Navigation Problem of Speed research for parameters dependence

Daniel Paliura

3/11/2021

Introduction

This document was created as additional to 'Control Theory' subject first laboratory work 'Navigation Problem of Speed'. This document contains research for a dependence of input parameters on the problem solution. Also, the cause of the influence will be explained if it's possible.

We will think that boat can be both rowboat or motorboat. Also, the stream will be recognized as unidirectional at each point, because it is such in rivers and can be recognized so in oceans or seas. So we will think that stream doesn't change it's direction on boat's way from origin to target. There wasn't said anything about boat turn speed so boat turn speed is neglected.

Formulation of the Navigation Problem of Speed

It's the boat on the water surface that has to catch the immovable target placed on the same water surface. Water has stream moving with some speed.
Formalized problem:
to find the boat control that moves the boat to reach the target. The boat is defined as a material point that moves with constant speed v relative to water surface. Initially boat is in origin - point $(0, 0)$ - of plane xOy, where target does not moves, which means that water surface moves relatively to plane. Stream vector is parallel to ordinate axis (Ox) in each point of plane and perpendicular to abscissa axis (Oy) if it isn't zero vector. Absolute value of stream vector depends on y coordinate and is described by expression:

$$s(y) = s_0 \cdot f(y),$$

where
s_0 - stream speed multiplier or initial stream speed,
$f(y)$ - some defined function of stream speed dependence. Target is in point distanced on l from origin and angle between x axis and vector from origin to target is ϕ, so the target is in point $x^* = l \cdot cos(\phi), y^* = l \cdot sin(\phi)$.

Solution method provided

To solve this problem was spelled aiming method, which chooses best control vector in current point considering stream speed in point. In this method, the speed of stream is considered as constant for a small time interval $\tau = \frac{l}{vN}$, where N is some large number. I set $N = 1e3$.
According to analytic calculations (Lagrange polynomial optimization), method uses following recurrent formulas:

$$\lambda(t_k) = v\tau\sqrt{(x^* - x_k - s(y_k)\tau)^2 + (y^* - y_k)^2 - (v\tau)^2},$$

where
$t_k = k\tau$ - the time moment on k iteration,
x_k, y_k - coordinates of boat position at moment t_k,

$k = 0 \cdots K, N \leq K \in \mathbb{N}$
and $t_0 = 0, x_0 = 0, y_0 = 0$.

$$u_x(t_k) = v\tau \frac{x^* - x_k - \tau s(y_k)}{\lambda(t_k) - (v\tau)^2},$$

$$u_y(t_k) = v\tau \frac{y^* - y_k}{\lambda(t_k) - (v\tau)^2}$$

u_x, u_y should be recognized as boat control in moment t_k. Hence next position defined by formulas:

$$x_{k+1} = x_k + \tau(s(y_k) + u_x(t_k)v),$$

$$y_{k+1} = y_k + u_y(t_k)v\tau$$

To exclude infinite looping when the target cannot be reached we shall set some maximum number of iterations as $K > N$ so $k \leq K$, and if $x_k = x^*, y_k = y^*$ (target reached) then we have to define $K = k$.

The full time of reaching the target is $T^* = \tau K$ if target reached, and if not then $T^* = +\infty$.

Parameters effect research

Boat speed

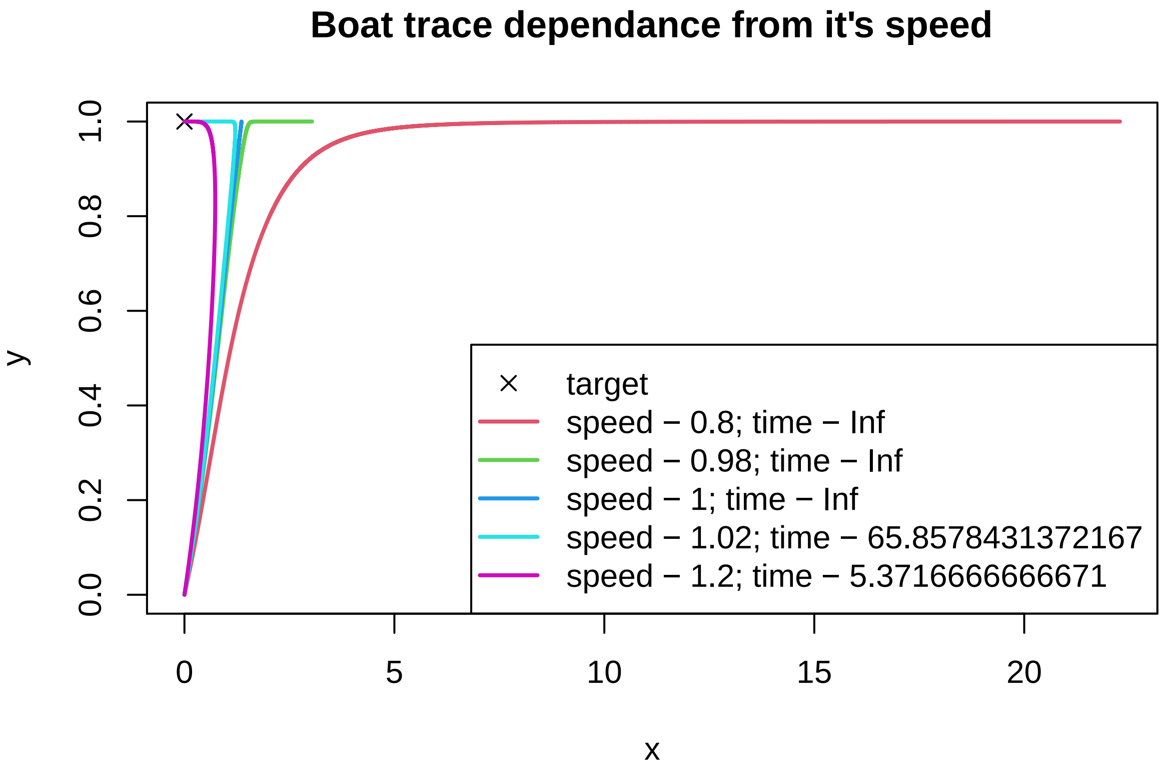
Analyzing the problem formulation and given method we can get the conclusion:
Stream moves the boat only by x axis and hence must condition to reach the target is $v > s(y^*)$, $s(y)$ is continuous on interval $y \in [0, y^*]$ and $\forall y \in [0, y^*), |s(y)| < \infty$. This conclusion says that if stream speed is finite on the way then the boat won't be blown into an infinitely distant point from which it can't return to target. It can be moved any far away but finitely. Also, conditions $s(y)$ continuous and $v > s(y^*)$ are must for the boat to be able to reach such a radius of y^* where boat speed will exceed stream speed in each point on way to the target, so the boat will be approaching to target on its way. BUT it must be said that **specified CONDITIONS are must but NOT SUFFICIENT**. It's so due to conditions don't guarantee that boat will converge that radius of y^* where $v > s(y)$. If the boat wouldn't aiming to target but would move only by vertical first to reach such radius, then conditions would be sufficient. But for our case, it can be the situation when $\frac{du_y}{dt} = 0$ while $t \to +\infty$, and hence $y = \int_0^\infty u_y(t)dt < y^*$, where y is boat coordinate. That's why named conditions are not sufficient.

According to formulated sentences, we can modify that method the next way: the boat must not decrease its vertical speed less than some predefined non-zero number independently on how it close to y^* just until it reach y^*.

Now let's build some models for this method to vary speed parameter to see how boat speed affects on ability to reach target. We shall set parameters in such a way that stream speed will be known in line $y = y^*$ to have the ability to check our hypothesis. So we will fix parameters except v as next:
$s_0 = 1$,
$l = 1$,
$\phi = \pi/2$,
$f(y) = 2 - y$.
So, according to formulated conditions, we have: target in point $x^* = 0, y^* = 1$ and hence speed of stream decreasing from 2 to 1 while approaching. So $s(y^*) = 1$ and it is expected that boat will reach a target with a speed greater than 1.

We could take $f(y) = sin(\frac{y\pi}{2})$ but in such a case it would be the situation when stream speed wouldn't move the boat from target because its speed would be less than the boat's on the whole way. And the boat would go to target directly.

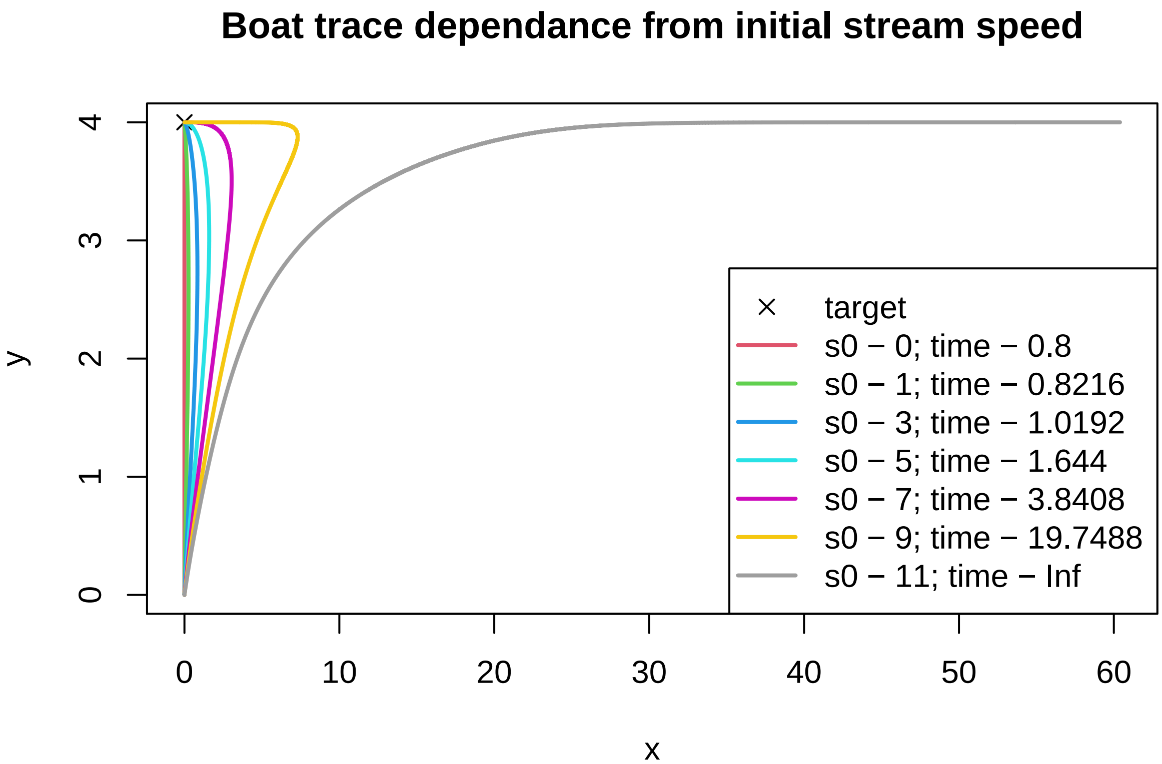
To see the influence of boat speed we will build models for a set of speeds: $V = \{0.8; 0.98; 1; 1.02; 1.2\}$ and see which ones will let the boat reach target.
Traces are shown on the plot:



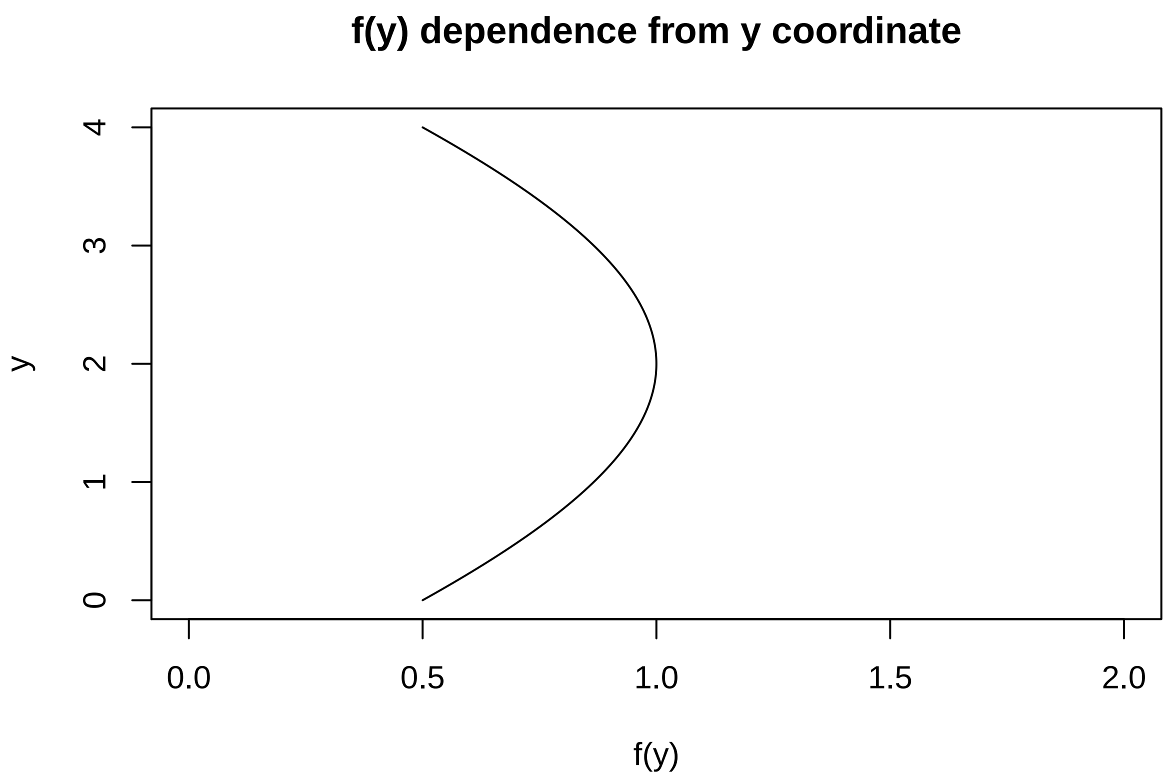
As we can see, there are 3 cases when the boat can't reach the target, and all they are when $v \leq s(y^*) = 1$ as it was expected. In such case, it is proofed to add constraint to stop algorithm when $y = y^*, x \neq x^*$ and $v \leq s(y^*)$.

Stream initial speed

It might be not hard to conclude that the initial stream speed affects the time spent to reach the target and probably nonlinear. Also, it will affect boat trace.
We will try to vary s_0 for next fixed values of other parameters:
$v = 5$,
$l = 4$,
$\phi = \pi/2$,
$f(y) = sin(\frac{y\pi}{6} + \frac{\pi}{6})$.
Chosen expression for $f(y)$ provides values of it between 0.5 and 1 with values 0.5 when $y \in \{0; y^*\}$. According to previous research we can variate initial stream speed as next: $s_0 \in \{0; 1; 3; 5; 7; 9; 11\}$.
Traces are shown on the plot:



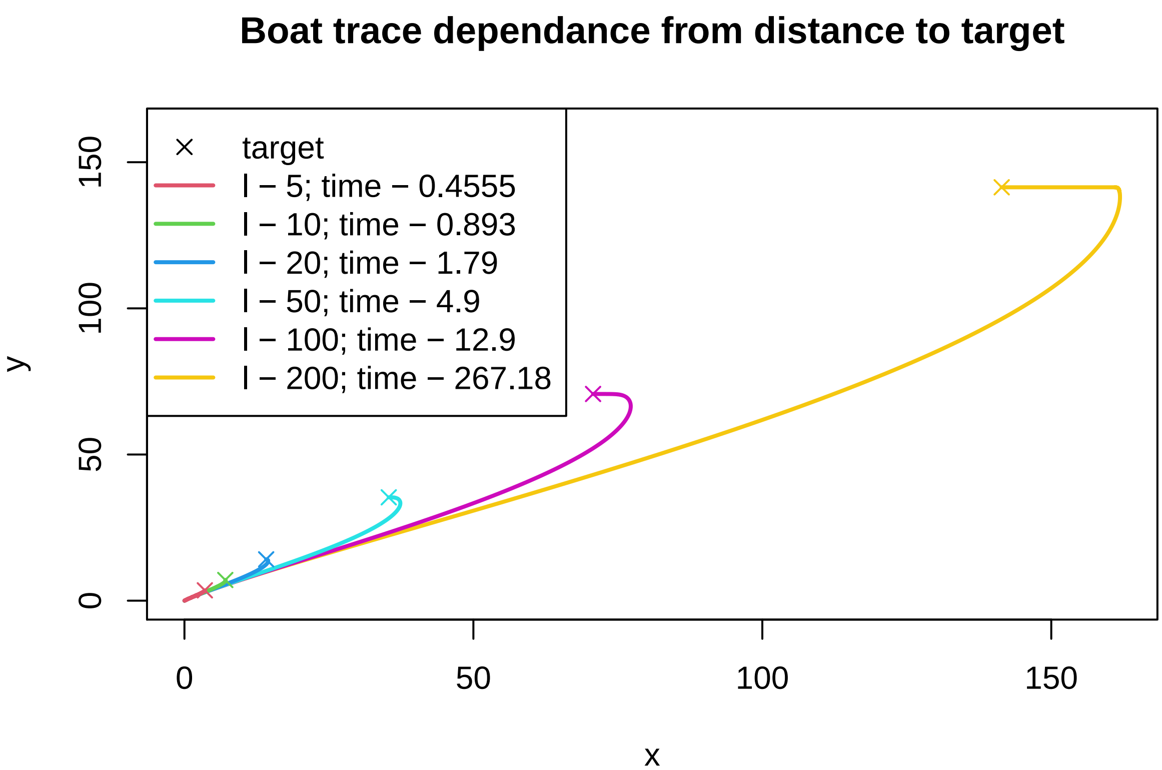
This plot may seem not very interesting, but there is one interesting moment which could be noticed earlier. But first about dependence. As we can see the time of target reach depends nonlinear on stream speed. When initial stream speed approaches such that whole stream speed near to the speed of the boat, the time increases much faster (the difference between the time for $s_0 = 7$ and $s_0 = 9$ is much more than between $s_0 = 3$ and $s_0 = 5$). This parameter and the previous one variations says that boat speed must be much greater than stream speed to reach the target fast. And also, in such cases, stream speed affects much less on time if the aiming method is used. But more interesting are curves. We can see that their forms (especially when $s_0 = 9$) have a bend near the line $y = y^*$. It's interesting because we may conclude that boat slides vertically and moves faster to the target by y coordinate at the first time, but then it slows vertical move loosing speed for fighting stream speed. But we should be careful making such conclusions. Let's see on the plot of $f(y)$ speed dependence:



As we can see, it would be better to overcome the middle part of the way (about between 1 and 3) straight vertically to pass rapid stream as fastest possible and hence to has least loss of distance on fighting stream.

Distance to target

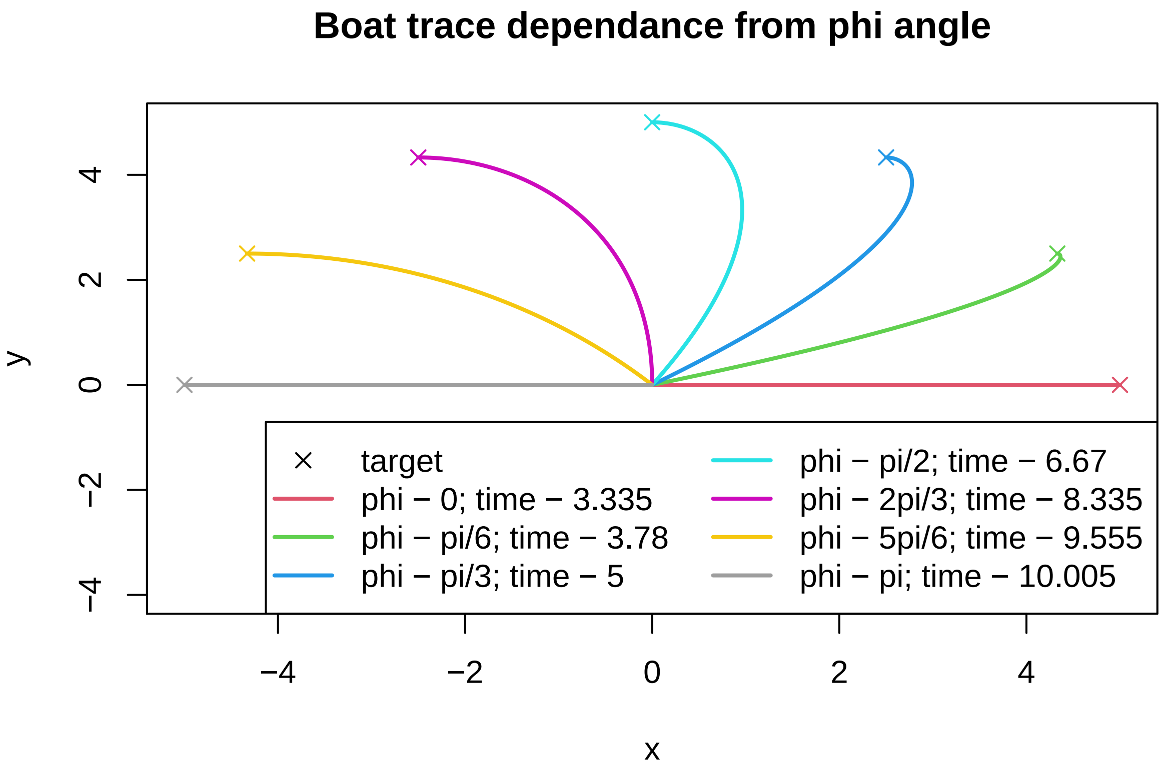
This parameter may influence on time and nonlinear but about it. To find something interesting we will fix parameters next way: $v = 10$ $s_0 = 2$ $\phi = \pi/4$ $f(y) = log(y+1)$ And variate $l \in L = \{5; 10; 20; 50; 100; 200\}$. As we took $\phi = \pi/4$, we expect that stream will help the boat to reach target faster on low distances. Traces are shown on the plot:



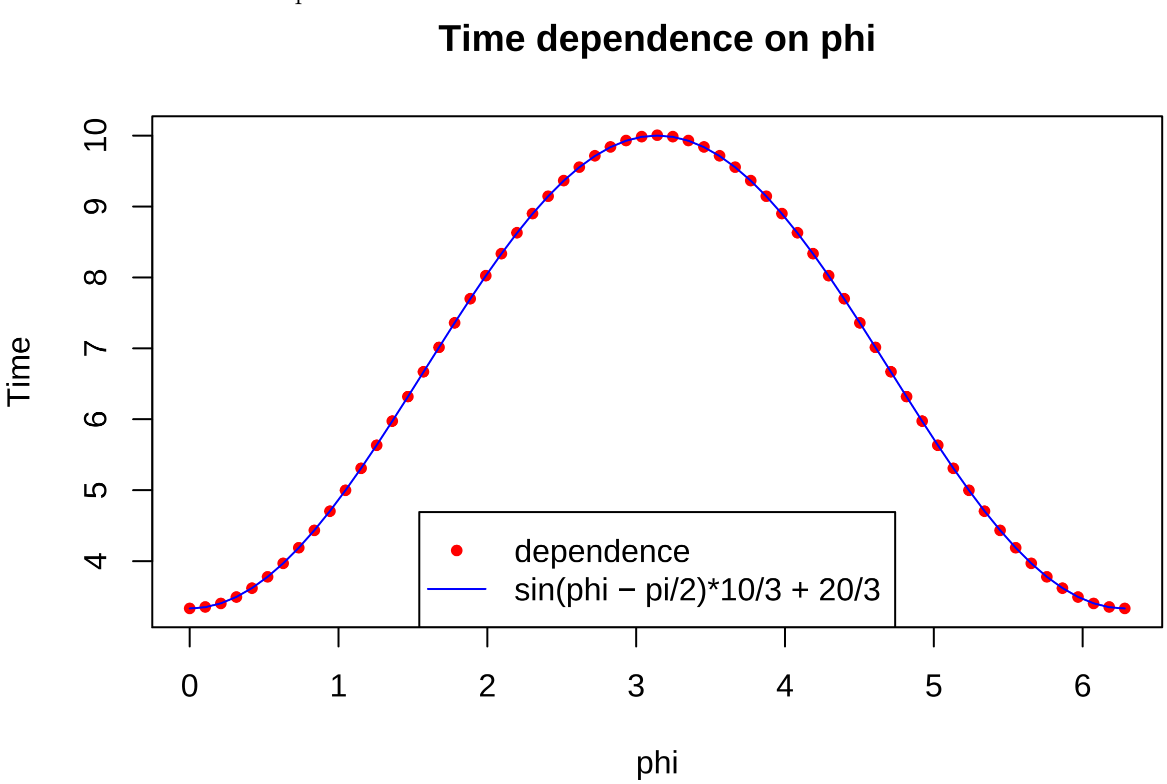
Without stream, the time would be $T = \{0.5; 1; 2; 5; 10; 20\}$, but with the stream, we have times less than without stream for $l \le 50$ and with $l = 10, l = 20$ there is the best profit of stream, which notably from traces (green and dark blue). The boat doesn't make a hook due to the stream. In the first case (red line), there is almost no profit of stream because its speed is very small on the whole way. When $l = 50$ the boat already made a hook but still won more time on moving by the stream than lost on a hook. And for more distant points $l = 100, l = 200$, the boat already lost more time on a hook than it got by moving along the stream. Also for $l = 200$ stream speed approaches very close to boat speed: $s(y^*) = 2 * log(\frac{200}{\sqrt{2}} + 1) \approx 9.91758$, So the boat lost a lot of time in this case due to reasons clarified before.

Angle between stream vector and vector from origin to target

Parameter ϕ may affect directly on time and trace but, as seen earlier, great influence has function $f(y)$ because it defines how much stream affects on the boat. We will take each $\pi/6$ angle as ϕ in interval $[0, \pi]$.
Other parameters will be next: $v = 1$ $s_0 = 0.5$ $l = 5$ $f(y) = 1$
Traces are shown on the plot:



It was expected that the time depends on the angle, it's because the angle defines how much the boat will move against the stream. It is interesting which dependence between angle and time. To view it I will make more models for angles $[0, 2\pi]$. It's OK that the target will be lower than the x axis in some cases, the method allows it. Other parameters will be left the same.



Dependence was such interesting and simple that I tried and built some sinus line which is similar to got dependence. According to this plot, we can conclude that angle almost doesn't affect time when angle is near to 0 or πk but affects almost linear when is around $\pi k/4$ for $k \in \mathbb{Z}$. But as we viewed dependence when stream speed is constant the conclusion is not general.

Stream speed function

This parameter was varied through researching other parameters and we found out that it can even make the target unreachable. But we can prove the next sentence: if $\exists \hat{y} \in [0, y^*] : |s(\hat{y})| < v$ and $\forall y \in [0, y^*] : |s(y)| < \infty$, where $s : \mathbb{R} \to \mathbb{R}$ is continuous, then the target can be reached. But it might be impossible with the aiming method. Another algorithm should be used, it also can be optimal.

While researching this problem I got an idea about an algorithm that could be used to provide reach control if mentioned conditions are true. I will try to prove this in the next part.

Reach condition proof

Conditions say that:
1) $\exists \hat{y} \in [0, y^*] : |s(\hat{y})| < v$
2) function $s : \mathbb{R} \to \mathbb{R}$ is continuous
3) $\forall y \in [0, y^*] : |s(y)| < \infty$

Every time I say "boat moves along the y axis" I mean its control along the y axis, so the boat uses all speed to change only the y coordinate, and the x coordinate is changed only by a stream.

According to conditions 2) and 3), the function $s(y)$ is Riemann integrable on interval $[0, y^*]$ due to the theorem (about continuous function integrability). Hence, according to Riemann integral properties, we have:

$$\exists I_1 \equiv \int_0^{\hat{y}} t(y)s(y)dy$$

$$\exists I_2 \equiv \int_{\hat{y}}^{y^*} t(y)s(y)dy$$

and hence

$$I = I_1 + I_2 = \int_0^{y^*} t(y)s(y)dy$$

This integral equals to whole boat shift through its way, but it depends on y - not directly on time t. If the boat moves vertically from origin to level $\hat{y}$ and after - from level $\hat{y}$ to level y^*, then the whole shift by x axis will be I. Wherein time intervals for this move will be

$$\Delta t = \frac{\Delta y}{v} \Leftarrow (v = \frac{\Delta y}{\Delta t} = \frac{y_2 - y_1}{t_2 - t_1} = const)$$

so the whole time for the straight vertical move will be:

$$T_{vrt} = \frac{\hat{y} - 0}{v} + \frac{y^* - \hat{y}}{v} = \frac{y^*}{v}.$$

And while moving vertically, we have $y(t) = vt$, so $dy = vdt$. So we have:

$$I = v \int_0^{\frac{y^*}{v}} ts(vt)dt$$

But there is some issue: the boat won't move its whole way by y coordinate. The idea is to move the boat to such a y coordinate where absolute stream speed will be less than boat speed, so the boat could move by the x axis in any direction. Got formula for I will just help us to know how much and whereto move the boat by the x axis to piece out the shift of stream got while vertical move. And got I value is directly those value of x shift which the boat have to piece out. We have to satisfy condition: $I + \Delta x = x^*$ or simpler $\Delta x = x^* - I$, where Δx is distance which the boat have to overcome along the x axis to shoot into target at the end of the move by the y axis. let's define direction as variable $d \in \{-1; 1\}$, and we have:

$$\Delta x = (s(\hat{y}) + d \cdot v)T_{hor} = x^* - I$$

According to condition 1): if $d = -1$ then $\Delta x \leq 0$, else if $d = 1$ then $\Delta x \geq 0$ and $\Delta x = 0$ only if $T_{hor} = 0$. According to all mentioned about direction and Δx we have next rule:

$$d = B^+_-(\Delta x) = B^+_-(x^* - I)$$

where B^+_- - bipolar step function which frequently used in Machine Learning. B^+_- is my designation which, I hope, is intuitive. $B^+_-(x)$ is -1 if $x < 0$ and is $+1$ if $x \geq 0$. The reason why I got bipolar step function instead of signum function is that bipolar step function doesn't return zero. Zero could cause division by zero in the future. And now we can get whole time of horizontal boat move:

$$T_{hor} = \frac{x^* - I}{s(\hat{y}) + vd}$$

It could be the situation when, for example, $s(y) \equiv 0, x^* = 0$ so in such case, stream shift would be zero, stream speed would be zero and hence we would have $\frac{0}{0}$ if we taken signum function instead of bipolar step function. And as $v \neq s(\hat{y})$ so in such case we would have zero division by some non-zero value. And it would be nice, because boat wouldn't move horizontally as it wouldn't need to.

Also the whole boat move time is:

$$T = T_{vrt} + T_{hor} = \frac{y^*}{v} + \frac{x^* - I}{s(\hat{y}) + vB^+_-(x^* - I)}$$

Boat trace will be next:
at first, the boat control is directly vertical until reach line $y = \hat{y}$ for a time equals to $\frac{\hat{y}}{v}$ then the boat turns up at the point $(v \int_0^{\frac{\hat{y}}{v}} t \cdot s(vt)dt; \hat{y})$. After that, the boat moves along the x axis for a time T_{hor} and appears at the point. $(x^* - v \int_{\frac{\hat{y}}{v}}^{\frac{y^*}{v}} t \cdot s(vt)dt; \hat{y})$. Finally, the boat controls again along the y axis for a time $\frac{y^* - \hat{y}}{v}$ and shots directly to the point (x^*, y^*), **which was to be proven**.

Algorithm proposition

As we found out next expression for the whole time:

$$T = \frac{y^*}{v} + \frac{x^* - I}{s(\hat{y}) + vd}$$

we could see that it depends on a single variable $\hat{y}$ because all other values are constant in it. Due to $s(y)$ continuity, many points can be got as $\hat{y}$. So the idea is next: at first, we should find value I using numeric integration. Then we must find point $\hat{y}$ as the optimal point. And it can be maximum or minimum. To recognize which optimum to find we should know the value for $d = B^+_-(x^* - I)$. If $d = -1$ then we have to find minimum to provide a positive value of T_{hor} and else if $d = +1$ then we have to find maximum. Also, such optimums will provide the greatest absolute value into divisor which will minimize the whole fraction. So we have to find value $\hat{y}$ as mentioned optimum. After that, find the time for the horizontal boat T_{hor}. To build the boat's trace we have to make a model with vertical control to level $y = \hat{y}$, then with horizontal control in the direction defined as d and again with vertical control until reaching level $y = y^*$. It might be a problem due to numeric methods used: the boat can end not in target point. In such a case, if the stream isn't faster than the boat at the target point, then it can be pieced out when boat reaches level $y = y^*$.

Proposed algorithm step by step

Set big number N. Split interval $[0, y^*]$ into N intervals of same length. Having $y_i = i \cdot \frac{y^*}{N} = i \cdot \Delta y, i = 0 \cdots N$ define $t_i = i \cdot \frac{\Delta y}{v} = i \cdot \Delta t$.
Set $x_0 = 0, y_0 = 0$.
Loop over $j = 0 \cdots N - 1$:
$x_{j+1} = x_j + \Delta t \cdot s(t_j)$;
$y_{j+1} = y_j + \Delta y$.

After loop ends set $I = x_N$ and $\Delta x = x^* - I$.
If $\Delta x = 0$ then END with got trace through points $(x_i, y_i), i = 0 \cdots N$;
If $\Delta x < 0$ then find $\hat{y} = argmin_{y \in [0, y^*]} s(y)$ and set $d = -1$.
If $\Delta x > 0$ then find $\hat{y} = argmax_{y \in [0, y^*]} s(y)$ and set $d = +1$;

Find such number $k, 1 \leq k \leq N$ that $y_{k-1} \leq \hat{y} \leq y_k$, in simple words, find the splitting interval which contains $\hat{y}$.

Then find value $\hat{x} = x_{k-1} + (x_k - x_{k-1}) \frac{\hat{y} - y_{k-1}}{y_k - y_{k-1}}$.

Loop over $j = N \cdots k$ (descending order):
set indexes shift (thus looping on descending order) and boat shift by x axis
$x_{j+2} = x_j + \Delta x$,
$y_{j+2} = y_j$

And after loop there is last hatch.
Set $x_k = \hat{x}, y_k = \hat{y}$ and
$x_{k+1} = \hat{x} + \Delta x, y_{k+1} = \hat{y}$.
Done.

Time of boat move should be calculated as such:

$$T = \frac{y^*}{v} + \frac{\Delta x}{s(\hat{y}) + vd}$$

That's all.

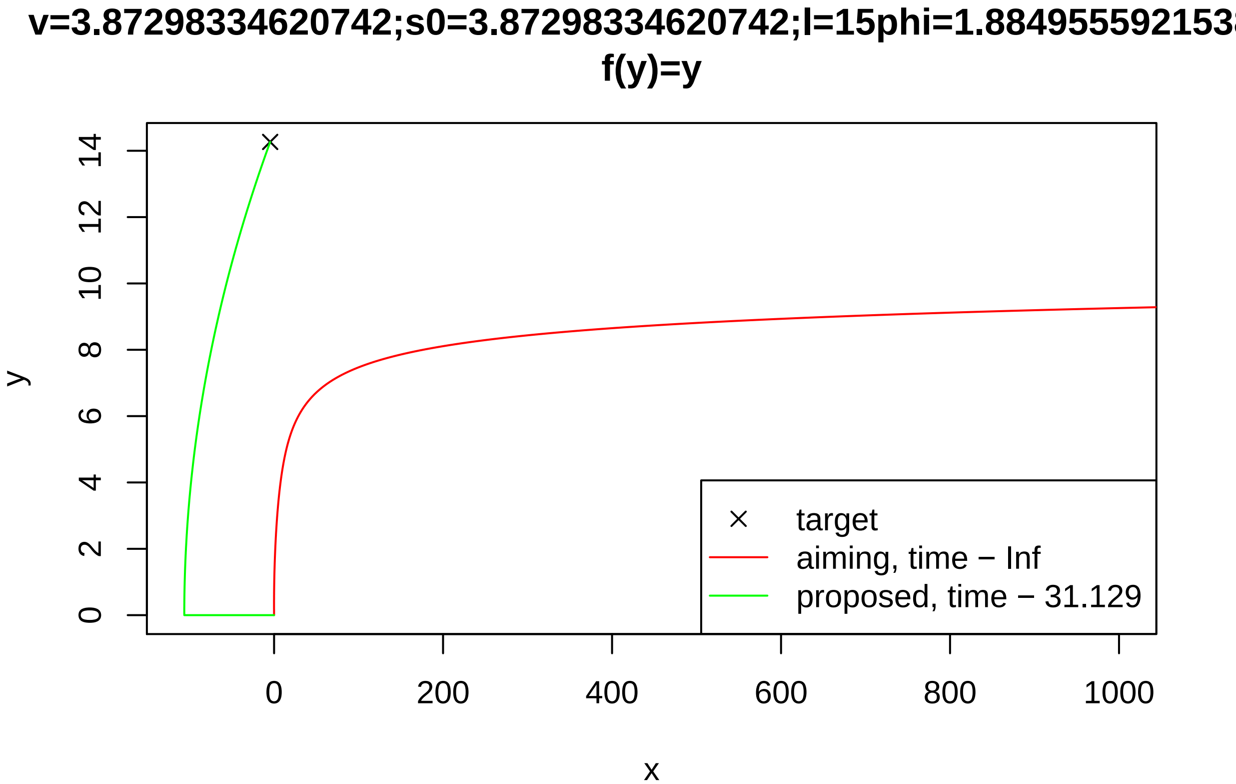
Proposed algorithm test

The algorithm written on R language is next:

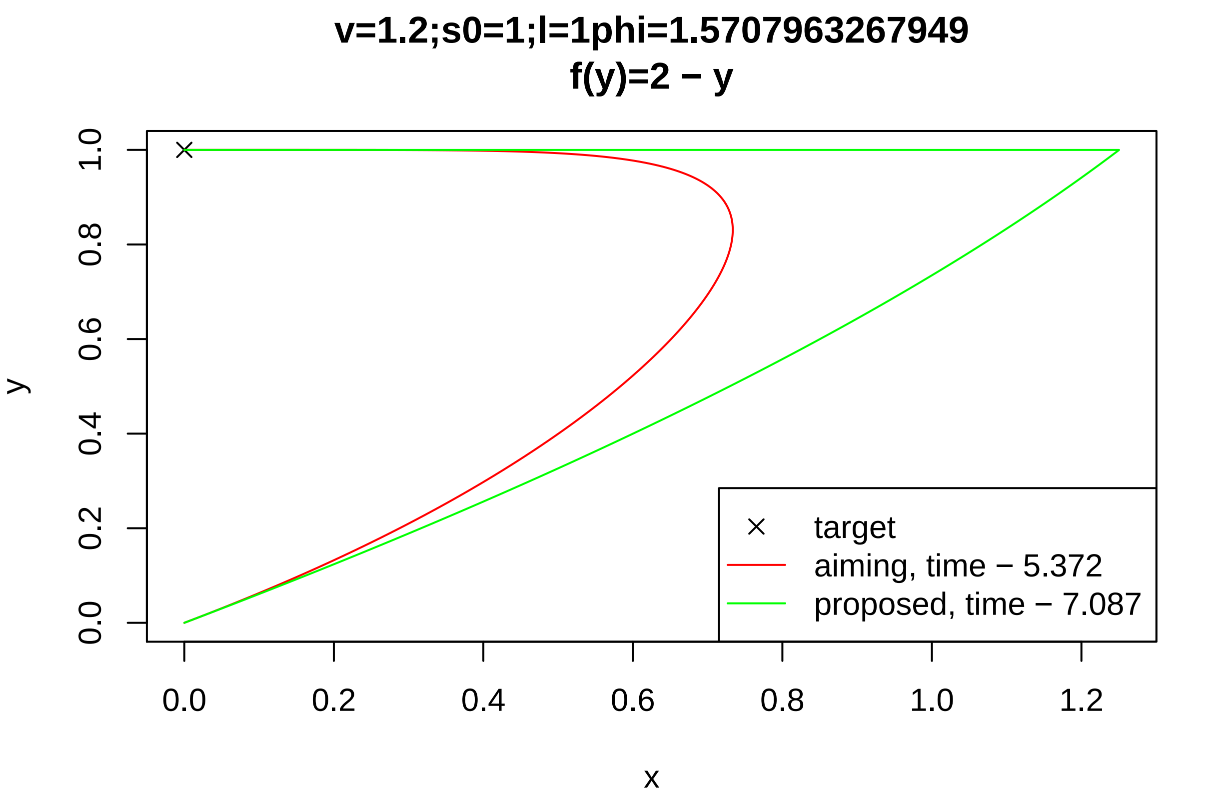
```
build.my.model <- function(input, N = 1e3){
    v <- input$v
    s0 <- input$s0
    l <- input$l
    phi <- input$phi
    f <- input$f

    target <- l * c(cos(phi), sin(phi))
    s <- function(y) s0 * f(y)

    y <- 0:N * target[2] / N
    dy <- target[2] / N
    t <- y/v
    dt <- dy/v

    x <- 0
```

```
    for (j in 1:N){
        x <- c(x, x[j] + dt*s(y[j]))
    }

    I <- x[N+1]
    Dx <- target[1] - I

    T_ver <- t[N+1]

    if (Dx == 0) return(list(points = cbind(x, y),
                             time = T_ver,
                             target = target))
    if (Dx < 0){
        d <- -1
        y_hat <- optimize(f, c(0, target[2]))$minimum
    }
    if (Dx > 0){
        d <- +1
        y_hat <- optimize(f, c(0, target[2]), maximum = TRUE)$maximum
    }

    k <- which((y[1:N] <= y_hat) & (y_hat <= y[(1:N)+1]))

    x_hat <- x[k] + (x[k+1]-x[k])*(y_hat-y[k])/(y[k+1]-y[k])

    T_hor <- Dx / (s(y_hat) + v*d)

    return(list(points=rbind(cbind(x[1:k], y[1:k]),
                             c(x_hat, y_hat),
                             c(x_hat+Dx, y_hat),
                             cbind(x[(k:N)+1]+Dx, y[(k:N)+1])),
                time = T_ver + T_hor,
                target = target))
}
```

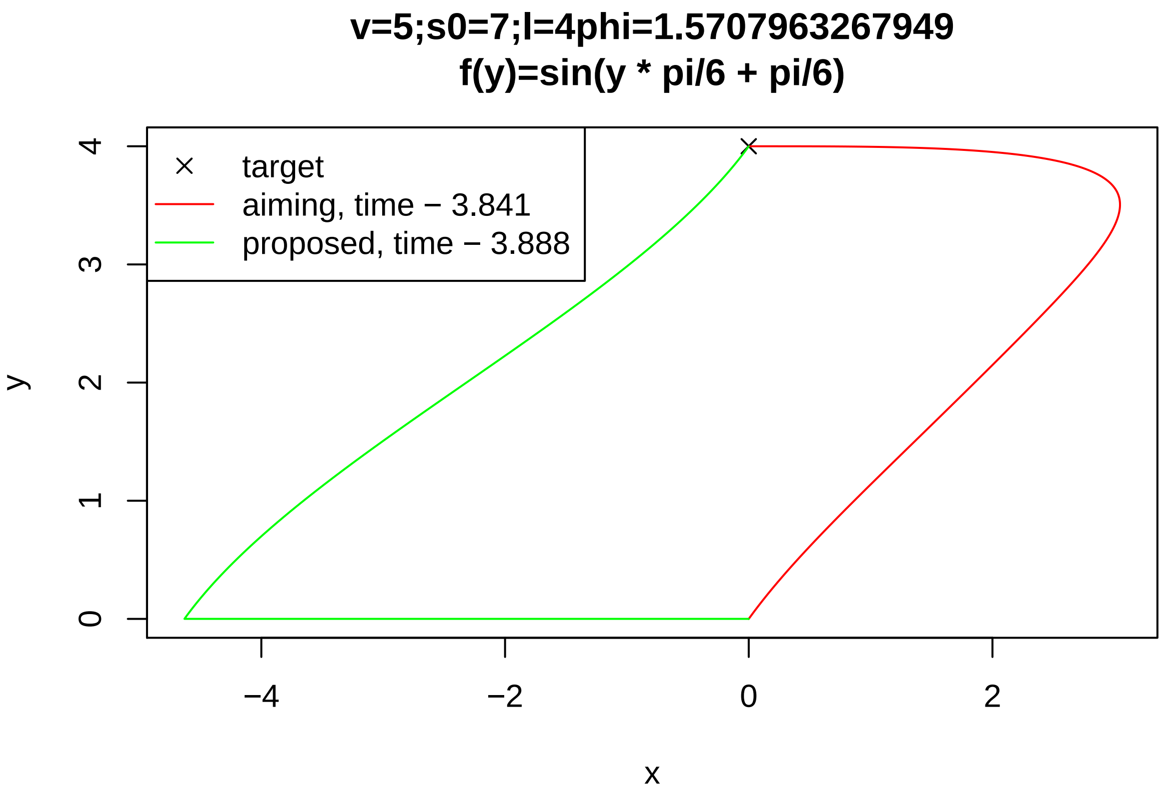
Test it with parameters for my laboratory variant (aiming method wasn't provide target catch).



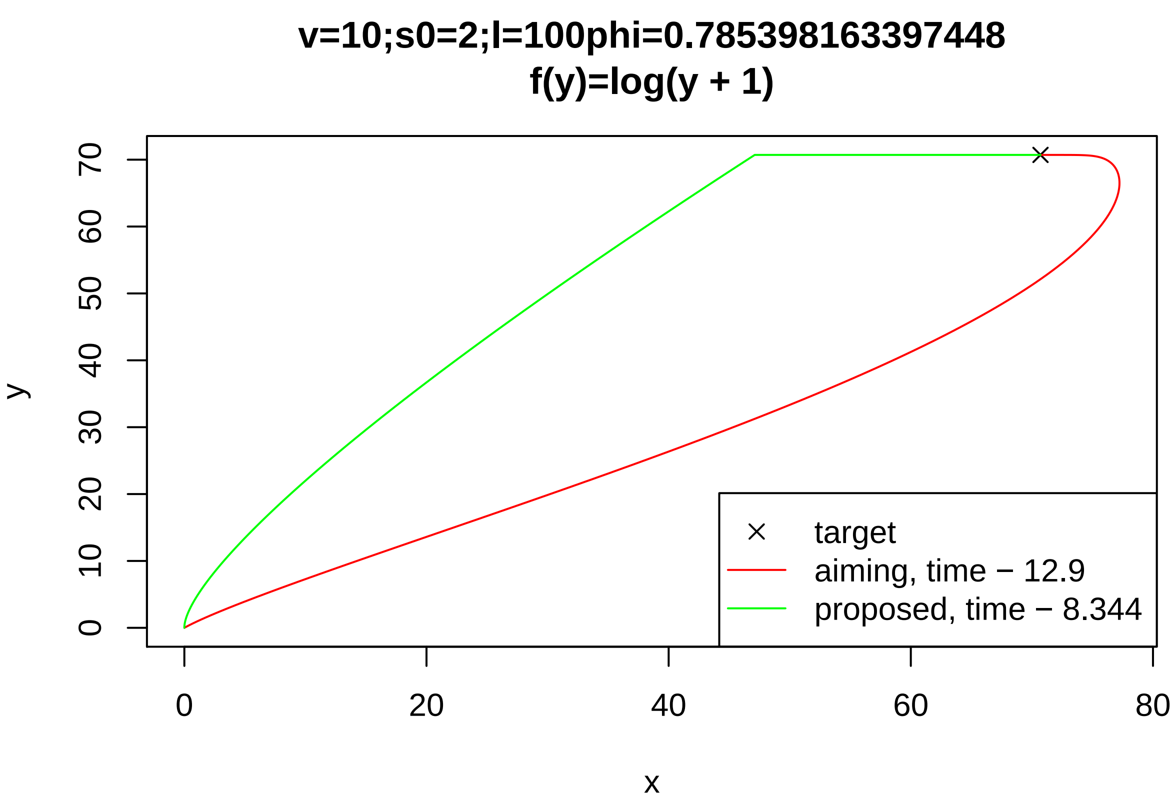

```
## Proposed algorithm gave better time than aiming method
```



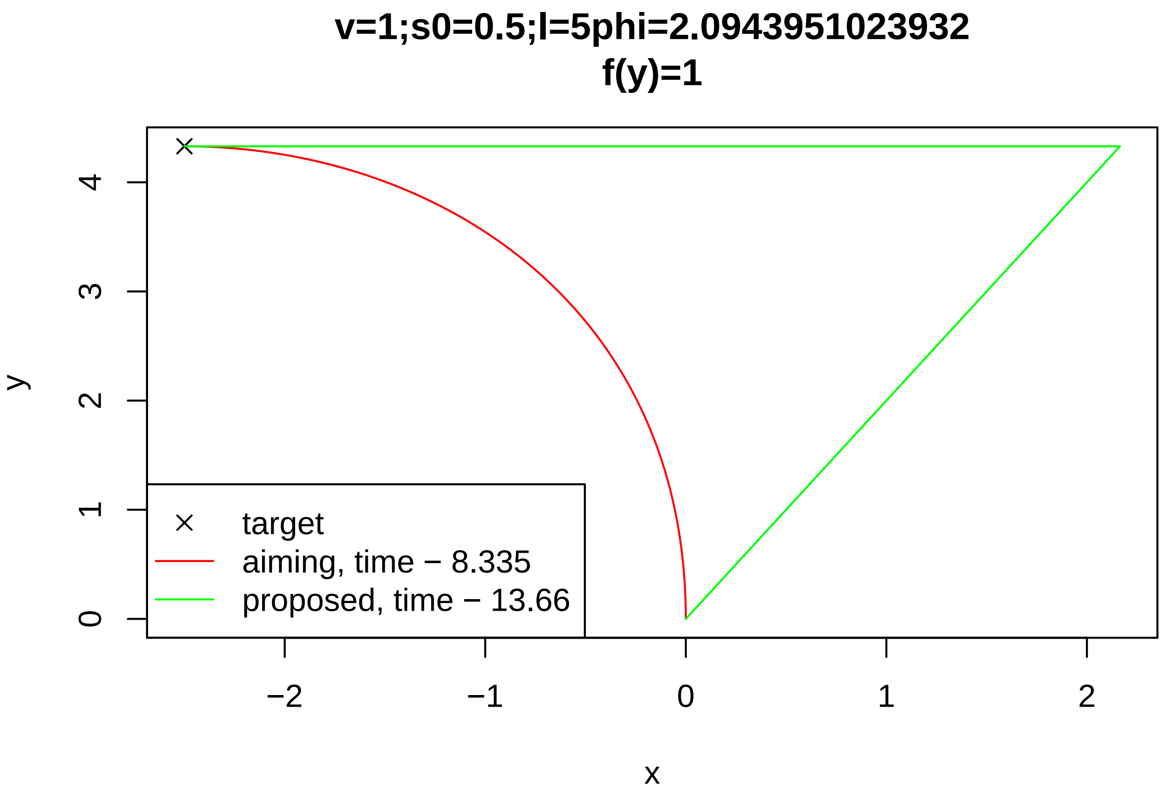

```
## Aiming method gave better time than proposed algorithm
```



Aiming method gave better time than proposed algorithm



Proposed algorithm gave better time than aiming method




```
## Aiming method gave better time than proposed algorithm
```

Proposed algorithm comparison with aiming method

We see that the proposed algorithm works nice but does not always provide better time than aiming does. It was expected. Analyzing traces, we can notice that the algorithm makes a hook when stream speed decreasing near the target. The problem of aiming was when stream speed greater near target than near origin. It leads to thoughts about available improvements. For example, we can choose an algorithm based on knowledge of how the stream behaves near the origin and near the target. But if think about the problem as real, we can say that stream of the river is faster near the river's middle and slower near shorelines. It's might be such due to depth and also because fixed shore slows moving water, while in the middle of river water has a laminar flow and is not slowed. Depth affects such as shore because the bottom is far from the surface in a middle of a river, and the bottom is close to the water surface near shore.

Also, the problem of the proposed algorithm is next: it neglects the boat's turn speed, so it does unreal things for real experience. This problem can be fixed by the insertion of an additional problem parameter - the boat's turn speed. And algorithm can be improved with a simple complication. Another issue - stream speed dependence must be known beforehand. The algorithm can't work if stream speed is evaluated in real-time. But the idea can be spelled to provide such an algorithm, which would do a similar thing: piecing out the stream shift while the boat is where stream speed is least, or in other words - to use the stream as an ally.

But a very big advantage of the algorithm is next: it provides target reach for a fixed number of iterations. Sufficient conditions of target reach are much stronger because include sufficient conditions for the aiming method. And while forming this document I can say confidently: this document was knitting from R Markdown for about 5 minutes due to the aiming method, while the proposed algorithm run off for about 5-10 seconds. I had to wait such a lot of time to see the knitted PDF and check for mistakes.

Conclusions

- Formulated must conditions to solve the problem with the aiming method.
- Relations between boat speed and stream speed affects very much on ability to reach target. And also they affect on time for target reaching.
- Distance to target affects the ability to reach the target and time which is natural.
- Angle ϕ affects time. It defines how long will boat move and whether against or along stream. So it affects the time and defines trace.
- Stream dependence function is the most fluent and it directly defines whether the boat reaches the target. It affects time very much if the boat moves by aiming because for different functions with the same mean value time should be different. Probably this parameter must be first while choosing the method to solve the problem.
- Formulated and proved sufficient conditions to solve the problem if the boat moves between $y = 0$ and $y = y^*$ lines.
- Proposed an algorithm which provides problem solution if sufficient conditions satisfied.
- The proposed algorithm is coded and tested.
- The proposed algorithm is compared with the aiming method. Found out that it gives better result when the stream is faster near the target than near the origin (like it used to be on rivers).
- Advantages and disadvantages found for the proposed algorithm.