

Typically, in our practice, we rely on standard Python libraries and sometimes customize them to suit our specific needs. However, a critical competency lies in comprehending the fundamental algorithms and effectively implementing or customizing them in code, even when integrating with unfamiliar tools. This assessment serves as a platform for you to demonstrate two key abilities: (a) your capacity to undertake such tasks and (b) your aptitude for executing them meticulously and coherently — an essential attribute for fostering collaborative work environments. While your grades demonstrate proficiency in algorithms, we're also interested in your ability to integrate that knowledge effectively.

We're thrilled to offer you an opportunity to showcase your skills as part of our hiring process. As a part of this evaluation, we have a coding exercise for you.

Problem statement:

Your task is to solve the following initial value problem (IVP) using the Euler method to solve the following initial value problem (IVP) in Rust:

$$y' = \cos(t) - y, 0 \leq t \leq 5, y(0) = 1, \text{ with } n = 20$$

Furthermore, we kindly request that you submit your solution via **GitHub**. Whether you choose to make your repository public or private is entirely up to you.

If you successfully solve this problem, we have an optional challenge for you: Try increasing  $n$  to 100 or 1000 and plot the difference between the numerical solution and the analytical one (if derived). You may use any Rust plotting library or export to CSV for visualization.

The last date for submission is **July 5th, 2025**. You can submit your responses through the Google form attached [here](#).

Some suggestions to help you get started:

1. Consider starting with Python if you're new to Rust. This approach can provide a smoother transition, allowing you to grasp the concepts more easily before delving into Rust-specific syntax and features. Feel free to share any progress or insights you gain along the way.
2. We understand that you may refer to external sources while working on the exercise. However, we expect each applicant to have a clear and full understanding of what they are submitting, and to be able to explain in detail the working of their solution during the interview.

3. This is essential, unless they proactively realise on their own that they should be able to do this — in which case, we're impressed :)
4. In terms of expectations, we encourage you to incorporate the following elements into your work:
  - Utilize code comments to explain the rationale behind your implementation and any notable design decisions.
  - Maintain clean code practices, considering tools like rustfmt to ensure consistent formatting and readability.
  - Adopt sensible naming conventions for functions and variables, enhancing code clarity and maintainability.
  - Aim for multiple git commits throughout your project, providing a clear record of your progress and facilitating collaboration.
  - By adhering to these guidelines, you'll not only demonstrate your technical proficiency but also showcase your commitment to producing high-quality, well-organized code.

We look forward to receiving your submissions and potentially welcoming you to our team.