# MATH36031 Project 2 - deadline 24th November 2023, time 1100hrs.

In this project, the dynamics between a fox and a rabbit will be investigated, by solving differential equations modelling their positions at different times. The initial configuration is shown in Figure 1, where the fox starts chasing the rabbit while the rabbit tries to escape from its predator and moves towards its burrow. The fox is initially located at the origin $\mathbf{O}(0,0)$ and the rabbit is at $(0, 800)$. There is a circular fence with an opening $\mathbf{G}$ at $(0, 300)$. The rabbit moves in a circular path of radius 800 with speed $s_r$[1] towards its burrow. The rabbit's burrow is located at $800(-\sin(\pi/3), \cos(\pi/3))$. The path of the fox is initially directed straight towards $\mathbf{G}$ with speed $s_f$. After having reached the opening at $\mathbf{G}$, the subsequent path the fox takes in trying to catch the rabbit depends on whether its view of the rabbit is blocked by the straight line fence between $\mathbf{A}$ and $\mathbf{E}$ or not, as follows:

- if after having passed through the opening $\mathbf{G}$ the rabbit is in sight, the fox's attack path points directly towards the rabbit (the direction of the velocity vector of the fox is exact from the fox to the rabbit);

- if the view of the rabbit is blocked by an impenetrable straight line fence $\mathbf{AE}$, see figure 1, then the fox runs directly towards the corner $\mathbf{A}$. The coordinates of $\mathbf{AE}$ are

---

[1]The subscripts $f$ and $r$ denote the fox and rabbit respectively. The units of the coordinates are metres.
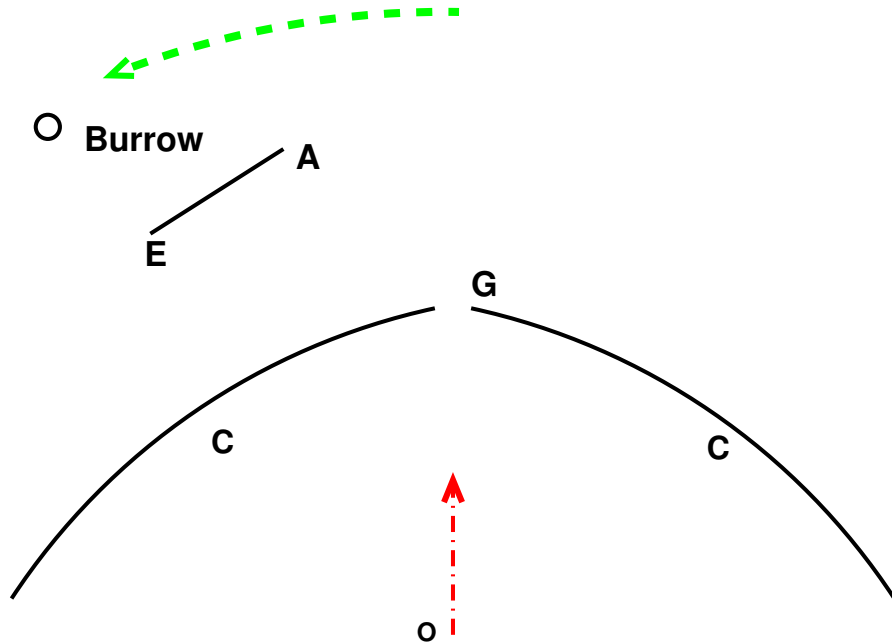


Figure 1: Schematic diagram showing the geometry and obstacles with $\mathbf{O}$ denoting the origin of coordinates. There is a circular fence $\mathbf{C}$ with a small opening at $\mathbf{G}$ $(0, 300)$. There is a fence at $\mathbf{AE}$ with $\mathbf{A}$ $(-350, 620)$ and $\mathbf{E}(-500, 350)$. The dashed green line shows the path followed by the rabbit towards it burrow. The dashed-dotted red line shows the initial path of the fox.

**A**$(-350, 620)$ and **E**$(-500, 350)$. Once the fox can see the rabbit again its attack path points directly towards the rabbit.

**Question 1: Constant speeds..** Assuming that both the fox and the rabbit run with constant speeds $s_f = s_{f0} = 17m/s$ and $s_r = s_{r0} = 12m/s$ respectively, determine whether the rabbit can be captured before it reaches its burrow. The rabbit is considered to be captured by the fox, if the distance between them is smaller than or equal to 0.1 meter.

**Question 2: Diminishing speeds.** Let us consider a more realistic scenario, when the hungry fox meets the tired rabbit. Because neither the fox nor the rabbit are in their best conditions, their chasing/escaping speeds diminish in time, according to the amount of distance (starting from the time they find each other and start running) they have travelled so far. More precisely, their speeds at time $t$ are given by

$$s_f(t) = s_{f0}e^{-\mu_f d_f(t)}, \quad s_r(t) = s_{r0}e^{-\mu_r d_r(t)},$$

where $s_{f0} = 17m/s$ and $s_{r0} = 12m/s$ are the same initial speeds as above, $\mu_f = 0.0002m^{-1}$ and $\mu_r = 0.0008m^{-1}$ are the rates of the diminishing speeds, $d_f(t)$ and $d_r(t)$ are the distance they have travelled up to time $t(> 0)$. Determine whether the rabbit can be captured before it reaches its burrow. (You may assume that this diminishing speed starts from $t = 0$).

**Outputs required** You are required to submit a report (maximum 8 pages including any appendices) in pdf form via the Turnitin submission box on Blackboard. Additionally you need to submit your m-files used for the MATLAB codes via the Blackboard Submission Box also on Blackboard. As part of the required output, in your report give (i) the time $T$ and the location of the fox when either the rabbit is captured or the rabbit escapes to the burrow; (ii) the distance travelled by the fox in time $T$. Additionally provide a plot showing the paths taken by both animals.

**Additional information and guidelines**

1. All coding must be done in MATLAB.

2. Treat both the fox and the rabbit as points, without worrying about their finite sizes (as in most models).

3. The questions can be answered with different approaches, **but you need to use the built-in ODE solver** `ode45` discussed during the lectures.

4. Avoid using hard-coded numbers. Any number in your code should either be given as initial condition, or be derived from these conditions.

5. Keep to the page length not exceeding eight A4 pages, and there is no need for a title page or abstract for a relative short report like this. Font sizes should be no smaller than 11 point, and page margins no smaller than 2cm.

6. List the complete code of the whole function at the end of each question, or in an appendix. Make your source code more readable, by keeping the indentation and

stylistic features, and can be copied from your submitted. Your published results should be reproducable from the code attached.

7. Have a look at the generic rubric about how your report will be marked, and also the intended learning outcomes about what you are expected to achieve in the end.

8. Avoid copying (too many) sentences directly from the project description, and try to restate the problem with your own words or examples if possible.

9. You may use your report in the future as evidences of written work, so take it seriously.

10. Your target audience is a fellow student on your course: explain the questions so that the report can be understood without this project description and your approach could be implemented in another computer language like Python. The report should indicate to the reader how well you understand the problem and the approach you took. Your goal will be to communicate your solutions to another person rather than to show you've completed the assignment.

11. Balance the explanation of the approach and the comments in the code. Avoid under-commenting and over-commenting.

12. Aim for precision and clarity of writing.

13. Since there is no final exam, you are advised to spend at least 15 hours on each project, with additional self-study if you are less experience with computer programming. Remember for a 10 credit module like this one, you are expect to spend $100 = 10 \times 10$ hours in total (including lectures, labs, self-study and coursework).

14. Please do not put any personal information on the report, only your student ID number.

15. The submission for each project will be open two weeks before the deadline. Only your last submission will be marked, and anything submitted after the deadline will be treated late and any penalty will be applied by the Teaching and Learning Office in June according to the Undergraduate Student Handbook.

16. Whilst this project can be done without the use of any artificial intelligence (AI) software tools, **if you use any AI tools or software to help you with your project, you must mention this in the report. Please study the guidelines at**

    **https://manchester-uk.libanswers.com/teaching-and-learning/faq/264824**

    **on how to do this correctly.**

    **The content and accuracy of the report will be your responsibility alone, and any factually incorrect statements or mathematically incorrect content will be penalised**.

17. Your attention is also drawn to to the University's Academic Malpractice Policy, see

    https://documents.manchester.ac.uk/display.aspx?DocID=639. See also the guidelines on the use of AI at:

    https://studentnews.manchester.ac.uk/2023/10/30/4-things-you-need-to-know-about-our-universitys-new-ai-guidance/

18. We are obliged to report cases of suspected academic malpractice, and people may be subject to an additional oral assessment on the content of the report and codes submitted.