Middle East Technical University
Department of Computer Engineering

CENG 140
[Section 3]

Spring '21

Worksheet 1

Regulations

Due date: 17 May 2021, Monday *(Not subject to postpone)*

Submission: Electronically. You will be submitting your program source code written in a file which you will name as `the1.c` through the ODTUCLASS system. In a couple of days you will find on ODTUCLASS instructions on the way of submission. Resubmission is allowed (till the last moment of the due date), the last will replace the previous, provided you answer the interactive question positively.

Team: THIS IS AN EXAM. There is **no** teaming up. The takehome exam has to be done/turned in individually.

Cheating: All parts involved (source(s) and receiver(s)) get zero. **Yes, we definitely mean it!**

Introduction

Welcome to the series of Ceng. 140 take-home exams for section 3. The first is about a simplified version of the so called *Zermelo Navigation Problem*, discussed by E. Zermelo in 1931.

The problem is optimizing the crossing of a river with a boat that has a maximal velocity with respect to the river current. The hardness of the problem heavily varies depending on the non uniformity of the speed of the river.

In our simplified version:

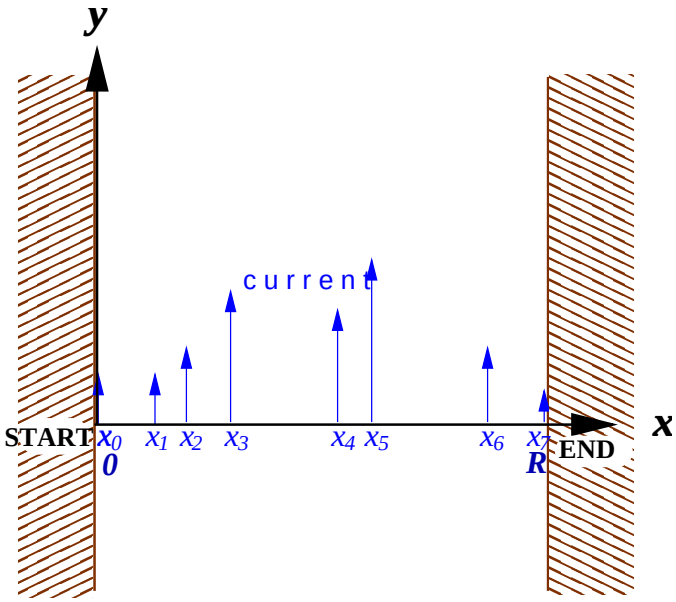
- The banks of the river are parallel.
- The line joining the start and end points of the route is perpendicular to the river bank.
- The boat will strictly travel on this line.
- The magnitude of the current in the river may vary but is strictly parallel to the banks at every point. The sense of the current may change.

Problem

The purpose is to calculate the time of travel from start to end.

You will be given the magnitude of the current at several points on the crossing line. To do a proper calculation you need to *interpolate* this knowledge to **any** point on the line. This will be done by *non-uniform quadratic spline interpolation*. After this process you will have a function to hand that provides the magnitude of the current at any point on the crossing line. How to do quadratic interpolation is explained in the relevant subsection below.

The boat velocity (with respect to the current) is of a fixed magnitude and will be given to you. The direction of this constant magnitude vector is altered (at every point, by the navigator) so that the velocity component in the direction of the current will cancel the current out. This, naturally, will have an effect of a reduced velocity in the direction of the crossing line. As greater the current gets the boat will turn towards the current and cancel out its effect. It is assured that the magnitude of the current will always remain smaller then the magnitude of the velocity of the boat. So, the boat will never stop on the crossing line.



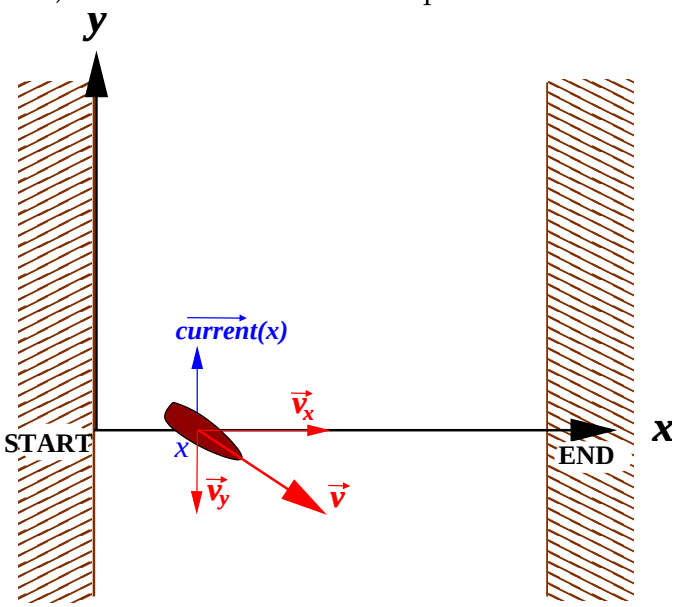
The figure on the left pictures the trajectory of the boat. The coordinate system is defined to have its origin at the start point and the x-axis passing through the end point. The end point's position on the x-axis will be given as a positive number as one of the input parameters. The y axis coincides with the starting bank. Though the example figure have all its currents in the positive y direction, this has not to be so, it is possible that the current changes sense (that means it becomes negative) at several points on the crossing line.

Since the actual current values are only given at a restricted number of points on the x-axis, when we need to have the value for a point which is in between, some intelligent guess has to be made. So, for example, if the current value at a point p which is between x_3 and x_4 is needed then an intelligent guess (we call this interpolation) could be the assumption of a linear relation that runs from x_3 to x_4. Making this assumption we can calculate a value for the point p, based on the distance p is located from x_3 and x_4. This type of a linearity assumption has its problems. The main problem is about the continuity of the derivative. At the actual points x_i (at which values are given) the slope of the curve will be undefined and this is far from reality. A better approach is to assume that the relation is not linear but quadratic. So, we have the chance to equate and have a single slope when we approach a 'known' point from left and right. All this is done ages ago and we have a computational technique (which is explained below in its relevant subsection) that gives as a smooth and nice behaving function $current(x)$ when we feed in the x_i values and the $current(x_i)$ values.

Now, let us have a closer look at how the boat proceed from one bank to the other. Since, we are working in a digitized environment physical quantities have to be discretized. Among this is the time domain. In our problem we will consider the events followed by small (but **not** infinitesimal) time increments. Let us call this time step Δt. This time step Δt will be given as an input parameter of the problem. During each Δt interval all varying parameters remain constant. The

laws of Physics provide the new position and speed of the boat that will hold for the next time step. Actually, this is not far from reality. In the theory of Physics, this Δt is taken to the limit of 0 (zero), and mathematics provides the analytic relations of the dynamics. We will not follow this route, and will not cook up our results analytically (i.e. we will not compute a 'formula' that will give us the total time of travel) but merely do a computation that walks through all the event (the event where the boat starts on one bank, and ends at the other) by Δt time increments. We start the 'simulation' at $x = 0$ with $t = 0$ and stop it when $x = R$ (R is the river width). The answer of the problem is what value the t (time) mark has reached at the stop moment.

At each time instance the boat is at a new position x. Depending on this position, the interpolation provides you with a value for the current. Since, it is said to be so, the boat navigator orients the boat so that the effect of the current is canceled by the y component of the boats speed vector (to make the decision and reorient the boat takes no time). You have a fixed magnitude for the boat velocity vector. This implies that you have a well computable x component of the velocity (v_x). The boat will travel exactly an amount of $\Delta x = v_x \times \Delta t$. So, after this Δt time has passed, the new position of the boat is Δx to the right of its former position. You have to repeat this loop until you reach $x = R$ where you have reached the other bank. At this point you output the time accumulated and stop your program.



Non-uniform quadratic spline interpolation

[Adapted from 'Spline interpolation' on Wikipedia]
Given are $n + 1$ distinct **knots** x_i such that

$$x_0 < x_1 < x_2 < \cdots x_{n-1} < x_n$$

and $n + 1$ **knot values** $current_i$. The aim is to provide a function $current(x)$ that has

$$current(x_i) = current_i \quad \forall\ x_i \text{ (knots)}$$

and

$$current(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases}$$

The *quadratic spline* function is constructed as

$$S_i(x) = current_i + \alpha_i(x - x_i) + \frac{\alpha_{i+1} - \alpha_i}{2(x_{i+1} - x_i)}(x - x_i)^2$$

The coefficients can be found (one by one) by

$$\alpha_{i+1} = -\alpha_i + 2\frac{current_{i+1} - current_i}{x_{i+1} - x_i}$$

The first coefficient α_0 is supposed to be taken as $\alpha_0 = 0$.

(A short note about the name of the procedure: 'Non-uniform' means the distance between the successive knots is not fixed. 'Quadratic' means the $S_i(x)$ are polynomials of second degree. 'Spline' means, that the interpolation is a juxtaposition of more then one functions.)

Specifications

- Input starts with two lines:

  First line: $\Delta t \;\; |\vec{v}|$ (both are `double`, separated by at least one blank)
  Second line: n (a positive integer in the range [3,100])

  The following $n + 1$ lines contain the x_i value followed by the $current_i$ information (both are `double`, separated by at least one blank) :

0	$current_0$
x_1	$current_1$
x_2	$current_2$
$\vdots$	$\vdots$
x_{n-1}	$current_{n-1}$
R	$current_n$

  As you have observed, the first line of this information starts with 0 (zero), that is the value of x_0, and the last line starts with R, which is the value of x_n. This (the positions of x_0 and x_n) will be so for all our inputs. Needless to say, there is no restriction on the remaining x_i values except that they will in an increasing order.

- No need to do an error check the input. We will not use erroneous input in evaluation. Furthermore, no nonsense input will be given (eg. a $\Delta t >$Total time!)

- The output is going to be a single `double` number. The total time the boat will take to reach the other side of the bank assuming the departure time as 0. There is a tolerance in the output and it can deviate $\pm\Delta t$. Any result that falls in this range will get full grade.

- We do not mention any units, since all the input/output are full compatible. The velocities, distances, and the time are all given in compatible units. You do not have to convert them to any other unit or multiple of a unit.

- The header information for all the mathematical functions (like `sqrt`, or any trigonometric function) present in Ansi C is located in `math.h`. This library is not auto-linked. You have to tell the linker about it by an `-lm` command line argument to the gcc (compiler+linker).

- For any floating point value to be stored use `double` variables.

- Though mathematically $current(x)$ is a function, it does not have to be coded as a C function. Simply calculate the value from the formula when you need it. Actually, there is no need to code any C function except `main()` in this take home exam.
- In no way beautify your output. The output is a single `double` number and that is all. Your programs will be checked by an evaluator program. Any other output will lead to a misevaluation.
- There is a helper page setup at:
  `http://144.122.71.48/zermelo`
  You can fetch a Linux executable of a helper program from there. The helper program, in addition to doing some file generation, also provides the solution value of the THE1 output. Also you will find some aid for visualization.