{1}{4}} = \sqrt[4]{X_0}$$

When we generalize, we have the following closed-form:

$$X_{k+1} = \sqrt[2^k]{X_0} = (X_0)^{\frac{1}{2^k}}$$

To study the convergence of this sequence, we study the convergence of the sequence of functions $f_k(x) = x^{\frac{1}{2^k}}$ on $\mathcal{R}^+$. We only study the point-wise convergence,

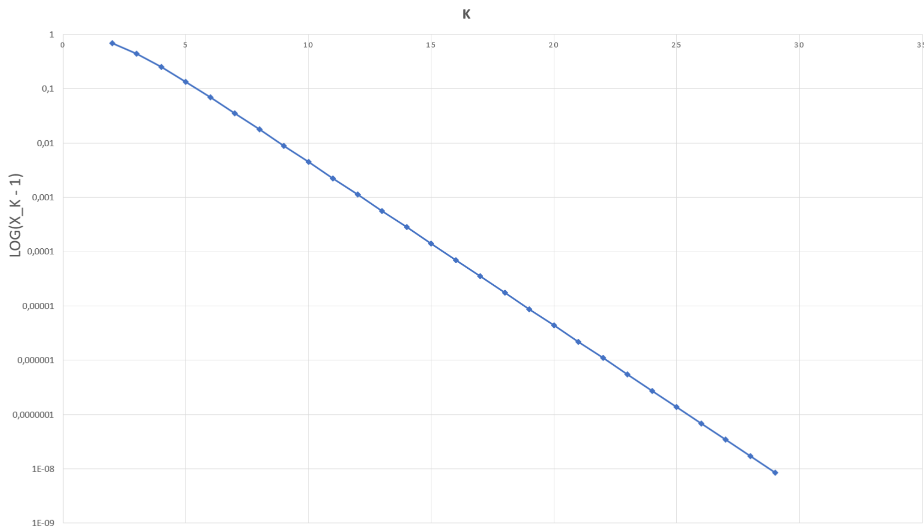

Figure 1: Plot of $\log \|X_k - 1\|$ according to k. We can see a linear relation.

as it is enough for our application.

$$\mathcal{R}^+ : \lim_{k\to\infty} f_k(x) = f(x)$$
$$x = 0 : \lim_{k\to\infty} 0^{\frac{1}{2^k}} = 0$$
$$x > 0 : \lim_{k\to\infty} x^{\frac{1}{2^k}} = 1$$

Therefore:

$$f(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x > 0 \end{cases}$$

And:

$$\lim_{k\to\infty} X_k = 1$$

Another way to prove this result is to use the Banach fixed-point theorem. $\mathcal{R}^+$ is complete. The square root function is stable on $]0;1[$, we can prove that with simple inequalities. It is also a contraction mapping on $[1;+\infty[$, we can prove that with the mean value inequality. Thus, the sequence X_k converges to the fixed-point, which is in $x = 1$ because $\sqrt{1} = 1$.

Concerning the converging rate, by plotting it on Microsoft Excel (see 1), we conjecture that it is a Q-linear convergence with a rate of 1/2.

We want to show that:

$$\frac{\| \sqrt[2^{k+1}]{x} - 1\|}{\| \sqrt[2^k]{x} - 1\|} \leq \rho \quad \text{for all } k \text{ sufficiently large}$$

For that we compute:

$$\lim_{k\to\infty} \frac{\| \sqrt[2^{k+1}]{x} - 1\|}{\| \sqrt[2^k]{x} - 1\|}$$

We use a change of variable $t = \sqrt[2^k]{x}$ and $x = t^{2^k}$, which are both converging expressions. It has been proved earlier that $t \to 1$.

$$\lim_{k\to\infty} \frac{\| \sqrt[2^{k+1}]{x} - 1\|}{\| \sqrt[2^k]{x} - 1\|} = \lim_{t\to 1} \frac{\|\sqrt{t} - 1\|}{\|t - 1\|}$$

We know that in the vicinity of 0, $\sqrt{1+x} \sim (1+\frac{x}{2})$. So, around 1, $\sqrt{x} \sim (1+\frac{x-1}{2})$

$$\lim_{t\to 1} \frac{\|\sqrt{t}-1\|}{\|t-1\|} = \lim_{t\to 1} \frac{\|1+\frac{t-1}{2}-1\|}{\|t-1\|} = \lim_{t\to 1} \frac{\frac{1}{2}\|t-1\|}{\|t-1\|} = \frac{1}{2}$$

Q.E.D.

The sequence $X_{k+1} = \sqrt{X_k}$ converges linearly with a rate 1/2.

Problem 2: Convergence theory

Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x) = \|x\|^2$. Show that the sequence of iterates $\{\mathbf{x}_k\}$ defined by:

$$\mathbf{x}_k = \left(1+\frac{1}{2^k}\right)\begin{bmatrix}\cos k\\ \sin k\end{bmatrix}$$

satisfies $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ for $k = 0, 1, \ldots$. Show that every point on the unit circle is a limit point for $\mathbf{x}_k$.

Let's compute the values of $f(\mathbf{x}_k)$ and $f(\mathbf{x}_{k+1})$.

$$\begin{aligned} f(\mathbf{x}_k) &= \left\| \left(1+\frac{1}{2^k}\right)\begin{bmatrix}\cos k\\ \sin k\end{bmatrix} \right\|^2 \\ &= \left(1+\frac{1}{2^k}\right)^2 (\cos^2 k + \sin^2 k) \\ &= \left(1+\frac{1}{2^k}\right)^2 \end{aligned}$$

In the same way:

$$f(\mathbf{x}_{k+1}) = \left(1+\frac{1}{2^{k+1}}\right)^2$$

By construction, we can show that $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ is true for $k = 0, 1, \ldots$

$$\begin{aligned} k &< k+1 \\ 2^k &< 2^{k+1} \\ \frac{1}{2^k} &> \frac{1}{2^{k+1}} \\ 1+\frac{1}{2^k} &> 1+\frac{1}{2^{k+1}} \\ \left(1+\frac{1}{2^k}\right)^2 &> \left(1+\frac{1}{2^{k+1}}\right)^2 \\ f(\mathbf{x}_k) &> f(\mathbf{x}_{k+1}) \end{aligned}$$

Q.E.D.

Now, we want to show that every point on the unit circle is a limit point for $\mathbf{x}_k$. In other words, $\mathbf{x}_k$ is dense on the unit circle. Thanks to the first part of the question, we know that $\mathbf{x}_k$ can be as close as needed to the unit circle because $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ and $\lim_{k\to\infty} f(\mathbf{x}_k) = 1$. Basically, if we are not close enough, we know that we just need to increase k.

We notice that:

$$\mathbf{x}_k = \left(1+\frac{1}{2^k}\right)\begin{bmatrix}\cos k\\ \sin k\end{bmatrix} = \left(1+\frac{1}{2^k}\right)\begin{bmatrix}\cos(k \mod 2\pi)\\ \sin(k \mod 2\pi)\end{bmatrix}$$

We simply need to prove that $\{k \mod 2\pi : n \in \mathbb{N}\}$ is dense in $[0; 2\pi]$. Intuitively the idea is that because 2π is irrational, with k increasing, $k \mod 2\pi$ is always going to land somewhere else in the interval. Thus, for any point q in the interval, it is always possible to find a k so that $k \mod 2\pi$ is as close as needed to q. A better proof is given later. Back to the initial problem, we want to show that for any point on the circle, we can find a subsequence that converges to this point. We choose this point $p = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$. Then, because $\{k \mod 2\pi : n \in \mathbb{N}\}$ is dense in $[0; 2\pi]$, there exists a subsequence of integers n_k such that $\theta = \lim_{k\to\infty} n_k \mod 2\pi$. Finally, because cos and sin are continuous, x_{n_k} converges to p.

I don't have enough time before the deadline to formalize the proof that $\{k \mod 2\pi : n \in \mathbb{N}\}$ is dense in $[0; 2\pi]$. Here are the two inspirations I found for this problem:

1. https://math.stackexchange.com/questions/39299
2. https://math.stackexchange.com/questions/1916529

We can use the Dirichlet Approximation Theorem to get a multiple of 2π that makes $k \mod 2\pi$ close enough to θ. Or we can use the fact that $\frac{1}{2\pi}$ is irrational so $\mathbb{Z} + 2\pi\mathbb{Z}$ is dense in $\mathbb{R}$. It is the same principal as when we show that cos is dense in $[-1; 1]$.

Problem 3: Raptors in space

Question 1:

Modify the the Raptor chase example function to compute the survival time of a human in a three-dimensional raptor problem. Show your modified function, and show the survival time when running directly at the slow raptor.

Following is my code. I added one raptor and used 3D vectors instead of 2D vectors. Instead of one angle, I used two angles: latitude and longitude.

```
using Plots
plotly(size = (1280, 1024));

vhuman = 6.0;
vraptor0 = 10.0; # the slow raptor velocity in m/s
vraptor = 15.0; # the regular raptors velocity in m/s

raptor_distance = 20.0;

raptor_min_distance = 0.2; # a raptor within 20 cm can attack
tmax = 3.0; # the maximum time in seconds
nsteps = 100000;
```

This function will compute the derivatives of the positions of the human and the raptors

```
function compute_derivatives(theta, phi, h, r0, r1, r2, r3)
  dh = [sin(phi) * cos(theta),
        sin(phi) * sin(theta),
        cos(phi)] * vhuman;
  dr0 = (h-r0)/norm(h-r0)*vraptor0;
  dr1 = (h-r1)/norm(h-r1)*vraptor;
  dr2 = (h-r2)/norm(h-r2)*vraptor;
  dr3 = (h-r3)/norm(h-r3)*vraptor;
  return dh, dr0, dr1, dr2, dr3;
end
```

This function will use forward Euler to simulate the Raptors

```
function simulate_raptors(theta, phi; output::Bool = true)
  # initial positions
  h = [0.0, 0.0, 0.0];
  r0 = [1.0, 0.0, 0.0]*raptor_distance;
  r1 = [-1.0/3.0,  sqrt(8.0)/3.0,            0.0]*raptor_distance;
  r2 = [-1.0/3.0, -sqrt(2.0)/3.0,  sqrt(2.0/3.0)]*raptor_distance;
  r3 = [-1.0/3.0, -sqrt(2.0)/3.0, -sqrt(2.0/3.0)]*raptor_distance;

  # how much time elapsed
  dt = tmax/nsteps;
  t = 0.0;

  hhist = zeros(3,nsteps+1);
  r0hist = zeros(3,nsteps+1);
  r1hist = zeros(3,nsteps+2);
  r2hist = zeros(3,nsteps+2);
  r3hist = zeros(3,nsteps+2);

  hhist[:,1] = h;
  r0hist[:,1] = r0;
  r1hist[:,1] = r1;
  r2hist[:,1] = r2;
  r3hist[:,1] = r3;

  for i=1:nsteps
    dh, dr0, dr1, dr2, dr3 = compute_derivatives(theta,
                                phi, h, r0, r1, r2, r3);

    h += dh*dt;
    r0 += dr0*dt;
    r1 += dr1*dt;
    r2 += dr2*dt;
    r3 += dr3*dt;
    t += dt;

    hhist[:,i+1] = h;
    r0hist[:,i+1] = r0;
    r1hist[:,i+1] = r1;
    r2hist[:,i+1] = r2;
    r3hist[:,i+1] = r3;

    if norm(r0-h) <= raptor_min_distance ||
      norm(r1-h) <= raptor_min_distance ||
      norm(r2-h) <= raptor_min_distance ||
      norm(r3-h) <= raptor_min_distance
      if output
        @printf("The raptors caught the human in \%f seconds\n", t);
      end

      # truncate the history
      hhist = hhist[:,1:i+1];
      r0hist = r0hist[:,1:i+1];
      r1hist = r1hist[:,1:i+1];
      r2hist = r2hist[:,1:i+1];
      r3hist = r3hist[:,1:i+1];

      break
    end
  end
  return hhist, r0hist, r1hist, r2hist, r3hist;
end
```

This function will display the simulation in a 3D plot.

```
function show_raptors(theta, phi; args...)
  hhist, r0h, r1h, r2h, r3h = simulate_raptors(theta, phi; args...);
  plot(vec(hhist[1,:]), vec(hhist[2,:]), vec(hhist[3,:]),linewidth=3);
  plot!(vec(r0h[1,:]), vec(r0h[2,:]), vec(r0h[3,:]),color=:red);
  plot!(vec(r1h[1,:]), vec(r1h[2,:]), vec(r1h[3,:]),color=:red);
  plot!(vec(r2h[1,:]), vec(r2h[2,:]), vec(r2h[3,:]),color=:red);
  plot!(vec(r3h[1,:]), vec(r3h[2,:]), vec(r3h[3,:]),color=:red);
  plot!(xlim=[-6.67, 20.0], ylim=[-9.43, 18.9],zlim=[-16.3, 16.3]);
  # 3D annotations are not supported
  title!(@sprintf("Survival time = \%f sec",(length(hhist[2,:]) - 1)*tmax/nsteps));
end
```

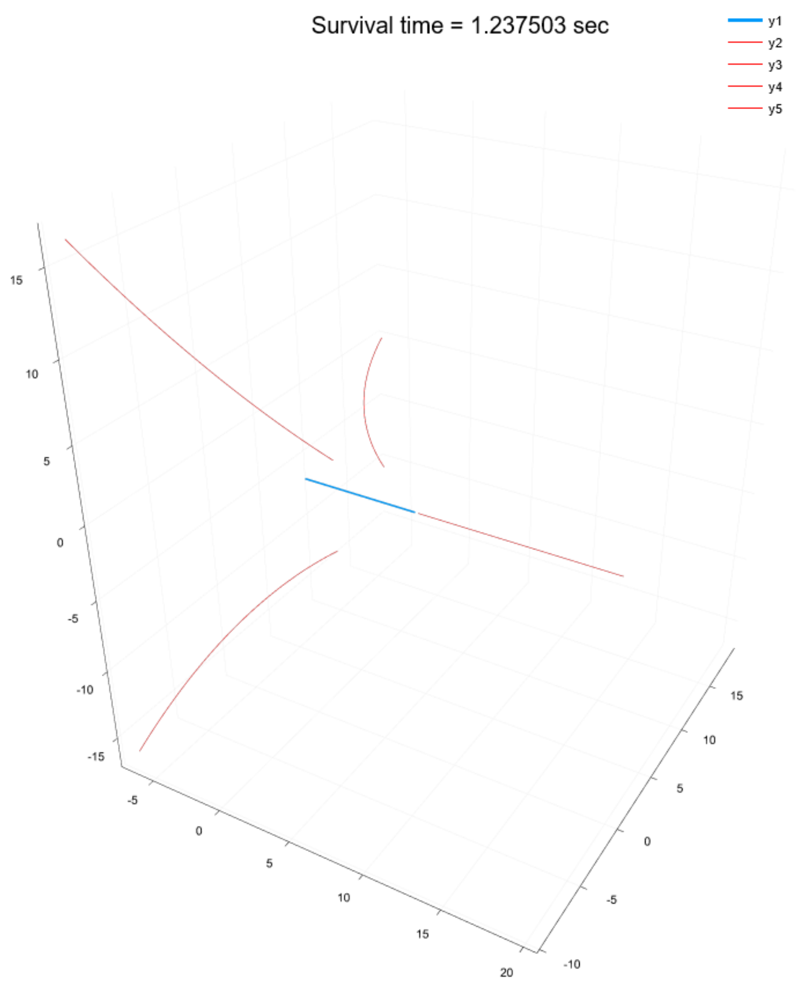

Figure 2: Survival time when running directly at the slow raptor

Parameters when the humain is headed towards the wounded raptor

```
theta = 0.0;
phi = pi / 2.0;
show_raptors(theta, phi);
```

When running directly at the slow raptor, the survival time is about **1.24 seconds**. (See figure 2)

Question 2:

Utilize a grid-search strategy to determine the best angle for the human to run to maximize the survival time. Show the angle.

I changed the simulation function to only output the survival time.

```
function simulate_raptors_no_histogram(theta, phi)
    # initial positions
    h = [0.0, 0.0, 0.0];
    r0 = [1.0, 0.0, 0.0]*raptor_distance;
    r1 = [-1.0/3.0,  sqrt(8.0)/3.0,            0.0]*raptor_distance;
    r2 = [-1.0/3.0, -sqrt(2.0)/3.0,  sqrt(2.0/3.0)]*raptor_distance;
    r3 = [-1.0/3.0, -sqrt(2.0)/3.0, -sqrt(2.0/3.0)]*raptor_distance;

    # how much time elapsed
    dt = tmax/nsteps;
    t = 0.0;

    for i=1:nsteps
        dh, dr0, dr1, dr2, dr3 = compute_derivatives(
                                     theta, phi, h, r0, r1, r2, r3);
        h += dh*dt;
        r0 += dr0*dt;
        r1 += dr1*dt;
        r2 += dr2*dt;
        r3 += dr3*dt;
        t += dt;

        if norm(r0-h) <= raptor_min_distance ||
            norm(r1-h) <= raptor_min_distance ||
            norm(r2-h) <= raptor_min_distance ||
            norm(r3-h) <= raptor_min_distance
            # Return the elapsed time
            return t;

            break
        end
    end

    return t;
end
```

Following is the grid search code:

```
# Grid search for the best parameters
# Parameters are near one solution (to increase the precision)
theta = 0.347:0.0001:0.351; # 0:0.1:2.0*pi (in the general case)
phi =   1.033:0.0001:1.037; # 0:0.05:pi    (in the general case)
time = zeros(length(theta), length(phi));

for i = 1:length(theta)
    for j = 1:length(phi)
        time[i, j] = simulate_raptors_no_histogram(theta[i], phi[j]);
    end
    println(i, "/" , length(theta));
end

(max_time, index) = findmax(time);
println("Best Theta ", theta[index[1]], " rad");
println("Best Phi ", phi[index[2]], " rad");
println("Survival time ", max_time, " sec");
# surface(theta, phi, time);

best_theta = theta[index[1]];
best_phi = phi[index[2]];
show_raptors(best_theta, best_phi);
```

Figure 3 is a plot of the surface of the function. We can see that there are 3 global maxima. We can choose one of them as solution to our problem. One possible solution to our problem is shown in figure 4. We can notice that 3 raptors are eating the human almost at the same time. The parameters are: $\Theta = 0.3493 \quad \text{and} \quad \phi = 1.0356$. The maximum survival time is 1.5599 seconds. I ran the simulation with 100,000 steps and a maximum time of 3.0 seconds.

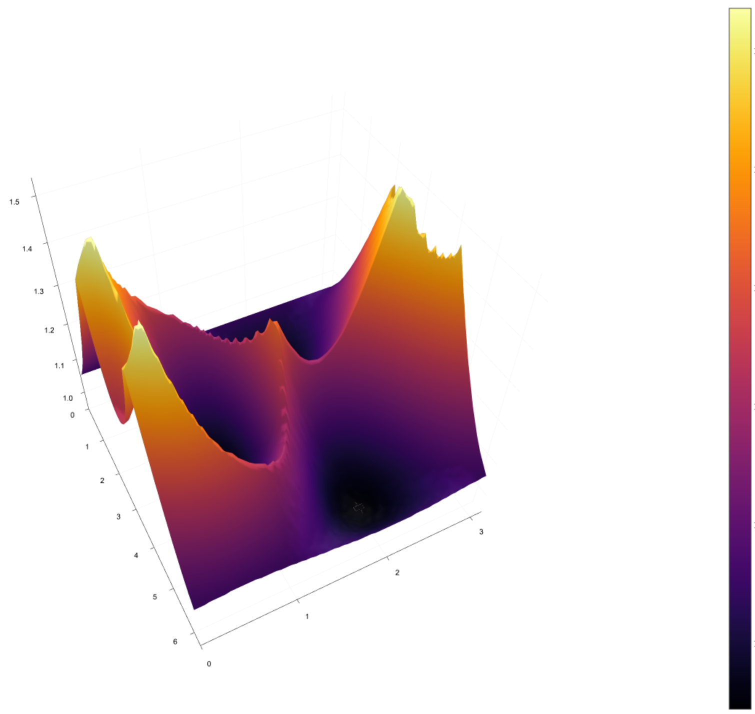

Figure 3: Surface of the function

Question 3:

Discuss the major challenge for solving this problem with the current strategy in four dimensions.

The major challenge with the grid search is that the complexity is in $O(n^k)$, with n being the grid size and k the dimension of the problem. Thus, every time we add a dimension, the problem is n times harder. For 2, 3 or 4 dimensions it is still tractable, however with more dimensions, for instance $k = 20$, it is too hard to optimize the function because the search space is too vast.

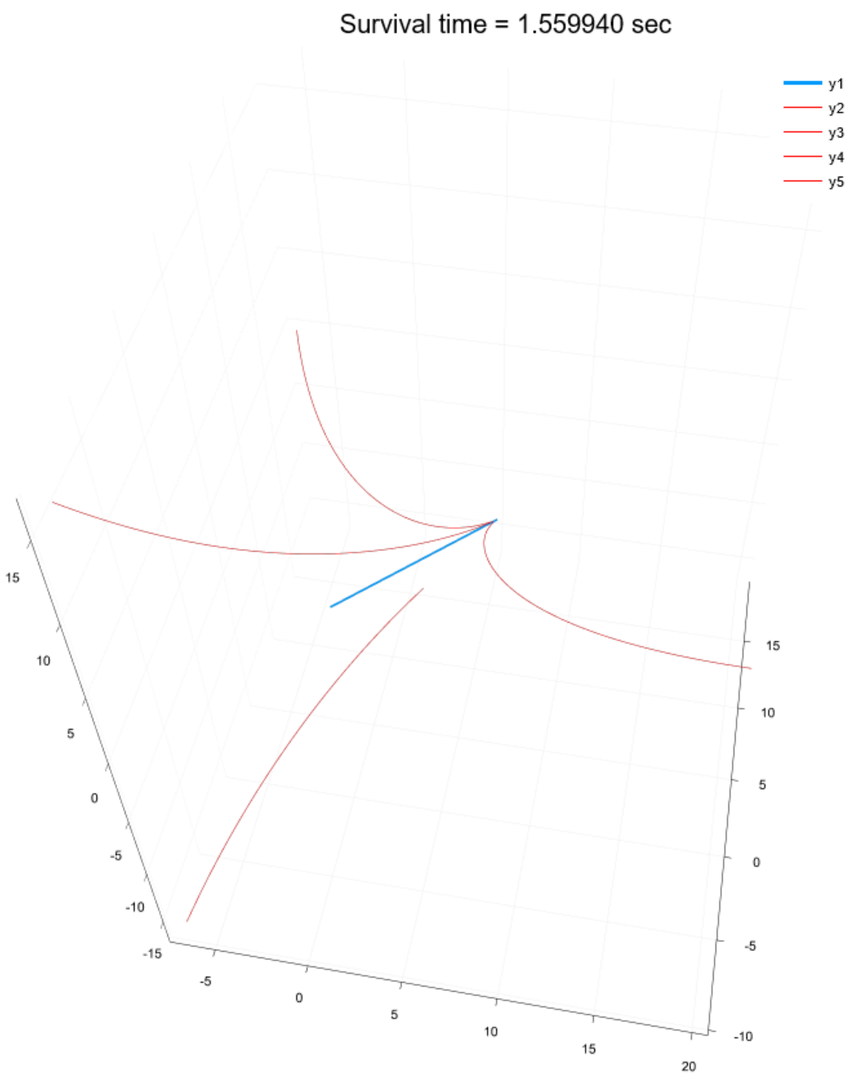

Figure 4: One of the 3 best configurations to maximize survival time.