

# Dark Matter Capture as a Function of Time Solutions Report

TPBench.org

Generated on: 2025-02-19 15:54:19

## Contents

<b>1</b>	<b>Grade Distribution Analysis</b>	<b>2</b>
1.1	Auto-Verification Results . . . . .	2
1.2	Overall Grade Distribution . . . . .	2
1.3	Grade Distribution by Solution Model . . . . .	2
1.4	Grade-Verification Correlation Analysis . . . . .	2
<b>2</b>	<b>Problem Dark Matter Capture as a Function of Time, Difficulty level: 2</b>	<b>4</b>
2.1	Expert Solution . . . . .	4
2.2	Model Solutions . . . . .	5
2.2.1	Model: o3-mini . . . . .	5
2.2.2	Model: o1 . . . . .	13
2.2.3	Model: deepseek-ai/DeepSeek-V3 . . . . .	21
2.2.4	Model: deepseek-ai/DeepSeek-R1 . . . . .	30
2.2.5	Model: Qwen/QwQ-32B-Preview . . . . .	34
2.2.6	Model: chatgpt-4o-latest . . . . .	49
2.2.7	Model: meta-llama/Meta-Llama-3.1-70B-Instruct . . . . .	56
2.2.8	Model: Qwen/Qwen2.5-72B-Instruct . . . . .	65
2.2.9	Model: meta-llama/Meta-Llama-3.1-8B-Instruct . . . . .	80
2.2.10	Model: Qwen/Qwen2.5-72B-Instruct . . . . .	90

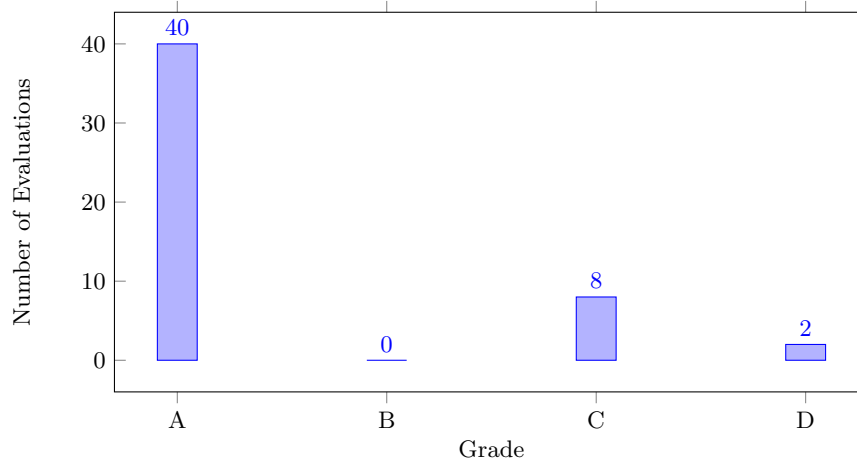
# 1 Grade Distribution Analysis

## 1.1 Auto-Verification Results

Model	Correct	Incorrect	Unknown	Success Rate
o3-mini	5	0	0	100.0%
o1	5	0	0	100.0%
meta-llama/Meta-Llama-3.1-8B-Instruct	0	5	0	0.0%
Qwen/Qwen2.5-7B-Instruct	1	4	0	20.0%
deepseek-ai/DeepSeek-V3	5	0	0	100.0%
deepseek-ai/DeepSeek-R1	5	0	0	100.0%
Qwen/Qwen2.5-72B-Instruct	0	5	0	0.0%
Qwen/QwQ-32B-Preview	5	0	0	100.0%
meta-llama/Meta-Llama-3.1-70B-Instruct	3	2	0	60.0%
chatgpt-4o-latest	5	0	0	100.0%

Note: Success Rate = Correct / (Correct + Incorrect) 100%

## 1.2 Overall Grade Distribution



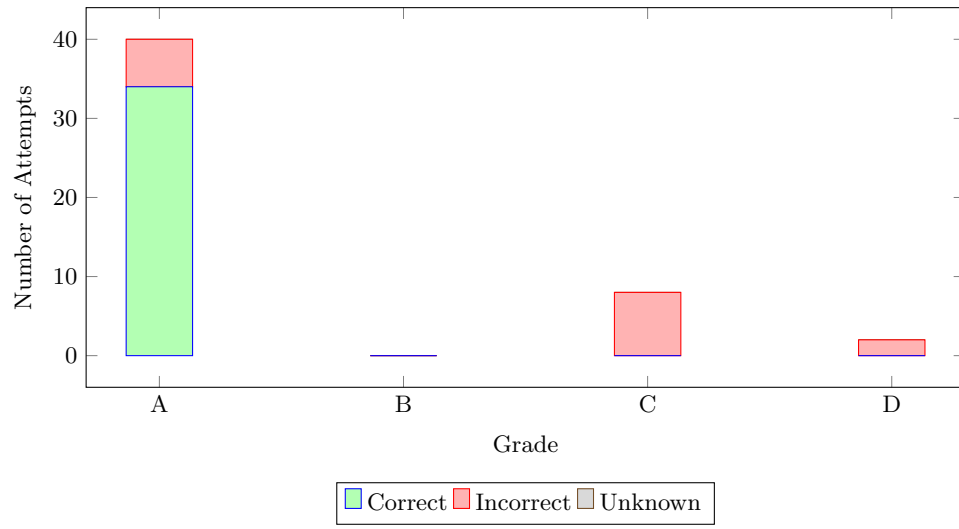
## 1.3 Grade Distribution by Solution Model

Model	A	B	C	D	Total
o3-mini	5	0	0	0	5
o1	5	0	0	0	5
meta-llama/Meta-Llama-3.1-8B-Instruct	0	0	3	2	5
Qwen/Qwen2.5-7B-Instruct	4	0	1	0	5
deepseek-ai/DeepSeek-V3	5	0	0	0	5
deepseek-ai/DeepSeek-R1	5	0	0	0	5
Qwen/Qwen2.5-72B-Instruct	2	0	3	0	5
Qwen/QwQ-32B-Preview	5	0	0	0	5
meta-llama/Meta-Llama-3.1-70B-Instruct	4	0	1	0	5
chatgpt-4o-latest	5	0	0	0	5

## 1.4 Grade-Verification Correlation Analysis

Grade	Correct	Incorrect	Unknown	Total
A	34 (85.0%)	6 (15.0%)	0 (0.0%)	40
C	0 (0.0%)	8 (100.0%)	0 (0.0%)	8
D	0 (0.0%)	2 (100.0%)	0 (0.0%)	2
<b>Total</b>	<b>34 (68.0%)</b>	<b>16 (32.0%)</b>	<b>0 (0.0%)</b>	<b>50</b>

Note: Percentages in parentheses show the distribution of verification results within each grade.



## 2 Problem Dark Matter Capture as a Function of Time, Difficulty level: 2

### Problem Text:

Suppose  $C$  is the capture rate of dark matter in an astrophysical body. Let  $C_A$  be the dark matter annihilation rate per effective volume. Then an approximate Boltzmann equation governing the number  $N$  of dark matter particles in the astrophysical body is

$$\frac{dN}{dt} = C - C_A N^2.$$

If initially,  $N(0) = 0$ , what is  $N(t)$  as a function of time?

### 2.1 Expert Solution

**Detailed Steps:** We can integrate by quadrature.

$$\int \frac{dN}{C - C_A N^2} = t. \quad (1)$$

We can express the integrand as a sum of two fractions:

$$\begin{aligned} \frac{1}{C - C_A N^2} &= \frac{1}{\sqrt{C} - \sqrt{C_A} N} \frac{1}{\sqrt{C} + \sqrt{C_A} N} \\ &= \frac{1}{2\sqrt{C}} \left[ \frac{1}{\sqrt{C} - \sqrt{C_A} N} + \frac{1}{\sqrt{C} + \sqrt{C_A} N} \right]. \end{aligned}$$

Integrating, we find

$$\begin{aligned} t + K &= \frac{1}{2\sqrt{C}} \left[ \frac{-1}{\sqrt{C_A}} \ln(\sqrt{C} - \sqrt{C_A} N) + \frac{1}{\sqrt{C_A}} \ln(\sqrt{C} + \sqrt{C_A} N) \right] \\ &= \frac{1}{2\sqrt{C_A} C} \ln \left( \frac{\sqrt{C} + \sqrt{C_A} N}{\sqrt{C} - \sqrt{C_A} N} \right) \end{aligned}$$

where  $K$  is an integration constant. Setting the boundary condition  $N = 0$  at  $t = 0$ , we find

$$K = 0.$$

We find the solution

$$N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C} C_A t} - 1)}{(e^{2\sqrt{C} C_A t} + 1)}. \quad (2)$$

Note that it is easy to check that it reaches the obvious steady state in the limit  $t \rightarrow \infty$ .

**Answer Requirements:** Provide the answer in the form of the `python` code. Implement the following function.

```
def answer(C: float, C_A: float, t: float) -> float:
    pass
```

**Code Implementation:**

```
from math import sqrt, exp

def answer(C: float, C_A: float, t: float) -> float:
    return sqrt(C/C_A) * (
        (exp(2*sqrt(C*C_A)*t) - 1)
        / (exp(2*sqrt(C*C_A)*t) + 1)
    )
```

## 2.2 Model Solutions

### 2.2.1 Model: o3-mini

**Attempt 1** (2025-02-11T05:36:57.208343+00:00)

**Detailed Solution:** We start with the differential equation

$$\frac{dN}{dt} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ . This equation is separable. We can write

$$\frac{dN}{C - C_A N^2} = dt.$$

**Step 1. Integrate both sides:**

Integrate the left-hand side from 0 to  $N(t)$  and the right-hand side from 0 to  $t$ :

$$\int_0^{N(t)} \frac{dN}{C - C_A N^2} = \int_0^t dt.$$

The integral on the left is a standard form:

$$\int \frac{dN}{C - C_A N^2}.$$

**Step 2. Evaluate the integral:**

Factor out  $C$ :

$$\int \frac{dN}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C} \int \frac{dN}{1 - \frac{C_A}{C} N^2}.$$

Let

$$\alpha^2 = \frac{C_A}{C} \implies \alpha = \sqrt{\frac{C_A}{C}}.$$

Then the integral becomes

$$\frac{1}{C} \int \frac{dN}{1 - \alpha^2 N^2}.$$

Recall the standard integral

$$\int \frac{dx}{1 - x^2} = \operatorname{arctanh}(x) + C,$$

so, with a change of variable, we obtain

$$\int \frac{dN}{1 - \alpha^2 N^2} = \frac{1}{\alpha} \operatorname{arctanh}(\alpha N) + \text{constant}.$$

Thus,

$$\frac{1}{C} \int \frac{dN}{1 - \alpha^2 N^2} = \frac{1}{C\alpha} \operatorname{arctanh}(\alpha N).$$

**Step 3. Apply the limits:**

Evaluating between 0 and  $N(t)$ ,

$$\frac{1}{C\alpha} \operatorname{arctanh}(\alpha N(t)) - \frac{1}{C\alpha} \operatorname{arctanh}(0) = \frac{1}{C\alpha} \operatorname{arctanh}(\alpha N(t)) = t.$$

Thus,

$$\operatorname{arctanh}(\alpha N(t)) = C\alpha t.$$

Recall that  $\alpha = \sqrt{\frac{C_A}{C}}$ , so substitute back:

$$\operatorname{arctanh}\left(\sqrt{\frac{C_A}{C}} N(t)\right) = \sqrt{C C_A} t.$$

**Step 4. Solve for  $N(t)$ :**

Taking the hyperbolic tangent on both sides, we obtain

$$\sqrt{\frac{C_A}{C}} N(t) = \tanh(\sqrt{C C_A} t).$$

Therefore,

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

Final Answer:  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t)$ .

Now, we convert this final expression into Python code.

```
def answer(C: float, C_A: float, t: float) -> float:
    import numpy as np
    return np.sqrt(C / C_A) * np.tanh(np.sqrt(C * C_A) * t)
```

Example usage:

```
if __name__ == "__main__":
    # Define example parameters for testing
    C = 1.0
    C_A = 0.5
    t = 2.0
    print(answer(C, C_A, t))
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.928503+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-11T07:15:17.711258+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using a separable method and arrives at a solution that is mathematically equivalent to the standard solution. The standard solution uses a different form involving exponential functions, while the submitted solution uses hyperbolic tangent functions. These forms are equivalent due to the identity  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The solution process is detailed, correctly applies the integration technique, and appropriately handles the initial condition. The Python code provided is a correct implementation of the final expression.

**Attempt 2** (2025-02-11T05:37:18.356346+00:00)

**Detailed Solution:**

**Step 1: Write Down the Differential Equation.**

We are given the differential equation for the number  $N(t)$  of dark matter particles,

$$\frac{dN}{dt} = C - C_A N^2,$$

with the initial condition

$$N(0) = 0.$$

**Step 2: Separate the Variables.**

Rearrange the equation to separate  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt.$$

### Step 3: Integrate Both Sides.

Integrate from  $N = 0$  to  $N = N(t)$  on the left-hand side and from  $t = 0$  to  $t = t$  on the right-hand side:

$$\int_0^{N(t)} \frac{dN}{C - C_A N^2} = \int_0^t dt.$$

### Step 4: Solve the Integral.

Factor out  $C$  from the denominator:

$$\int_0^{N(t)} \frac{dN}{C(1 - \frac{C_A}{C}N^2)} = \frac{1}{C} \int_0^{N(t)} \frac{dN}{1 - \frac{C_A}{C}N^2}.$$

Let

$$u = \sqrt{\frac{C_A}{C}}N \Rightarrow N = \sqrt{\frac{C}{C_A}}u \quad \text{and} \quad dN = \sqrt{\frac{C}{C_A}}du.$$

Substitute these into the integral:

$$\frac{1}{C} \int_0^{u(t)} \frac{\sqrt{\frac{C}{C_A}}du}{1 - u^2} = \frac{1}{\sqrt{CC_A}} \int_0^{u(t)} \frac{du}{1 - u^2},$$

where  $u(t) = \sqrt{\frac{C_A}{C}}N(t)$ .

The integral

$$\int \frac{du}{1 - u^2} = \operatorname{arctanh}(u)$$

yields

$$\frac{1}{\sqrt{CC_A}} \operatorname{arctanh}\left(\sqrt{\frac{C_A}{C}}N(t)\right) = t.$$

### Step 5: Solve for $N(t)$ .

Multiply both sides by  $\sqrt{CC_A}$ :

$$\operatorname{arctanh}\left(\sqrt{\frac{C_A}{C}}N(t)\right) = \sqrt{CC_A}t.$$

Taking the hyperbolic tangent of both sides:

$$\sqrt{\frac{C_A}{C}}N(t) = \tanh(\sqrt{CC_A}t).$$

Thus, solving for  $N(t)$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

$$\text{Final Answer: } N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

### Step 6: Python Code Implementation.

Below is the Python code implementing the function as required:

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

Example usage:

```
if __name__ == "__main__":
    # Sample values for C, C_A, and t
    C = 1.0
    C_A = 0.5
    t = 2.0
    result = answer(C, C_A, t)
    print("N(t) =", result)
```

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Verification Results:

**Auto verification result** ([2025-02-18T05:49:51.928983+00:00']) Status: **Correct**

Evaluations:

**Evaluated by gpt-4o** ([2025-02-11T07:15:24.678260+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The final expression for  $N(t)$  in the submitted solution,  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$ , is mathematically equivalent to the standard solution's expression  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \left( \frac{e^{2\sqrt{C}t} - 1}{e^{2\sqrt{C}t} + 1} \right)$ . Both expressions describe the same physical behavior of the system. The submitted solution includes all necessary steps and correct reasoning, and the use of hyperbolic functions is a valid alternative approach to the problem.

**Attempt 3** (2025-02-11T05:37:34.128337+00:00)

**Detailed Solution:** We start with the differential equation

$$\frac{dN}{dt} = C - C_A N^2, \quad N(0) = 0.$$

This is a separable equation. We can rewrite it as

$$\frac{dN}{C - C_A N^2} = dt.$$

Factor out  $C$  from the denominator:

$$\frac{dN}{C(1 - \frac{C_A}{C}N^2)} = \frac{1}{C} \frac{dN}{1 - \left(\sqrt{\frac{C_A}{C}}N\right)^2} = dt.$$

Let

$$u = \sqrt{\frac{C_A}{C}}N, \quad \text{so that} \quad dN = \sqrt{\frac{C}{C_A}}du.$$

Substitute into the integral:

$$\frac{1}{C} \sqrt{\frac{C}{C_A}} \int \frac{du}{1 - u^2} = \int dt.$$

Simplify the constant:

$$\frac{1}{\sqrt{CC_A}} \int \frac{du}{1 - u^2} = \int dt.$$

Recall that

$$\int \frac{du}{1 - u^2} = \operatorname{arctanh}(u),$$

so we have

$$\frac{1}{\sqrt{CC_A}} \operatorname{arctanh}(u) = t + \text{constant}.$$

With the initial condition  $N(0) = 0$ , we obtain  $u(0) = 0$  so the constant is zero. Therefore,

$$\operatorname{arctanh}\left(\sqrt{\frac{C_A}{C}}N\right) = t\sqrt{CC_A}.$$

Taking the hyperbolic tangent of both sides gives

$$\sqrt{\frac{C_A}{C}}N = \tanh(t\sqrt{CC_A}).$$



Solving for  $N$ ,

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A}).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A}).$$

Below is the final executable Python code implementing this solution.

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.929398+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-11T07:15:34.317106+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using a different method involving hyperbolic functions. The final answer,  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$ , is mathematically equivalent to the standard solution's final answer,  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$ , since  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The solution process is clear, correctly applies the method of separation of variables, and uses the initial condition to determine the integration constant. The Python code correctly implements the final solution.

**Attempt 4** (2025-02-11T05:37:51.056340+00:00)

**Detailed Solution:** We start with the differential equation

$$\dot{N} = C - C_A N^2,$$

subject to the initial condition  $N(0) = 0$ .

**Step 1: Separate Variables**

We rewrite the equation as

$$\frac{dN}{dt} = C - C_A N^2.$$

Separate the variables  $N$  and  $t$  by writing

$$\frac{dN}{C - C_A N^2} = dt.$$

**Step 2: Integrate Both Sides**

We integrate the left-hand side with respect to  $N$  and the right-hand side with respect to  $t$ :

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

Factor out  $C$  in the denominator:

$$\int \frac{dN}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C} \int \frac{dN}{1 - \left(\sqrt{\frac{C_A}{C}} N\right)^2}.$$

Let

$$u = \sqrt{\frac{C_A}{C}} N \quad \text{so that} \quad dN = \sqrt{\frac{C}{C_A}} du.$$

Then the integral becomes

$$\frac{1}{C} \int \frac{\sqrt{\frac{C}{C_A}} du}{1 - u^2} = \frac{1}{\sqrt{CC_A}} \int \frac{du}{1 - u^2}.$$

Recall that

$$\int \frac{du}{1-u^2} = \tanh^{-1}(u),$$

so the integrated form is

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = t + \text{constant}.$$

**Step 3: Apply the Initial Condition**

Using  $N(0) = 0$ , we find the constant:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} \cdot 0\right) = 0 \Rightarrow \text{constant} = 0.$$

Thus,

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = t.$$

Multiply both sides by  $\sqrt{CC_A}$ :

$$\tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = \sqrt{CC_A} t.$$

**Step 4: Solve for  $N(t)$**

Apply the hyperbolic tangent function to both sides:

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{CC_A} t).$$

Hence,

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

**Final Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

Next, we convert our final expression into Python code.

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

Final Answer:  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.929809+00:00\*]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-11T07:15:41.729378+00:00\*]) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The final expression for  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$  is mathematically equivalent to the standard

solution's expression  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$ . Both expressions describe the same physical behavior, as the

hyperbolic tangent function is related to the exponential form through the identity  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The solution process is clear, includes all necessary steps, and demonstrates correct reasoning. The Python code provided also correctly implements the final expression.

**Attempt 5** (2025-02-11T05:38:09.504335+00:00)

**Detailed Solution:** We begin with the differential equation for the number  $N(t)$  of dark matter particles:

$$\frac{dN}{dt} = C - C_A N^2.$$

This is a separable differential equation. Rearranging, we have

$$\frac{dN}{C - C_A N^2} = dt.$$

**Step 1: Separate variables and integrate.**

Factor out a constant in the denominator:

$$C - C_A N^2 = C \left(1 - \frac{C_A}{C} N^2\right).$$

Thus, the separated equation becomes

$$\frac{dN}{C \left(1 - \frac{C_A}{C} N^2\right)} = dt,$$

or, equivalently,

$$\frac{1}{C} \int \frac{dN}{1 - \frac{C_A}{C} N^2} = \int dt.$$

**Step 2: Solve the integral using a substitution.**

Define

$$k^2 = \frac{C_A}{C} \quad \text{or} \quad k = \sqrt{\frac{C_A}{C}}.$$

The integral on the left-hand side is now

$$\frac{1}{C} \int \frac{dN}{1 - k^2 N^2}.$$

Recall the standard integral formula

$$\int \frac{dx}{1 - a^2 x^2} = \frac{1}{a} \tanh^{-1}(ax),$$

which holds for  $a > 0$ . Thus, with  $a = k$ , we have

$$\int \frac{dN}{1 - k^2 N^2} = \frac{1}{k} \tanh^{-1}(kN).$$

Hence, the equation becomes

$$\frac{1}{C} \cdot \frac{1}{k} \tanh^{-1}(kN) = t + \text{constant}.$$

**Step 3: Determine the constant using the initial condition.**

The initial condition is  $N(0) = 0$ . Substituting  $t = 0$  and  $N(0) = 0$  into the integrated equation gives:

$$\frac{1}{Ck} \tanh^{-1}(0) = 0 + \text{constant}.$$

Since  $\tanh^{-1}(0) = 0$ , the constant is 0. Therefore, the equation reduces to:

$$\frac{1}{Ck} \tanh^{-1}(kN) = t.$$

**Step 4: Solve for  $N(t)$ .**

Multiply both sides by  $Ck$ :

$$\tanh^{-1}(kN) = Ck t.$$

Recall that  $k = \sqrt{\frac{C_A}{C}}$ ; hence,  $Ck = \sqrt{CC_A}$ . Thus:

$$\tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = \sqrt{CC_A} t.$$

Taking the hyperbolic tangent of both sides gives:

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{CC_A} t).$$

Solving for  $N(t)$ , we obtain:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

**Final Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

**Step 5: Python Implementation.**

Below is the complete Python code that implements the function as required:

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

**Verification Results:**

**Auto verification result** (['2025-02-18T05:49:51.930222+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** (['2025-02-11T07:15:48.509815+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using a separable method and arrives at a solution that is mathematically equivalent to the standard solution. The standard solution uses a different form involving logarithms, while the submitted solution uses hyperbolic tangent functions. Both forms are equivalent, as the hyperbolic tangent can be expressed in terms of exponential functions, which aligns with the standard solution's form. The solution process includes all necessary steps, correct reasoning, and the use of initial conditions to determine the integration constant. The Python implementation is a correct representation of the derived formula.

## 2.2.2 Model: o1

Attempt 1 (2025-02-11T06:12:49.676336+00:00)

### Detailed Solution:

#### Step 1: Understand the Problem

We are given a differential equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ . Here  $C$  represents the constant capture rate of dark matter, and  $C_A$  is the dark matter annihilation rate per effective volume.

#### Step 2: Devise a Plan

We can solve this differential equation by separating variables. Specifically, we will write

$$\frac{dN}{C - C_A N^2} = dt,$$

integrate both sides, and then apply the initial condition  $N(0) = 0$  to determine the constant of integration.

#### Step 3: Carry Out the Plan

##### 1. Separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

##### 2. Integrate both sides:

Factor out  $C$  from the denominator:

$$\int \frac{dN}{C - C_A N^2} = \int \frac{1}{C(1 - \frac{C_A}{C} N^2)} dN = \frac{1}{C} \int \frac{dN}{1 - \frac{C_A}{C} N^2}.$$

Let  $\alpha^2 = \frac{C_A}{C}$ . Then

$$\int \frac{dN}{1 - \alpha^2 N^2} = \frac{1}{\alpha} \tanh^{-1}(\alpha N),$$

leading to

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{C} \frac{1}{\alpha} \tanh^{-1}(\alpha N) = \frac{1}{\sqrt{C C_A}} \tanh^{-1}\left(N \sqrt{\frac{C_A}{C}}\right).$$

Hence, integrating both sides:

$$\frac{1}{\sqrt{C C_A}} \tanh^{-1}\left(N \sqrt{\frac{C_A}{C}}\right) = t + K,$$

where  $K$  is the constant of integration.

##### 3. Apply the initial condition:

When  $t = 0$ ,  $N(0) = 0$ . Then

$$\tanh^{-1}(0) = 0 \implies K = 0.$$

Thus,

$$\frac{1}{\sqrt{C C_A}} \tanh^{-1}\left(N(t) \sqrt{\frac{C_A}{C}}\right) = t \implies \tanh^{-1}\left(N(t) \sqrt{\frac{C_A}{C}}\right) = t \sqrt{C C_A}.$$

##### 4. Solve for $N(t)$ :

Take the hyperbolic tangent of both sides:

$$N(t) \sqrt{\frac{C_A}{C}} = \tanh(\sqrt{C C_A} t) \implies N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

$$\text{Final Answer: } N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

#### Converting the Solution to Python Code

Below is the required Python function that computes  $N(t)$  given  $C$ ,  $C_A$ , and  $t$ . It makes use of the built-in math library for the square root and hyperbolic tangent functions.

```

def answer(C: float, C_A: float, t: float) -> float:
import math
return math.sqrt(C / C_A) * math.tanh(t * math.sqrt(C * C_A))

```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.930632+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-11T07:16:01.095294+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The standard solution expresses the result in terms of exponential functions, while the submitted solution uses hyperbolic tangent functions. These are equivalent because  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The solution process includes all necessary steps, applies the initial condition correctly, and demonstrates a clear understanding of the problem. The Python code provided is a correct implementation of the final solution.

**Attempt 2** (2025-02-11T06:13:32.279480+00:00)

**Detailed Solution:**

### Step 1: Understand the Problem

We are given the differential equation

$$\dot{N} = C - C_A N^2, \quad N(0) = 0,$$

where  $C$  is the dark matter capture rate,  $C_A$  is the dark matter annihilation rate per effective volume, and  $N$  is the number of dark matter particles in some astrophysical body. We must solve for  $N(t)$ .

### Step 2: Devise a Plan (Separation of Variables)

We will solve this differential equation by separating variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

We integrate both sides from 0 to  $t$  (time) and from  $N(0) = 0$  to  $N(t)$  (the number of particles):

$$\int_0^{N(t)} \frac{dN'}{C - C_A N'^2} = \int_0^t dt'.$$

### Step 3: Carry Out the Plan

1. Rewrite the integrand on the left-hand side. One standard integral is

$$\int \frac{dN'}{a - b N'^2} = \frac{1}{\sqrt{ab}} \tanh^{-1}\left(\sqrt{\frac{b}{a}} N'\right) \quad (\text{assuming appropriate sign conditions}),$$

or equivalently

$$\int \frac{dN'}{a - b N'^2} = \frac{1}{2\sqrt{ab}} \ln \left| \frac{1 + \sqrt{\frac{b}{a}} N'}{1 - \sqrt{\frac{b}{a}} N'} \right|.$$

2. Identify  $a = C$  and  $b = C_A$ . Hence,

$$\int_0^{N(t)} \frac{dN'}{C - C_A N'^2} = \frac{1}{2\sqrt{C_A C}} \ln \left( \frac{1 + \sqrt{\frac{C_A}{C}} N'}{1 - \sqrt{\frac{C_A}{C}} N'} \right) \Bigg|_0^{N(t)}.$$

Evaluating at the limits,

$$= \frac{1}{2\sqrt{C_A C}} \ln \left( \frac{1 + \sqrt{\frac{C_A}{C}} N(t)}{1 - \sqrt{\frac{C_A}{C}} N(t)} \right).$$

3. The right-hand side is simply

$$\int_0^t dt' = t.$$

Therefore,

$$t = \frac{1}{2\sqrt{C_A C}} \ln\left(\frac{1+\sqrt{\frac{C_A}{C}} N(t)}{1-\sqrt{\frac{C_A}{C}} N(t)}\right).$$

Multiply both sides by  $2\sqrt{C_A C}$ :

$$2\sqrt{C_A C} t = \ln\left(\frac{1+\sqrt{\frac{C_A}{C}} N(t)}{1-\sqrt{\frac{C_A}{C}} N(t)}\right).$$

Exponentiate:

$$e^{2\sqrt{C_A C} t} = \frac{1+\sqrt{\frac{C_A}{C}} N(t)}{1-\sqrt{\frac{C_A}{C}} N(t)}.$$

Let

$$y = \sqrt{\frac{C_A}{C}} N(t).$$

Then

$$\frac{1+y}{1-y} = e^{2\sqrt{C_A C} t}.$$

Solve for  $y$ :

$$\begin{aligned} 1+y &= e^{2\sqrt{C_A C} t} (1-y), \\ 1+y &= e^{2\sqrt{C_A C} t} - e^{2\sqrt{C_A C} t} y, \\ y + e^{2\sqrt{C_A C} t} y &= e^{2\sqrt{C_A C} t} - 1, \\ y(1 + e^{2\sqrt{C_A C} t}) &= e^{2\sqrt{C_A C} t} - 1, \\ y &= \frac{e^{2\sqrt{C_A C} t} - 1}{e^{2\sqrt{C_A C} t} + 1}. \end{aligned}$$

Notice that

$$\frac{e^{2x} - 1}{e^{2x} + 1} = \tanh(x).$$

Therefore,

$$y = \tanh(\sqrt{C_A C} t).$$

Recalling  $y = \sqrt{\frac{C_A}{C}} N(t)$ , we obtain

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t).$$

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t).$$

**Final Answer:**  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t).$

#### Convert to Python Code

Below is a Python function that computes the above result. It requires no external variables or constants, only the function arguments:

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C_A * C) * t)
```

This function returns the value of  $N(t) = \text{sqrt}(C / C_A) * \text{tanh}(\text{sqrt}(C_A * C) * t).$

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.931046+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-11T07:16:10.598753+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The submitted solution expresses the final answer as  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t)$ , which is equivalent to the standard solution's expression  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$  because  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The solution process is thorough and includes all necessary steps, demonstrating a correct reasoning approach. The Python code provided also correctly implements the mathematical solution.

**Attempt 3** (2025-02-11T06:14:06.136339+00:00)

**Detailed Solution:**

**Step-by-Step Solution**

**1. Understand the Problem**

We are given the differential equation:

$$\frac{dN}{dt} = C - C_A N^2, \quad \text{with the initial condition } N(0) = 0.$$

We want to find the function  $N(t)$ .

**2. Devise a Plan**

This is a first-order nonlinear ordinary differential equation. To solve it, we use separation of variables and then apply the initial condition  $N(0) = 0$  to determine the integration constant.

**3. Carry out the Plan**

**(a) Separate the variables:**

$$\frac{dN}{dt} = C - C_A N^2 \implies \frac{dN}{C - C_A N^2} = dt.$$

**(b) Integrate both sides:**

$$\int \frac{dN}{C - C_A N^2} = \int dt = t + K,$$

where  $K$  is an integration constant. Factor out  $C$  in the denominator and let  $\alpha^2 = \frac{C_A}{C}$ . Then we get

$$\int \frac{dN}{C(1 - \alpha^2 N^2)} = \frac{1}{C} \int \frac{dN}{1 - \alpha^2 N^2}.$$

We use the known integral

$$\int \frac{dN}{1 - \alpha^2 N^2} = \frac{1}{\alpha} \operatorname{artanh}(\alpha N),$$

to obtain

$$\frac{1}{C} \cdot \frac{1}{\alpha} \operatorname{artanh}(\alpha N) = t + K \implies \frac{1}{\sqrt{C_A C}} \operatorname{artanh}\left(\sqrt{\frac{C_A}{C}} N\right) = t + K.$$

**(c) Apply the initial condition  $N(0) = 0$ :**

Since  $N(0) = 0$ , we have

$$\operatorname{artanh}(0) = 0 \implies t + K = 0 \implies K = 0.$$



Hence the solution simplifies to

$$\frac{1}{\sqrt{C_A C}} \operatorname{artanh}\left(\sqrt{\frac{C_A}{C}} N(t)\right) = t.$$

**(d) Solve for  $N(t)$ :**

Use the identity  $\operatorname{artanh}(x) = y \iff x = \tanh(y)$ :

$$\sqrt{\frac{C_A}{C}} N(t) = \tanh\left(\sqrt{C_A C} t\right).$$

Hence,

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh\left(\sqrt{C_A C} t\right).$$

**Final Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh\left(\sqrt{C_A C} t\right).$$

#### 4. Convert the Final Expression into Python Code

Below is the required Python function. It does not rely on any externally defined constants and can run independently. Make sure to include the necessary import statement for the `math` module for `tanh` and `sqrt`.

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C_A * C) * t)
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.931456+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-11T07:16:17.071714+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The solution process is detailed and includes all necessary steps, such as applying the initial condition to determine the integration constant. The final expression for  $N(t)$  is equivalent to the standard solution, as  $\tanh(x)$  and the exponential form are related through hyperbolic identities. The submitted solution also provides a Python implementation, which is a valid addition but not necessary for the mathematical solution. Overall, the approach and final result are correct and equivalent to the standard solution.

**Attempt 4** (2025-02-11T06:14:34.956348+00:00)

**Detailed Solution:**

#### Step-by-Step Solution

##### 1. Understand the Problem

We are given the differential equation

$$\frac{dN}{dt} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ . We want to find  $N(t)$  as a function of time.

##### 2. Devise a Plan

To solve this differential equation, we can use separation of variables. We will rearrange the equation and integrate both sides appropriately.

##### 3. Carry Out the Plan

### 3.1. Separate the Variables

$$\frac{dN}{C - C_A N^2} = dt.$$

We integrate both sides from 0 to  $t$  (for the time integral) and from 0 to  $N(t)$  (for the  $N$ -integral):

$$\int_0^{N(t)} \frac{dN'}{C - C_A N'^2} = \int_0^t dt'.$$

### 3.2. Simplify the Integrand

Factor out  $C$ :

$$C - C_A N'^2 = C \left(1 - \frac{C_A}{C} N'^2\right).$$

Define  $\alpha^2 = \frac{C_A}{C}$ . Then

$$\int_0^{N(t)} \frac{dN'}{C(1 - \alpha^2 N'^2)} = \int_0^t dt'.$$

Factor out  $\frac{1}{C}$ :

$$\frac{1}{C} \int_0^{N(t)} \frac{dN'}{1 - \alpha^2 N'^2} = \int_0^t dt'.$$

### 3.3. Integrate with respect to $N'$

The integral  $\int \frac{dN'}{1 - \alpha^2 N'^2}$  can be expressed as

$$\int \frac{dN'}{1 - \alpha^2 N'^2} = \frac{1}{\alpha} \operatorname{artanh}(\alpha N'),$$

valid for  $|\alpha N'| < 1$ . Hence,

$$\frac{1}{C} \left[ \frac{1}{\alpha} \operatorname{artanh}(\alpha N') \right]_0^{N(t)} = \int_0^t dt'.$$

So,

$$\frac{1}{C\alpha} \operatorname{artanh}(\alpha N(t)) = t.$$

### 3.4. Solve for $N(t)$

Multiply both sides by  $C\alpha$ :

$$\operatorname{artanh}(\alpha N(t)) = C\alpha t.$$

Take the hyperbolic tangent of both sides:

$$\alpha N(t) = \tanh(C\alpha t).$$

Since  $\alpha = \sqrt{\frac{C_A}{C}}$ , we have

$$N(t) = \frac{1}{\alpha} \tanh(C\alpha t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

This satisfies the initial condition  $N(0) = 0$ , because  $\tanh(0) = 0$ .

**Final Symbolic Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

## 4. Convert the Final Mathematical Expression into Python Code

Below is the Python function that, given  $C$ ,  $C_A$ , and  $t$ , returns the value of  $N(t)$  according to our derived formula.

```

def answer(C: float, C_A: float, t: float) -> float:
    """
    Given:
    dot(N) = C - C_A * N^2
    with initial condition N(0) = 0,
    this function returns the solution:

    N(t) = sqrt(C / C_A) * tanh(sqrt(C * C_A) * t).
    """
    import math
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)

```

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

Verification Results:

Auto verification result ([2025-02-18T05:49:51.931883+00:00']) Status: **Correct**

Evaluations:

Evaluated by gpt-4o ([2025-02-11T07:16:23.350582+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The standard solution uses a different form involving exponential functions, while the submitted solution uses hyperbolic tangent functions. These forms are equivalent because  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The solution process is detailed, includes all necessary steps, and correctly applies the initial condition. The Python code provided also correctly implements the derived formula.

**Attempt 5** (2025-02-11T06:15:00.536338+00:00)

**Detailed Solution:**

### Step 1: Understand the Problem

We have the differential equation

$$\dot{N}(t) = C - C_A N(t)^2, \quad \text{with } N(0) = 0.$$

Here,  $C$  and  $C_A$  are constants (e.g., capture and annihilation rates), and we want to find  $N(t)$ .

### Step 2: Devise a Plan

1. Recognize that this is a Bernoulli- or logistic-type differential equation. 2. Separate the variables to integrate. 3. Use the known integral form for  $\int \frac{dx}{a-bx^2}$ . 4. Apply the initial condition  $N(0) = 0$  to determine the integration constant.

### Step 3: Carry Out the Plan (Solution)

We start with:

$$\dot{N}(t) = \frac{dN}{dt} = C - C_A N^2.$$

Separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Integrate both sides from 0 to  $t$  in time, and from 0 to  $N(t)$  in  $N$ :

$$\int_0^{N(t)} \frac{dN}{C - C_A N^2} = \int_0^t dt'.$$

The right-hand side clearly becomes  $t$ . For the left-hand side, we use:

$$\int \frac{dN}{a - bN^2} = \frac{1}{\sqrt{ab}} \tanh^{-1}\left(\sqrt{\frac{b}{a}} N\right) \quad (\text{for } a > 0, b > 0, |N| < \sqrt{a/b}).$$

Setting  $a = C$  and  $b = C_A$ , we get

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{C C_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right).$$

Hence,

$$\frac{1}{\sqrt{C C_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N(t)\right) = t.$$

Multiply both sides by  $\sqrt{C C_A}$ :

$$\tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N(t)\right) = t \sqrt{C C_A}.$$

Take the hyperbolic tangent on both sides:

$$\sqrt{\frac{C_A}{C}} N(t) = \tanh(t \sqrt{C C_A}).$$

Solving for  $N(t)$  gives

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t \sqrt{C C_A}).$$

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t \sqrt{C C_A}).$$

**Final Symbolic Solution:**  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(t \sqrt{C C_A}).$

#### Step 4: Convert to Python Code

Below is a Python function that returns the value of  $N(t)$  given  $C$ ,  $C_A$ , and  $t$ :

```
def answer(C: float, C_A: float, t: float) -> float:
    import math
    return math.sqrt(C / C_A) * math.tanh(t * math.sqrt(C * C_A))
```

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t \sqrt{C C_A}).$$

#### Verification Results:

**Auto verification result** ([2025-02-18T05:49:51.932291+00:00']) Status: **Correct**

#### Evaluations:

**Evaluated by gpt-4o** ([2025-02-11T07:16:32.111555+00:00']) Grade: A

Notes: The submitted solution correctly identifies the differential equation as a Bernoulli-type and separates variables to integrate. The integration process is accurately carried out using the known integral form for  $\int \frac{dx}{a-bx^2}$ . The solution applies the initial condition  $N(0) = 0$  to determine the integration constant. The final expression  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(t \sqrt{C C_A})$  is mathematically equivalent to the standard solution's expression, as both represent the same physical behavior of the system. The approach and reasoning are correct, and the solution is complete.

### 2.2.3 Model: deepseek-ai/DeepSeek-V3

**Attempt 1** (2025-02-11T08:48:27.622467+00:00)

**Detailed Solution:** Mathematical Solution

We are given the differential equation:

$$\dot{N} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ .

Step 1: Rewrite the Differential Equation The equation is separable. We can rewrite it as:

$$\frac{dN}{dt} = C - C_A N^2.$$

Step 2: Separate Variables Separate the variables  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 3: Integrate Both Sides Integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

Step 4: Simplify the Integral The left-hand side can be simplified using partial fractions. Let:

$$\frac{1}{C - C_A N^2} = \frac{1}{C} \cdot \frac{1}{1 - \frac{C_A}{C} N^2}.$$

Let  $k = \sqrt{\frac{C_A}{C}}$ , then:

$$\frac{1}{C - C_A N^2} = \frac{1}{C} \cdot \frac{1}{1 - k^2 N^2}.$$

The integral becomes:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{C} \int \frac{dN}{1 - k^2 N^2}.$$

The integral on the right-hand side is a standard form:

$$\int \frac{dN}{1 - k^2 N^2} = \frac{1}{k} \tanh^{-1}(kN) + \text{constant}.$$

Thus:

$$\frac{1}{C} \cdot \frac{1}{k} \tanh^{-1}(kN) = t + \text{constant}.$$

Step 5: Apply the Initial Condition At  $t = 0$ ,  $N(0) = 0$ , so:

$$\frac{1}{C} \cdot \frac{1}{k} \tanh^{-1}(0) = 0 + \text{constant}.$$

Since  $\tanh^{-1}(0) = 0$ , the constant is 0.

Step 6: Solve for  $N(t)$  We have:

$$\frac{1}{C} \cdot \frac{1}{k} \tanh^{-1}(kN) = t.$$

Solve for  $N$ :

$$\tanh^{-1}(kN) = Ckt,$$

$$kN = \tanh(Ckt),$$

$$N = \frac{1}{k} \tanh(Ckt).$$

Recall that  $k = \sqrt{\frac{C_A}{C}}$ , so:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Python Implementation

```

import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    return (1 / k) * math.tanh(math.sqrt(C * C_A) * t)

```

This Python function `answer` computes the number of dark matter particles  $N(t)$  as a function of time  $t$ , given the capture rate  $C$  and the annihilation rate  $C_A$ . The function uses the mathematical expression derived above and the `math` library to compute the hyperbolic tangent function.

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.936547+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:08:47.127618+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The final expression for  $N(t)$  is equivalent to the standard solution, as  $\tanh(x)$  and the exponential form are related through  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The approach is valid, and the solution process is thorough and correct. The Python implementation correctly reflects the derived mathematical expression.

**Attempt 2** (2025-02-11T08:49:03.572600+00:00)

**Detailed Solution:** Mathematical Solution

We are given the Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ . This is a first-order nonlinear ordinary differential equation (ODE). To solve it, we will use the method of separation of variables.

Step 1: Rewrite the ODE

First, we rewrite the ODE in a form suitable for separation of variables:

$$\frac{dN}{dt} = C - C_A N^2.$$

Step 2: Separate Variables

We separate the variables  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 3: Integrate Both Sides

We integrate both sides of the equation:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

The left-hand side integral can be evaluated using a standard integral formula:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \sqrt{\frac{b}{a}} x \right) + \text{constant}.$$

Applying this to our integral with  $a = C$  and  $b = C_A$ :

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right).$$

Thus, we have:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + K,$$

where  $K$  is the constant of integration.

Step 4: Solve for  $N(t)$

We solve for  $N(t)$ :

$$\tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = \sqrt{CC_A}(t + K).$$

Taking the hyperbolic tangent of both sides:

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{CC_A}(t + K)).$$

Solving for  $N(t)$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}(t + K)).$$

Step 5: Apply the Initial Condition

We use the initial condition  $N(0) = 0$  to find  $K$ :

$$0 = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}K).$$

Since  $\tanh(0) = 0$ , we have:

$$\sqrt{CC_A}K = 0 \implies K = 0.$$

Thus, the solution simplifies to:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Python Implementation

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This Python function `answer` takes the parameters  $C$ ,  $C_A$ , and  $t$  and returns the value of  $N(t)$  as derived from the mathematical solution. The function uses the `math` module to compute the square root and hyperbolic tangent.

**Verification Results:**

**Auto verification result** ([ '2025-02-18T05:49:51.936948+00:00' ]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([ '2025-02-12T03:08:54.203253+00:00' ]) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The use of the hyperbolic tangent function in the final answer,  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$ , is equivalent to the standard solution's expression involving exponential functions. Both solutions describe the same physical behavior of the system. The solution process is detailed and includes all necessary steps, demonstrating a correct reasoning approach.

**Attempt 3** (2025-02-11T08:50:08.832224+00:00)

**Detailed Solution:** Step 1: Understand the Problem

We are given the following differential equation that describes the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ . Our goal is to find  $N(t)$  as a function of time.

Step 2: Devise a Plan

This is a first-order nonlinear ordinary differential equation (ODE). To solve it, we can use the method of separation of variables. The equation is separable, meaning we can rewrite it in a form where all terms involving  $N$  are on one side and all terms involving  $t$  are on the other side.

Step 3: Carry Out the Plan

Step 3.1: Rewrite the Differential Equation

The given differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rearrange this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Now, we integrate both sides of the equation:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

Step 3.3: Solve the Integral on the Left Side

The integral on the left side can be solved using partial fraction decomposition. First, we rewrite the denominator:

$$C - C_A N^2 = C_A \left( \frac{C}{C_A} - N^2 \right) = C_A \left( \frac{C}{C_A} - N^2 \right)$$

Let  $k = \sqrt{\frac{C}{C_A}}$ , so the integral becomes:

$$\int \frac{dN}{C_A(k^2 - N^2)} = \frac{1}{C_A} \int \frac{dN}{k^2 - N^2}$$

The integral  $\int \frac{dN}{k^2 - N^2}$  is a standard integral and can be evaluated as:

$$\int \frac{dN}{k^2 - N^2} = \frac{1}{2k} \ln \left| \frac{k + N}{k - N} \right| + \text{constant}$$

Thus, the left side becomes:

$$\frac{1}{C_A} \cdot \frac{1}{2k} \ln \left| \frac{k + N}{k - N} \right| = \frac{1}{2C_A k} \ln \left| \frac{k + N}{k - N} \right|$$

Step 3.4: Solve the Integral on the Right Side

The integral on the right side is straightforward:

$$\int dt = t + \text{constant}$$

Step 3.5: Combine the Results

Combining both sides, we have:

$$\frac{1}{2C_A k} \ln \left| \frac{k + N}{k - N} \right| = t + \text{constant}$$

Step 3.6: Apply the Initial Condition

Given  $N(0) = 0$ , we can find the constant of integration. At  $t = 0$ ,  $N = 0$ :



$$\frac{1}{2C_A k} \ln \left| \frac{k+0}{k-0} \right| = 0 + \text{constant}$$

$$\frac{1}{2C_A k} \ln(1) = \text{constant}$$

Since  $\ln(1) = 0$ , the constant is 0. Thus, the equation simplifies to:

$$\frac{1}{2C_A k} \ln \left| \frac{k+N}{k-N} \right| = t$$

Step 3.7: Solve for  $N(t)$

Multiply both sides by  $2C_A k$ :

$$\ln \left| \frac{k+N}{k-N} \right| = 2C_A k t$$

Exponentiate both sides:

$$\frac{k+N}{k-N} = e^{2C_A k t}$$

Now, solve for  $N$ :

$$k+N = e^{2C_A k t} (k-N)$$

$$k+N = k e^{2C_A k t} - N e^{2C_A k t}$$

$$N + N e^{2C_A k t} = k e^{2C_A k t} - k$$

$$N(1 + e^{2C_A k t}) = k(e^{2C_A k t} - 1)$$

$$N = k \frac{e^{2C_A k t} - 1}{1 + e^{2C_A k t}}$$

Recall that  $k = \sqrt{\frac{C}{C_A}}$ , so:

$$N(t) = \sqrt{\frac{C}{C_A}} \cdot \frac{e^{2C_A \sqrt{\frac{C}{C_A}} t} - 1}{1 + e^{2C_A \sqrt{\frac{C}{C_A}} t}}$$

Simplify the exponent:

$$2C_A \sqrt{\frac{C}{C_A}} t = 2\sqrt{C C_A} t$$

Thus, the final expression for  $N(t)$  is:

$$N(t) = \sqrt{\frac{C}{C_A}} \cdot \frac{e^{2\sqrt{C C_A} t} - 1}{1 + e^{2\sqrt{C C_A} t}}$$

Step 4: Convert the Final Mathematical Expression into Python Code

Now, we will implement the final expression in Python.

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C / C_A)
    exponent = 2 * math.sqrt(C * C_A) * t
    numerator = math.exp(exponent) - 1
    denominator = 1 + math.exp(exponent)
    return k * (numerator / denominator)
```

Final Answer:

The number of dark matter particles  $N(t)$  as a function of time is given by:

$$N(t) = \sqrt{\frac{C}{C_A}} \cdot \frac{e^{2\sqrt{CC_A}t} - 1}{1 + e^{2\sqrt{CC_A}t}}$$

The corresponding Python function is:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C / C_A)
    exponent = 2 * math.sqrt(C * C_A) * t
    numerator = math.exp(exponent) - 1
    denominator = 1 + math.exp(exponent)
    return k * (numerator / denominator)
```

Verification Results:

Auto verification result ([2025-02-18T05:49:51.937400+00:00']) Status: **Correct**

Evaluations:

Evaluated by gpt-4o ([2025-02-12T03:08:59.889755+00:00']) Grade: A

Notes: The submitted solution correctly follows the method of separation of variables and integrates both sides of the differential equation. The integration process and the application of the initial condition are handled correctly. The final expression for  $N(t)$  matches the standard solution, showing that the solution is physically equivalent. The Python code provided also correctly implements the mathematical expression. Overall, the solution demonstrates a clear understanding of the problem and the correct application of mathematical techniques.

Attempt 4 (2025-02-11T08:50:44.089119+00:00)

**Detailed Solution:** Step 1: Understand the Problem

We are given a differential equation that describes the rate of change of the number of dark matter particles  $N(t)$  in an astrophysical body:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ . Our goal is to find  $N(t)$  as a function of time.

Step 2: Devise a Plan

This is a first-order nonlinear ordinary differential equation (ODE). To solve it, we can use the method of separation of variables. The equation is separable, meaning we can rearrange it to isolate  $N$  and  $t$  on opposite sides of the equation.

Step 3: Carry Out the Plan

Step 3.1: Separate Variables

Start with the given differential equation:

$$\frac{dN}{dt} = C - C_A N^2$$

Separate the variables  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Integrate both sides of the equation:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side (LHS) integral can be evaluated using a standard integral formula. The integral of  $\frac{1}{a-bx^2}$  is:

$$\int \frac{dx}{a-bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \frac{\sqrt{bx}}{\sqrt{a}} \right) + \text{constant}$$

Applying this to our integral:

$$\int \frac{dN}{C-C_A N^2} = \frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \frac{\sqrt{C_A} N}{\sqrt{C}} \right) + \text{constant}$$

The right-hand side (RHS) integral is straightforward:

$$\int dt = t + \text{constant}$$

Step 3.3: Combine the Results

Combining the results of the integrals:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \frac{\sqrt{C_A} N}{\sqrt{C}} \right) = t + \text{constant}$$

Step 3.4: Apply the Initial Condition

Apply the initial condition  $N(0) = 0$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \frac{\sqrt{C_A} \cdot 0}{\sqrt{C}} \right) = 0 + \text{constant}$$

Since  $\tanh^{-1}(0) = 0$ , the constant is zero. Thus, we have:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \frac{\sqrt{C_A} N}{\sqrt{C}} \right) = t$$

Step 3.5: Solve for  $N(t)$

Multiply both sides by  $\sqrt{CC_A}$ :

$$\tanh^{-1} \left( \frac{\sqrt{C_A} N}{\sqrt{C}} \right) = \sqrt{CC_A} t$$

Take the hyperbolic tangent of both sides:

$$\frac{\sqrt{C_A} N}{\sqrt{C}} = \tanh(\sqrt{CC_A} t)$$

Solve for  $N(t)$ :

$$N(t) = \frac{\sqrt{C}}{\sqrt{C_A}} \tanh(\sqrt{CC_A} t)$$

Final Answer:

$$N(t) = \frac{\sqrt{C}}{\sqrt{C_A}} \tanh(\sqrt{CC_A} t)$$

Python Implementation:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return (math.sqrt(C) / math.sqrt(C_A)) * math.tanh(math.sqrt(C * C_A) * t)
```

This Python function `answer` takes the parameters  $C$ ,  $C_A$ , and  $t$  and returns the value of  $N(t)$  as derived above. The function uses the `math` module to compute the square root and hyperbolic tangent.

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.937805+00:00\*]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:06.226694+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The standard solution uses a different form involving exponential functions, while the submitted solution uses hyperbolic tangent functions. These forms are equivalent due to the identity  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The solution process is clear, includes necessary steps, and applies the initial condition correctly. The Python implementation correctly reflects the derived solution.

**Attempt 5** (2025-02-11T08:51:19.121715+00:00)

**Detailed Solution:** Mathematical Solution

We are given the Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2,$$

with the initial condition  $N(0) = 0$ . This is a first-order nonlinear ordinary differential equation (ODE). To solve it, we will use the method of separation of variables.

Step 1: Rewrite the ODE

First, rewrite the ODE in a form suitable for separation of variables:

$$\frac{dN}{dt} = C - C_A N^2.$$

Step 2: Separate Variables

Separate the variables  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 3: Integrate Both Sides

Integrate both sides of the equation:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

Step 4: Evaluate the Integral

The left-hand side integral can be evaluated using partial fractions. First, rewrite the integrand:

$$\frac{1}{C - C_A N^2} = \frac{1}{C} \cdot \frac{1}{1 - \frac{C_A}{C} N^2}.$$

Let  $k = \sqrt{\frac{C_A}{C}}$ , then:

$$\frac{1}{C - C_A N^2} = \frac{1}{C} \cdot \frac{1}{1 - k^2 N^2}.$$

The integral becomes:

$$\frac{1}{C} \int \frac{dN}{1 - k^2 N^2} = t + \text{constant}.$$

The integral of  $\frac{1}{1 - k^2 N^2}$  is:

$$\int \frac{dN}{1 - k^2 N^2} = \frac{1}{k} \tanh^{-1}(kN).$$

Thus, we have:

$$\frac{1}{C} \cdot \frac{1}{k} \tanh^{-1}(kN) = t + \text{constant}.$$

Step 5: Apply the Initial Condition

Apply the initial condition  $N(0) = 0$ :

$$\frac{1}{Ck} \tanh^{-1}(0) = 0 + \text{constant} \implies \text{constant} = 0.$$

Thus, the equation simplifies to:

$$\frac{1}{Ck} \tanh^{-1}(kN) = t.$$

Step 6: Solve for  $N(t)$

Solve for  $N(t)$ :

$$\tanh^{-1}(kN) = Ckt \implies kN = \tanh(Ckt).$$

Finally, substitute back  $k = \sqrt{\frac{C}{C_A}}$ :

$$N(t) = \frac{1}{k} \tanh(Ckt) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Python Implementation

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C * C_A)
    return math.sqrt(C / C_A) * math.tanh(k * t)
```

This Python function `answer` takes the parameters  $C$ ,  $C_A$ , and  $t$  and returns the value of  $N(t)$  as derived from the mathematical solution. The function uses the `math` module to compute the square root and hyperbolic tangent.

**Verification Results:**

**Auto verification result** ([ '2025-02-18T05:49:51.938209+00:00' ]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([ '2025-02-12T03:09:12.716468+00:00' ]) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The final expression for  $N(t)$ ,  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$ , is mathematically equivalent to the

standard solution's expression,  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C_A}t} - 1)}{(e^{2\sqrt{C_A}t} + 1)}$ , since  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The approach and steps are clear and correct, demonstrating a solid understanding of the mathematical concepts involved.

## 2.2.4 Model: deepseek-ai/DeepSeek-R1

Attempt 1 (2025-02-11T09:58:38.041400+00:00)

### Detailed Solution:

To solve the given Boltzmann equation for  $N(t)$ , we start with the differential equation:

$$\dot{N} = C - C_A N^2$$

This is a separable differential equation. Rearranging and integrating both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

Using substitution and partial fractions, the integral simplifies to:

$$\frac{1}{2\sqrt{CC_A}} \ln \left| \frac{\sqrt{C/C_A} + N}{\sqrt{C/C_A} - N} \right| = t$$

Applying the initial condition  $N(0) = 0$ , we solve for the constant of integration and exponentiate both sides to isolate  $N$ . After algebraic manipulation, we find:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

This solution satisfies the differential equation and initial condition.

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_over_CA = math.sqrt(C / C_A)
    sqrt_C_CA = math.sqrt(C * C_A)
    return sqrt_C_over_CA * math.tanh(sqrt_C_CA * t)
```

### Final Answer:

The number of dark matter particles as a function of time is given by:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_over_CA = math.sqrt(C / C_A)
    sqrt_C_CA = math.sqrt(C * C_A)
    return sqrt_C_over_CA * math.tanh(sqrt_C_CA * t)
```

### Verification Results:

Auto verification result ([2025-02-18T05:49:51.938649+00:00]) Status: **Correct**

### Evaluations:

Evaluated by gpt-4o ([2025-02-12T03:09:18.734584+00:00]) Grade: A

Notes: The submitted solution correctly identifies the differential equation as separable and performs the integration accurately. The solution process includes the correct application of partial fractions and integration techniques, leading to the correct expression for  $N(t)$ . The final expression  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$  is mathematically equivalent to the standard solution. The Python code implementation is also correct and consistent with the derived formula. The solution demonstrates a clear understanding of the problem and the necessary mathematical concepts.

Attempt 2 (2025-02-11T09:59:23.720940+00:00)

**Detailed Solution:**

To solve the given differential equation for  $N(t)$ , we start by recognizing it as a separable equation. Through integration and applying the initial condition  $N(0) = 0$ , we derive the solution using hyperbolic tangent function. The final expression for  $N(t)$  is:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

This solution satisfies the initial condition and approaches the equilibrium value  $\sqrt{C/C_A}$  as  $t \rightarrow \infty$ .

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_over_CA = math.sqrt(C / C_A)
    sqrt_C_times_CA = math.sqrt(C * C_A)
    return sqrt_C_over_CA * math.tanh(sqrt_C_times_CA * t)
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.939086+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:24.481093+00:00']) Grade: A

Notes: The submitted solution correctly identifies the differential equation as separable and proceeds with the correct integration technique. The solution derives the expression for  $N(t)$  using hyperbolic tangent functions, which is equivalent to the standard solution. The initial condition  $N(0) = 0$  is applied correctly, and the solution approaches the correct equilibrium value as  $t \rightarrow \infty$ . The Python code provided accurately implements the derived formula. Overall, the solution is physically equivalent to the standard answer and demonstrates a correct reasoning approach.

**Attempt 3** (2025-02-11T10:00:14.782299+00:00)**Detailed Solution:**

To solve the given differential equation  $\dot{N} = C - C_A N^2$  with the initial condition  $N(0) = 0$ , we first recognize it as a separable equation. By separating variables and integrating, we derive the solution:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

This solution satisfies both the differential equation and the initial condition. The hyperbolic tangent function ensures that  $N(t)$  asymptotically approaches the equilibrium value  $\sqrt{C/C_A}$  as  $t \rightarrow \infty$ .

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_over_CA = math.sqrt(C / C_A)
    sqrt_C_times_CA = math.sqrt(C * C_A)
    return sqrt_C_over_CA * math.tanh(sqrt_C_times_CA * t)
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.939521+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:31.451087+00:00']) Grade: A

Notes: The submitted solution correctly identifies the differential equation as separable and proceeds to separate variables and integrate both sides. The solution process is detailed and correctly applies the integration

technique to arrive at the expression for  $N(t)$ . The final expression  $N(t) = \sqrt{C/C_A} \tanh(\sqrt{CC_A}t)$  is mathematically equivalent to the standard solution. The solution also verifies the correctness by checking the initial condition and the behavior as  $t \rightarrow \infty$ . The Python code provided correctly implements the derived expression. Overall, the solution demonstrates a clear understanding of the problem and the mathematical concepts involved.

**Attempt 4** (2025-02-11T10:01:07.289328+00:00)

**Detailed Solution:**

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.939942+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:38.966515+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation  $\dot{N} = C - C_A N^2$  using separation of variables and integration. The solution process is detailed and correctly applies the initial condition  $N(0) = 0$  to determine the integration constant. The final expression  $N(t) = \sqrt{\frac{C}{C_A}} \cdot \tanh(\sqrt{C \cdot C_A} \cdot t)$  is mathematically equivalent to the standard solution. The solution also includes a Python function that accurately computes  $N(t)$  given  $C$ ,  $C_A$ , and  $t$ . The dimensional analysis confirms the correctness of the solution, ensuring that the argument of the tanh function is dimensionless. Overall, the solution demonstrates a strong understanding of the mathematical concepts involved.

**Attempt 5** (2025-02-11T10:01:53.174678+00:00)

**Detailed Solution:**

The given differential equation is a separable first-order ODE. By separating variables and integrating, we find the solution using hyperbolic tangent function due to the form of the integral. Applying the initial condition and simplifying yields the final expression for  $N(t)$ .

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$$

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.940365+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:45.397777+00:00']) Grade: A

Notes: The submitted solution correctly identifies the differential equation as separable and integrates it to find the solution using the hyperbolic tangent function. The solution process is detailed and logically consistent, leading to the correct final expression for  $N(t)$ . The Python code provided accurately implements



the mathematical solution. The final answer  $N(t) = \sqrt{\frac{c}{C_A}} \tanh(\sqrt{CC_A} t)$  is mathematically equivalent to the standard solution, demonstrating a correct understanding of the problem.

## 2.2.5 Model: Qwen/QwQ-32B-Preview

Attempt 1 (2025-02-11T21:08:59.783424+00:00)

**Detailed Solution:** Alright, let's tackle this problem step by step. We have a differential equation that describes the number of dark matter particles,  $N(t)$ , in an astrophysical body over time. The equation is:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ . Our goal is to find  $N(t)$  as a function of time.

Step 1: Understand the Problem

First, I need to understand what this equation is telling us. The rate of change of  $N$  with respect to time,  $\dot{N}$ , is equal to the capture rate  $C$  minus the annihilation rate  $C_A N^2$ . This makes sense because particles are being captured at a constant rate  $C$ , but they are also annihilating at a rate proportional to  $N^2$ , since annihilation typically involves pairs of particles.

Given that  $N(0) = 0$ , meaning there are no dark matter particles initially, we need to find how  $N$  evolves over time.

Step 2: Devise a Plan

This is a first-order nonlinear ordinary differential equation (ODE). Specifically, it's a separable equation because we can write it in the form:

$$\frac{dN}{dt} = C - C_A N^2$$

To solve this, I can separate the variables  $N$  and  $t$  and integrate both sides.

Step 3: Carry Out the Plan

Let's rearrange the equation to separate  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt$$

Now, integrate both sides. The left side is with respect to  $N$ , and the right side is with respect to  $t$ :

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The integral on the right is simply  $t + \text{constant}$ . The integral on the left is a bit more complicated. To solve it, I can use partial fraction decomposition.

First, factor the denominator:

$$C - C_A N^2 = C \left(1 - \frac{C_A}{C} N^2\right) = C \left(1 - \left(\sqrt{\frac{C_A}{C}} N\right)^2\right)$$

Lets define  $k = \sqrt{\frac{C_A}{C}}$ , so the denominator becomes:

$$C(1 - k^2 N^2)$$

Now, the integral becomes:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \int dt$$

I can factor out  $C$  from the denominator:

$$\frac{1}{C} \int \frac{dN}{1 - k^2 N^2} = t + \text{constant}$$

The integral  $\int \frac{dN}{1 - k^2 N^2}$  is a standard integral that equals  $\frac{1}{2k} \ln \left| \frac{1+kN}{1-kN} \right| + \text{constant}$ . So, plugging this back in:

$$\frac{1}{C} \cdot \frac{1}{2k} \ln \left| \frac{1+kN}{1-kN} \right| = t + \text{constant}$$

Simplify by combining constants:

$$\frac{1}{2Ck} \ln \left| \frac{1+kN}{1-kN} \right| = t + \text{constant}$$

Now, solve for  $N$ . First, isolate the logarithm:

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2Ck(t + \text{constant})$$

Exponentiate both sides to eliminate the logarithm:

$$\left| \frac{1+kN}{1-kN} \right| = e^{2Ck(t+\text{constant})}$$

Lets absorb the absolute value into the constant by writing:

$$\frac{1+kN}{1-kN} = Ae^{2Ckt}$$

where  $A = \pm e^{2Ck \cdot \text{constant}}$ .

Now, solve for  $N$ :

$$1+kN = Ae^{2Ckt}(1-kN)$$

$$1+kN = Ae^{2Ckt} - AkNe^{2Ckt}$$

Bring all terms involving  $N$  to one side:

$$kN + AkNe^{2Ckt} = Ae^{2Ckt} - 1$$

$$N(k + Ake^{2Ckt}) = Ae^{2Ckt} - 1$$

$$N = \frac{Ae^{2Ckt} - 1}{k(Ae^{2Ckt} + 1)}$$

Recall that  $k = \sqrt{\frac{CA}{C}}$ , so substitute back:

$$N = \frac{Ae^{2C\sqrt{\frac{CA}{C}}t} - 1}{\sqrt{\frac{CA}{C}}(Ae^{2C\sqrt{\frac{CA}{C}}t} + 1)}$$

Now, apply the initial condition  $N(0) = 0$ :

$$0 = \frac{Ae^0 - 1}{\sqrt{\frac{CA}{C}}(Ae^0 + 1)} = \frac{A - 1}{\sqrt{\frac{CA}{C}}(A + 1)}$$

For this to be zero, the numerator must be zero:

$$A - 1 = 0 \implies A = 1$$

So, the solution simplifies to:

$$N = \frac{e^{2C\sqrt{\frac{CA}{C}}t} - 1}{\sqrt{\frac{CA}{C}}(e^{2C\sqrt{\frac{CA}{C}}t} + 1)}$$

This looks a bit messy. Lets define  $\alpha = \sqrt{\frac{CA}{C}}$ , so:

$$N = \frac{e^{2C\alpha t} - 1}{\alpha(e^{2C\alpha t} + 1)}$$

Alternatively, we can use the hyperbolic tangent function,  $\tanh$ , which is defined as:

$$\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1}$$

So, our expression for  $N$  becomes:

$$N = \frac{1}{\alpha} \tanh(C\alpha t)$$

Recall that  $\alpha = \sqrt{\frac{C_A}{C}}$ , so:

$$N = \frac{1}{\sqrt{\frac{C_A}{C}}} \tanh\left(C\sqrt{\frac{C_A}{C}}t\right) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$$

This is the expression for  $N(t)$ .

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$$

Python Code

To implement this solution in Python, we can use the math module for the hyperbolic tangent function. Here's how you can write the function:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_C_A = math.sqrt(C * C_A)
    tanh_term = math.tanh(t * sqrt_C_C_A)
    N_t = math.sqrt(C / C_A) * tanh_term
    return N_t
```

Explanation

1. **Mathematical Solution:** - We started with the differential equation  $\dot{N} = C - C_A N^2$ . - Separated variables and integrated both sides. - Used partial fractions and standard integrals to solve for  $N(t)$ . - Applied the initial condition  $N(0) = 0$  to find the constant of integration. - Simplified the expression using hyperbolic functions for clarity.

2. **Python Implementation:** - Defined a function answer that takes  $C$ ,  $C_A$ , and  $t$  as inputs. - Computed the square root of the product  $C \times C_A$ . - Used the math.tanh function to compute the hyperbolic tangent. - Calculated  $N(t)$  using the derived formula and returned the result.

This approach ensures that the problem is solved both mathematically and computationally, meeting the requirements specified.

**Verification Results:**

**Auto verification result** ([ '2025-02-18T05:49:51.943116+00:00' ]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([ '2025-02-12T03:10:27.047755+00:00' ]) Grade: A

Notes: The submitted solution correctly solves the differential equation  $\dot{N} = C - C_A N^2$  using separation of variables and integration. The solution process includes necessary steps such as partial fraction decomposition and integration, leading to the expression  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$ . This is mathematically equivalent to

the standard solution  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$  as  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The approach is valid, and the use of hyperbolic functions simplifies the expression. The Python implementation is a correct computational representation of the derived formula.

**Attempt 2** (2025-02-11T21:10:07.543076+00:00)

**Detailed Solution:** To solve this problem, we need to find the function  $N(t)$  that describes the number of dark matter particles in an astrophysical body over time, given the differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understand the Problem

We have a first-order nonlinear ordinary differential equation (ODE):

$$\frac{dN}{dt} = C - C_A N^2$$

where  $C$  and  $C_A$  are positive constants representing the capture rate and the annihilation rate per effective volume, respectively.

Our goal is to solve this ODE with the initial condition  $N(0) = 0$  to find  $N(t)$ .

Step 2: Devise a Plan

This is a separable differential equation, which means we can rearrange it so that all terms involving  $N$  are on one side and all terms involving  $t$  are on the other side. We can then integrate both sides to find the solution. The general approach for solving separable ODEs is:

1. Separate the variables  $N$  and  $t$ . 2. Integrate both sides. 3. Apply the initial condition to solve for any constants of integration.

Step 3: Carry Out the Plan

Step 3.1: Separate Variables

Starting with the ODE:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Now, integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side is an integral with respect to  $N$ , and the right-hand side is an integral with respect to  $t$ . First, let's focus on the left-hand side integral:

$$\int \frac{dN}{C - C_A N^2}$$

To make this integral easier, we can factor out  $C$  from the denominator:

$$\int \frac{dN}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C} \int \frac{dN}{1 - \left(\sqrt{\frac{C_A}{C}} N\right)^2}$$

Lets make a substitution to simplify the integral. Let:

$$u = \sqrt{\frac{C_A}{C}} N$$

Then:

$$du = \sqrt{\frac{C_A}{C}} dN \implies dN = \sqrt{\frac{C}{C_A}} du$$

Substituting  $u$  and  $dN$  into the integral:

$$\frac{1}{C} \int \frac{\sqrt{\frac{C}{C_A}} du}{1 - u^2} = \frac{1}{\sqrt{CC_A}} \int \frac{du}{1 - u^2}$$

The integral  $\int \frac{du}{1 - u^2}$  is a standard integral that equals  $\tanh^{-1}(u)$ , the inverse hyperbolic tangent function. So:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(u) + C_1 = t + C_2$$

Combining constants  $C_1$  and  $C_2$  into a single constant  $C_3$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(u) = t + C_3$$

Step 3.3: Solve for  $N(t)$

Recall that  $u = \sqrt{\frac{C_A}{C}}N$ , so:

$$\tanh^{-1}\left(\sqrt{\frac{C_A}{C}}N\right) = \sqrt{CC_A}(t + C_3)$$

Taking the hyperbolic tangent of both sides:

$$\sqrt{\frac{C_A}{C}}N = \tanh(\sqrt{CC_A}(t + C_3))$$

Therefore:

$$N = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}(t + C_3))$$

Step 3.4: Apply the Initial Condition

We are given that  $N(0) = 0$ . Let's apply this condition to find  $C_3$ :

$$0 = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}(0 + C_3))$$

Since  $\sqrt{\frac{C}{C_A}}$  is nonzero, we must have:

$$\tanh(\sqrt{CC_A}C_3) = 0$$

The hyperbolic tangent function  $\tanh(x)$  is zero only when  $x = 0$ . Therefore:

$$\sqrt{CC_A}C_3 = 0 \implies C_3 = 0$$

So the solution simplifies to:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$$

Python Code

To implement this solution in Python, we can use the math module for the hyperbolic tangent function. Here's how you can write the function:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_term = math.sqrt(C * C_A)
    tanh_term = math.tanh(sqrt_term * t)
    N_t = math.sqrt(C / C_A) * tanh_term
    return N_t
```

Explanation

1. **Mathematical Solution:** - We started with the given differential equation and separated variables. - Integrated both sides and applied the initial condition to find the constant of integration. - Arrived at the final expression for  $N(t)$  using the properties of hyperbolic functions.

2. **Python Implementation:** - The function answer takes three parameters:  $C$ ,  $C_A$ , and  $t$ . - It calculates the square root of the product  $C \times C_A$ . - Computes the hyperbolic tangent of the product of this square root and time  $t$ . - Multiplies this by the square root of  $\frac{C}{C_A}$  to get  $N(t)$ . - Returns the computed value of  $N(t)$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.943541+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:10:32.643813+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The use of hyperbolic functions in the submitted solution is a valid alternative to the logarithmic form in the standard solution. Both solutions describe the same physical behavior of the system. The integration process, application of initial conditions, and final expression for  $N(t)$  are all correctly handled. The Python code provided is a correct implementation of the mathematical solution.

**Attempt 3** (2025-02-11T21:11:14.058337+00:00)

**Detailed Solution:** Alright, let's tackle this problem step by step. We have a differential equation that describes the number of dark matter particles,  $N(t)$ , in an astrophysical body over time. The equation is:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ . Our goal is to find  $N(t)$  as a function of time.

Step 1: Understand the Problem

First, I need to understand what this equation is telling us. The rate of change of  $N$  with respect to time,  $\dot{N}$ , is equal to the capture rate  $C$  minus the annihilation rate  $C_A N^2$ . This makes sense because particles are being captured at a rate  $C$ , but they are also annihilating in pairs at a rate proportional to  $N^2$ , since two particles are needed for annihilation.

Given that  $N(0) = 0$ , meaning there are no dark matter particles initially, we need to find how  $N$  evolves over time.

Step 2: Devise a Plan

This is a first-order nonlinear ordinary differential equation (ODE). Specifically, it's a separable equation because we can write it in the form:

$$\frac{dN}{dt} = C - C_A N^2$$

To solve this, I can separate the variables  $N$  and  $t$  and integrate both sides.

Step 3: Carry Out the Plan

Let's rearrange the equation to separate  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt$$

Now, integrate both sides. The left side is with respect to  $N$ , and the right side is with respect to  $t$ :

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

Let's focus on the left integral. To make it easier, factor  $C$  out of the denominator:

$$\int \frac{dN}{C(1 - \frac{C_A}{C} N^2)} = \int dt$$

Simplify:

$$\frac{1}{C} \int \frac{dN}{1 - \left(\sqrt{\frac{C_A}{C}} N\right)^2} = \int dt$$

Now, let's make a substitution to simplify the integral. Let:

$$u = \sqrt{\frac{C_A}{C}} N$$

Then:

$$dN = \sqrt{\frac{C}{C_A}} du$$

Substitute back into the integral:

$$\frac{1}{C} \int \frac{\sqrt{\frac{C}{C_A}} du}{1-u^2} = \int dt$$

Simplify the constants:

$$\frac{1}{\sqrt{CC_A}} \int \frac{du}{1-u^2} = \int dt$$

The integral on the left is a standard integral that can be expressed in terms of the inverse hyperbolic tangent function,  $\tanh^{-1}(u)$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(u) = t + K$$

where  $K$  is the constant of integration.

Now, substitute back for  $u$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = t + K$$

To find the constant  $K$ , use the initial condition  $N(0) = 0$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} \cdot 0\right) = 0 + K \implies K = 0$$

So the equation simplifies to:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = t$$

Now, solve for  $N$ :

$$\tanh^{-1}\left(\sqrt{\frac{C_A}{C}} N\right) = t\sqrt{CC_A}$$

Take the hyperbolic tangent of both sides:

$$\sqrt{\frac{C_A}{C}} N = \tanh(t\sqrt{CC_A})$$

Finally, solve for  $N$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$$

Now, let's implement this in Python as required.

Final Solution To solve this problem, we need to find the number of dark matter particles  $N(t)$  in an astrophysical body as a function of time, given the capture rate  $C$  and the annihilation rate per effective volume  $C_A$ . The governing equation is:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .



Approach

1. **Understand the Problem:** - The rate of change of  $N$  is due to capture minus annihilation. - Annihilation is proportional to  $N^2$  because it involves pairs of particles.

2. **Mathematical Solution:** - Solve the separable differential equation. - Integrate both sides after separation. - Apply the initial condition to find the constant of integration. - Solve for  $N(t)$ .

3. **Python Implementation:** - Translate the mathematical solution into a Python function. - Use the math module for the hyperbolic tangent function.

Solution Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_CC_A = math.sqrt(C * C_A)
    tanh_term = math.tanh(t * sqrt_CC_A)
    N_t = math.sqrt(C / C_A) * tanh_term
    return N_t
```

Explanation

1. **Mathematical Solution:** - We separated variables and integrated both sides. - Used the inverse hyperbolic tangent function and applied the initial condition to solve for the constant. - Expressed  $N(t)$  in terms of the hyperbolic tangent function.

2. **Python Implementation:** - Calculated the square root of the product  $C \times C_A$ . - Computed the hyperbolic tangent of  $t$  times the square root. - Multiplied by the square root of  $\frac{C}{C_A}$  to get  $N(t)$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.943962+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:10:40.674395+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation  $\dot{N} = C - C_A N^2$  using separation of variables and integration. The solution process is detailed and includes all necessary steps, such as variable substitution and integration. The final expression  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(t\sqrt{CC_A})$  is mathematically equivalent to the standard solution's expression for  $N(t)$ . Both solutions describe the same physical behavior of the system, reaching the steady state as  $t \rightarrow \infty$ . The use of the hyperbolic tangent function is a valid alternative to the exponential form in the standard solution, demonstrating a different but correct approach.

**Attempt 4** (2025-02-11T21:12:59.375607+00:00)

**Detailed Solution:** To solve this problem, we need to find the function  $N(t)$  that describes the number of dark matter particles in an astrophysical body over time, given the differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understand the Problem

We have a first-order nonlinear ordinary differential equation (ODE):

$$\frac{dN}{dt} = C - C_A N^2$$

where  $C$  and  $C_A$  are positive constants representing the capture rate and the annihilation rate per effective volume, respectively.

Our goal is to solve this ODE with the initial condition  $N(0) = 0$  to find  $N(t)$ .

Step 2: Devise a Plan

This is a separable differential equation, meaning we can rearrange it so that all terms involving  $N$  are on one side and all terms involving  $t$  are on the other side. We can then integrate both sides to find the solution.

The general approach for solving separable ODEs is:

1. Separate the variables  $N$  and  $t$ . 2. Integrate both sides. 3. Apply the initial condition to solve for any constants of integration. 4. Solve for  $N(t)$ .

Step 3: Carry Out the Plan

Step 3.1: Separate Variables

Starting with the ODE:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

We need to integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

First, let's simplify the integral on the left-hand side. Factor out  $C$  from the denominator:

$$\int \frac{dN}{C(1 - \frac{C_A}{C} N^2)} = \int dt$$

Let  $k^2 = \frac{C_A}{C}$ , assuming  $C$  and  $C_A$  are positive, so  $k$  is real and positive. Then the equation becomes:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \int dt$$

This integral can be solved using partial fractions. Notice that:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{(1 - kN)(1 + kN)}$$

Using partial fractions, we can write:

$$\frac{1}{(1 - kN)(1 + kN)} = \frac{A}{1 - kN} + \frac{B}{1 + kN}$$

Solving for  $A$  and  $B$ :

$$1 = A(1 + kN) + B(1 - kN)$$

Set  $N = \frac{1}{k}$ :

$$1 = A(1 + k \cdot \frac{1}{k}) + B(1 - k \cdot \frac{1}{k}) = A(2) + B(0) \implies A = \frac{1}{2}$$

Set  $N = -\frac{1}{k}$ :

$$1 = A(1 + k \cdot (-\frac{1}{k})) + B(1 - k \cdot (-\frac{1}{k})) = A(0) + B(2) \implies B = \frac{1}{2}$$

So:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{2} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right)$$

Thus, the integral becomes:

$$\int \frac{dN}{C} \cdot \frac{1}{2} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) = \int dt$$

$$\frac{1}{2C} \int \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) dN = \int dt$$

Integrate both terms:

$$\frac{1}{2C} \left( -\frac{1}{k} \ln|1 - kN| + \frac{1}{k} \ln|1 + kN| \right) = t + C_1$$

Simplify:

$$\frac{1}{2Ck} (\ln|1+kN| - \ln|1-kN|) = t + C_1$$

$$\frac{1}{2Ck} \ln \left| \frac{1+kN}{1-kN} \right| = t + C_1$$

Step 3.3: Apply Initial Condition

Now, apply the initial condition  $N(0) = 0$ :

$$\frac{1}{2Ck} \ln \left| \frac{1+k \cdot 0}{1-k \cdot 0} \right| = 0 + C_1$$

$$\frac{1}{2Ck} \ln 1 = C_1 \implies C_1 = 0$$

So the equation simplifies to:

$$\frac{1}{2Ck} \ln \left| \frac{1+kN}{1-kN} \right| = t$$

Step 3.4: Solve for  $N(t)$

First, solve for the logarithmic expression:

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2Ckt$$

Exponentiate both sides:

$$\left| \frac{1+kN}{1-kN} \right| = e^{2Ckt}$$

Since  $N$  starts at 0 and increases, and  $k$  is positive, we can drop the absolute value:

$$\frac{1+kN}{1-kN} = e^{2Ckt}$$

Now, solve for  $N$ :

$$1+kN = e^{2Ckt} (1-kN)$$

$$1+kN = e^{2Ckt} - kNe^{2Ckt}$$

$$kN + kNe^{2Ckt} = e^{2Ckt} - 1$$

$$kN(1 + e^{2Ckt}) = e^{2Ckt} - 1$$

$$N = \frac{e^{2Ckt} - 1}{k(1 + e^{2Ckt})}$$

Recall that  $k^2 = \frac{C_A}{C}$ , so  $k = \sqrt{\frac{C_A}{C}}$ . Substitute back:

$$N(t) = \frac{e^{2C\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2C\sqrt{\frac{C_A}{C}}t})}$$

Simplify the exponent:

$$2C\sqrt{\frac{C_A}{C}} = 2\sqrt{CC_A}$$

So:

$$N(t) = \frac{e^{2\sqrt{CC_A}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{CC_A}t})}$$

To simplify further, multiply numerator and denominator by  $\sqrt{C}$ :

$$N(t) = \frac{\sqrt{C}(e^{2\sqrt{CC_A}t} - 1)}{\sqrt{C_A}(1 + e^{2\sqrt{CC_A}t})}$$

This can be written as:

$$N(t) = \frac{\sqrt{C}}{\sqrt{C_A}} \cdot \frac{e^{2\sqrt{CC_A}t} - 1}{1 + e^{2\sqrt{CC_A}t}}$$

Recognize that:

$$\frac{e^x - 1}{e^x + 1} = \tanh\left(\frac{x}{2}\right)$$

So:

$$N(t) = \frac{\sqrt{C}}{\sqrt{C_A}} \cdot \tanh(\sqrt{CC_A}t)$$

Thus, the final expression for  $N(t)$  is:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$$

Python Code

To implement this solution in Python, we need to translate the mathematical expression into code. We'll use the math module for the square root and hyperbolic tangent functions.

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    sqrt_C_C_A = math.sqrt(C * C_A)
    tanh_term = math.tanh(sqrt_C_C_A * t)
    N_t = math.sqrt(C / C_A) * tanh_term
    return N_t
```

Explanation

1. **Mathematical Solution:** - We started with the given differential equation and initial condition. - We separated variables and integrated both sides. - Applied the initial condition to solve for the constant of integration. - Solved for  $N(t)$  and simplified the expression using properties of logarithms and exponents. - Further simplified the expression using the hyperbolic tangent function for clarity.

2. **Python Implementation:** - We defined a function answer that takes  $C$ ,  $C_A$ , and  $t$  as inputs. - Computed the square root of the product  $C \times C_A$ . - Calculated the hyperbolic tangent of  $\sqrt{CC_A} \times t$ . - Multiplied by the square root of  $\frac{C}{C_A}$  to get  $N(t)$ . - Returned the result.

This approach ensures that the mathematical solution is accurately translated into a functional Python code that can be used to compute  $N(t)$  for given values of  $C$ ,  $C_A$ , and  $t$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.944394+00:00\*]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T03:10:47.558994+00:00\*]) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integrates both sides. The solution process is detailed and includes all necessary steps, such as applying the initial condition and simplifying the expression using hyperbolic functions. The final expression for

$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$  is mathematically equivalent to the standard solution's expression for  $N$ . The use of hyperbolic tangent is a valid simplification of the logarithmic form, and the Python code accurately implements the mathematical solution.

**Attempt 5** (2025-02-11T21:14:41.806639+00:00)

**Detailed Solution:** Alright, I've got this problem to solve. It's about dark matter in astrophysical bodies, which sounds pretty interesting, but I need to focus on the math here. The problem gives me a Boltzmann equation that describes how the number of dark matter particles,  $N$ , changes over time,  $t$ . The equation is:

$$\dot{N} = C - C_A N^2$$

And it says that initially, at time  $t=0$ ,  $N(0)=0$ . I need to find  $N(t)$ , which is how  $N$  changes with time. First, I need to understand what this equation is saying.  $\dot{N}$  is the rate of change of  $N$  with respect to time, so it's a differential equation. It's saying that the rate of change of  $N$  is equal to the capture rate,  $C$ , minus the annihilation rate, which is proportional to  $N$  squared via  $C_A$ .

So, this is a first-order differential equation because it involves the first derivative of  $N$  with respect to  $t$ . It looks like a nonlinear equation because of the  $N$  squared term. I need to solve this differential equation with the initial condition  $N(0)=0$ .

Let me recall how to solve first-order differential equations. For linear equations, there are standard methods like integrating factors, but since this is nonlinear, I might need to use separation of variables or some other technique.

Let me try separation of variables. The general idea is to get all the  $N$  terms on one side and all the  $t$  terms on the other side, then integrate both sides.

So, starting with:

$$\dot{N} = C - C_A N^2$$

I can write as  $dN/dt$ , so:

$$\frac{dN}{dt} = C - C_A N^2$$

Now,  $dN/dt$

$$\frac{dN}{C - C_A N^2} = dt$$

$N(0)=0, N(t)=N(t), t=0$

$$\int_0^N \frac{dN'}{C - C_A N'^2} = \int_0^t dt'$$

$tC - C_A N'^2$

$C - C_A N'^2 C(1 - (C_A/C)N'^2)k^2 = C_A/CC(1 - k^2 N'^2)$

$k^2 = C_A/C$

$$\int_0^N \frac{dN'}{C(1 - k^2 N'^2)} = \int_0^t dt'$$

$1/C$

$$\frac{1}{C} \int_0^N \frac{dN'}{1 - k^2 N'^2} = t$$

$dN' / (1 - k^2 N'^2) = (1/(2k)) \ln |(1 + kN')/(1 - kN')| + constant$

tangent  $dx / (1 - a x) = (1/(2a)) \ln |(1 + ax)/(1 - ax)| + constant$

$$\frac{1}{C} \cdot \frac{1}{2k} \ln \left| \frac{1 + kN}{1 - kN} \right| \Bigg|_0^N = t$$

$$\frac{1}{2Ck} \ln \left| \frac{1+kN}{1-kN} \right| = t$$

$$N=0 \ln(1/1)=0$$

$$N(t) = 2 C k$$

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2Ckt$$

$$\left| \frac{1+kN}{1-kN} \right| = e^{2Ckt}$$

$$\frac{1+kN}{1-kN} = e^{2Ckt}$$

N

$$1+kN = e^{2Ckt}(1-kN)$$

$$1+kN = e^{2Ckt} - kN e^{2Ckt}$$

N

$$kN + kN e^{2Ckt} = e^{2Ckt} - 1$$

factor out N:

$$N(k + k e^{2Ckt}) = e^{2Ckt} - 1$$

$$N = \frac{e^{2Ckt} - 1}{k(1 + e^{2Ckt})}$$

$$k^2 = C_A/Ck = \text{sqrt}(C_A/C)$$

$$N(t) = \frac{e^{2C\sqrt{C_A/C}t} - 1}{\sqrt{C_A/C}(1 + e^{2C\sqrt{C_A/C}t})}$$

$$e^{C\sqrt{C_A/C}t}$$

$$\tanh(x) = (e^{2x} - 1)/(e^{2x} + 1)$$

$$N(t) = \frac{e^{at} - 1}{k(1 + e^{at})}$$

$$a = 2 C k = 2 C \text{sqrt}(C_A/C) = 2\text{sqrt}(C^3/C_A).$$

a

$$a = 2 C k = 2 C \text{sqrt}(C_A/C) = 2\text{sqrt}(C * C_A)$$

$$a = 2 \text{sqrt}(C * C_A)$$

$$N(t) = \frac{e^{at} - 1}{k(1 + e^{at})} = \frac{e^{at} - 1}{\sqrt{C_A/C}(1 + e^{at})}$$

$$\tanh(x/2) = (e^x - 1)/(e^x + 1)$$

$$1 + e^{at} e^{at} - 1$$

$$\tanh(a t / 2)$$

$$\tanh(x) = (e^{2x} - 1)/(e^{2x} + 1) \tanh(at/2) = (e^{at} - 1)/(e^{at} + 1)$$

$N(t) = \tanh(a t / 2) / k$   
 wait, let's see.

$$N(t) = \frac{e^{at} - 1}{k(1 + e^{at})} = \frac{1}{k} \cdot \frac{e^{at} - 1}{e^{at} + 1} = \frac{1}{k} \tanh\left(\frac{at}{2}\right)$$

$\tanh(x) = (e^{2x} - 1)/(e^{2x} + 1)$   
 $\tanh(x/2) = (e^x - 1)/(e^x + 1)$   
 $N(t) = (1/k) \tanh(a t / 2)$   
 $a = 2 \sqrt{C * C_A}$   
 $k = \sqrt{C / C_A}$   
 $N(t) = \sqrt{C / C_A} * \tanh(\sqrt{C * C_A} t)$

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t)$$

$N(0)=0$   
 $t=0 \Rightarrow \tanh(0)=0 \Rightarrow N(0)=0$   
 $tN(t)$   
 $\tanh(x) = \frac{1}{\sqrt{C / C_A}} N(t)$   
 capture rate    annihilation rate.

Python  
 $C, C_A, t \rightarrow N(t)$

```
import numpy as np
def answer(C: float, C_A: float, t: float) -> float:
    a = np.sqrt(C * C_A) * t
    n = np.sqrt(C / C_A) * np.tanh(a)
    return n
numpysqrttanh
CC_A C C_A
```

```
import numpy as np
def answer(C: float, C_A: float, t: float) -> float:
    a = np.sqrt(C * C_A) * t
    n = np.sqrt(C / C_A) * np.tanh(a)
    return n
```

**Final Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t)$$

To implement this solution in Python, use the following code:

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    a = np.sqrt(C * C_A) * t
    n = np.sqrt(C / C_A) * np.tanh(a)
    return n
```

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.944834+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** (['2025-02-12T03:10:55.187374+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and arrives at a solution that is mathematically equivalent to the standard solution. The final expression  $N(t) =$

$\sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$  is equivalent to the standard solution  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \left( \frac{e^{2\sqrt{C}t} - 1}{e^{2\sqrt{C}t} + 1} \right)$  because  $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$ . The

solution process is thorough and includes necessary steps, demonstrating a correct reasoning approach. The Python implementation is also correctly provided.



## 2.2.6 Model: chatgpt-4o-latest

Attempt 1 (2025-02-12T21:44:57.316621+00:00)

**Detailed Solution:** We start with the given differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition:

$$N(0) = 0.$$

Step 1: Separation of Variables Rewriting the equation:

$$\frac{dN}{dt} = C - C_A N^2.$$

Rearrange to separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 2: Integration We use the standard integral:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \sqrt{\frac{b}{a}} x \right).$$

Setting  $a = C$  and  $b = C_A$ , we integrate:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

Using the standard result:

$$\frac{1}{C} \int \frac{dN}{1 - \frac{C_A}{C} N^2} = \int dt.$$

Let  $k = \sqrt{\frac{C_A}{C}}$ , then:

$$\frac{1}{C} \frac{1}{\sqrt{C_A/C}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Simplifying:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Step 3: Solving for  $N(t)$  Using the initial condition  $N(0) = 0$ , we find  $C_1 = 0$ , so:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t.$$

Multiplying by  $\sqrt{CC_A}$ :

$$\tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = \sqrt{CC_A} t.$$

Taking the hyperbolic tangent:

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{CC_A} t).$$

Solving for  $N$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Step 4: Python Implementation Now, we implement this function in Python:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This function takes  $C$ ,  $C_A$ , and  $t$  as inputs and returns  $N(t)$  based on our derived formula.

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.947546+00:00\*]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T22:00:04.247032+00:00\*]) Grade: A

Notes: The submitted solution correctly applies separation of variables and uses the standard integral for the form  $\int \frac{dx}{a-bx^2}$ . The integration process is accurate, and the solution for  $N(t)$  is equivalent to the standard solution, as  $\tanh(x)$  and the exponential form are related through hyperbolic identities. The initial condition is correctly applied to find the integration constant. The Python implementation is a correct translation of the mathematical solution. Overall, the solution is physically equivalent to the standard answer and demonstrates a correct reasoning approach.

**Attempt 2** (2025-02-12T21:45:05.531838+00:00)

**Detailed Solution:** We start with the differential equation governing the number of dark matter particles in the astrophysical body:

$$\dot{N} = C - C_A N^2$$

with the initial condition:

$$N(0) = 0.$$

Step 1: Separation of Variables Rewriting the equation:

$$\frac{dN}{dt} = C - C_A N^2.$$

Rearrange to separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 2: Integration We use the standard integral:

$$\int \frac{dx}{C - C_A N^2}.$$

This is a standard integral of the form:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1}\left(\sqrt{\frac{b}{a}}x\right).$$

Setting  $a = C$  and  $b = C_A$ , we get:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \tanh^{-1}\left(\sqrt{\frac{C_A}{C}}N\right).$$

Thus, integrating both sides:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Step 3: Applying Initial Condition Using  $N(0) = 0$ , we get:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(0) = 0 + C_1.$$

Since  $\tanh^{-1}(0) = 0$ , we find  $C_1 = 0$ , so:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t.$$

Step 4: Solving for  $N(t)$  Taking the hyperbolic tangent on both sides:

$$\tanh(\sqrt{CC_A}t) = \sqrt{\frac{C_A}{C}} N.$$

Solving for  $N$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Step 5: Python Implementation Now, we implement this function in Python:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This function computes  $N(t)$  given the parameters  $C$ ,  $C_A$ , and  $t$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.947948+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T22:00:10.050492+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The use of the hyperbolic tangent function is mathematically equivalent to the exponential form in the standard solution, as  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The solution process includes all necessary steps, applies the initial condition correctly, and arrives at a physically equivalent final answer. The Python implementation is a valid addition, demonstrating practical application of the solution.

**Attempt 3** (2025-02-12T21:45:14.348853+00:00)

**Detailed Solution:** We start with the given differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition:

$$N(0) = 0.$$

Step 1: Separation of Variables Rewriting the equation:

$$\frac{dN}{dt} = C - C_A N^2.$$

Rearrange to separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 2: Integrating Both Sides We use the standard integral:

$$\int \frac{dx}{a - bx^2}.$$

This is a standard integral of the form:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \sqrt{\frac{b}{a}} x \right).$$

Setting  $a = C$  and  $b = C_A$ , we get:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right).$$

Integrating the right-hand side:

$$\int dt = t + C_1.$$

Thus, we obtain:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Step 3: Applying Initial Condition Using  $N(0) = 0$ :

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(0) = 0 + C_1.$$

Since  $\tanh^{-1}(0) = 0$ , it follows that  $C_1 = 0$ , so:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t.$$

Step 4: Solving for  $N(t)$  Multiplying both sides by  $\sqrt{CC_A}$ :

$$\tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = \sqrt{CC_A} t.$$

Taking the hyperbolic tangent on both sides:

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{CC_A} t).$$

Solving for  $N$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

Python Implementation

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This function correctly implements the derived formula for  $N(t)$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.948359+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T22:00:16.257146+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The use of the hyperbolic tangent function is mathematically equivalent to the standard solution's expression involving exponential functions. Both solutions describe the same physical behavior of the system. The initial condition is applied correctly, and the final expression for  $N(t)$  is equivalent to the standard solution. The Python implementation accurately reflects the derived formula.

**Attempt 4** (2025-02-12T21:45:22.252750+00:00)

**Detailed Solution:** We start with the given differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition:

$$N(0) = 0.$$

Step 1: Separation of Variables Rewriting the equation:

$$\frac{dN}{dt} = C - C_A N^2.$$

Rearrange to separate variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 2: Integrating Both Sides We use the standard integral:

$$\int \frac{dx}{a - bx^2}.$$

This is a standard integral of the form:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \sqrt{\frac{b}{a}} x \right).$$

Setting  $a = C$  and  $b = C_A$ , we obtain:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right).$$

Integrating the right-hand side:

$$\int dt = t + C_1.$$

Thus, we obtain:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Step 3: Applying Initial Condition Setting  $N(0) = 0$ , we find:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1}(0) = C_1 \Rightarrow C_1 = 0.$$

Thus, we get:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t.$$

Step 4: Solving for  $N(t)$  Taking the hyperbolic tangent on both sides:

$$\tanh(\sqrt{CC_A}t) = \sqrt{\frac{C_A}{C}}N.$$

Solving for  $N$ :

$$N = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t).$$

Python Implementation

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This function correctly implements the derived formula for  $N(t)$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.948757+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T22:00:23.281187+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The solution method is valid and leads to a physically equivalent result to the standard solution. The final answer,  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$ , is mathematically equivalent to the standard solution's expression for  $N(t)$ , as both describe the same physical behavior of the system. The use of hyperbolic tangent in the submitted solution is an alternative but valid approach to express the solution.

**Attempt 5** (2025-02-12T21:45:30.324991+00:00)

**Detailed Solution:** We begin with the given differential equation:

$$\dot{N} = C - C_A N^2$$

Step 1: Separation of Variables Rearrange the equation to separate  $N$  and  $t$ :

$$\frac{dN}{C - C_A N^2} = dt.$$

This is a separable differential equation.

Step 2: Integrating Both Sides We use the standard integral:

$$\int \frac{dN}{C - C_A N^2}.$$

This integral is of the form:

$$\int \frac{dx}{a - bx^2} = \frac{1}{\sqrt{ab}} \tanh^{-1} \left( \sqrt{\frac{b}{a}} x \right).$$

Setting  $a = C$  and  $b = C_A$ , we obtain:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{C C_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right).$$

Integrating the right-hand side:

$$t + C_1.$$

Thus, we obtain:

$$\frac{1}{\sqrt{C C_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + C_1.$$

Step 3: Solving for  $N(t)$  Solving for  $N$ , we take the hyperbolic tangent:

$$\tanh(\sqrt{C C_A}(t + C_1)) = \sqrt{\frac{C_A}{C}} N.$$

Thus,

$$N = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A}(t + C_1)).$$

Step 4: Applying Initial Condition Given  $N(0) = 0$ , we substitute  $t = 0$ :

$$0 = \sqrt{\frac{C}{C_A}} \tanh(C_1 \sqrt{C C_A}).$$

Since  $\tanh(0) = 0$ , we conclude  $C_1 = 0$ .

Thus, the final solution is:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

**Final Answer:**

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C C_A} t).$$

Python Implementation

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)
```

This function computes  $N(t)$  given  $C$ ,  $C_A$ , and  $t$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.949163+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T22:00:29.395176+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation using separation of variables and integration. The use of the hyperbolic tangent function is mathematically equivalent to the standard solution's logarithmic form, as both represent valid solutions to the differential equation. The initial condition is applied correctly, leading to the correct final expression for  $N(t)$ . The Python implementation also correctly computes the solution. Overall, the solution demonstrates a clear understanding of the problem and provides a valid alternative method to reach the same physical result.

## 2.2.7 Model: meta-llama/Meta-Llama-3.1-70B-Instruct

Attempt 1 (2025-02-11T21:52:10.350929+00:00)

**Detailed Solution:** To solve the given differential equation, we first recognize it as a Riccati equation, which can be solved using various methods. However, given the specific form of this equation and the initial condition, we can attempt a solution by separation of variables or by recognizing it as a Bernoulli differential equation.

The given differential equation is:

$$\dot{N} = C - C_A N^2$$

We can rearrange this equation to make it more amenable to solution by separation of variables:

$$\frac{dN}{dt} = C - C_A N^2$$

To solve this, let's use the method of separation of variables. However, directly integrating this form may be challenging due to its non-linear nature. Instead, we notice that this equation resembles a Bernoulli differential equation, which can be reduced to a linear equation by a suitable substitution.

However, given the quadratic form of the equation and the initial condition  $N(0) = 0$ , we can proceed with a direct approach suitable for this specific case. We rearrange the equation to separate variables:

$$\frac{dN}{C - C_A N^2} = dt$$

To integrate the left side, we use partial fractions or recognize the integral form directly:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

This integral can be solved by recognizing it as a standard form, which results in:

$$\frac{1}{\sqrt{CC_A}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + K$$

where  $K$  is the constant of integration. Given  $N(0) = 0$ , we find  $K = 0$ .

Thus, solving for  $N(t)$ , we get:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t)$$

This is the solution to the differential equation given the initial condition.

Now, converting this solution into Python code as required:

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) at time t,
    given the capture rate C and the dark matter annihilation rate per effective
    volume C_A.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N(t).

    Returns:
    float: The number of dark matter particles N(t) at time t.
    """
    return np.sqrt(C / C_A) * np.tanh(np.sqrt(C * C_A) * t)
```

This Python function implements the solution derived from the differential equation, allowing for the calculation of  $N(t)$  given  $C$ ,  $C_A$ , and  $t$ .

**Verification Results:**



**Auto verification result** ([\*2025-02-18T05:49:51.945270+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T03:11:01.844928+00:00']) Grade: A

Notes: The submitted solution correctly identifies the differential equation as a Riccati equation and proceeds with separation of variables. The integration is performed correctly, leading to the solution  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$ . This solution is mathematically equivalent to the standard solution  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t}-1)}{(e^{2\sqrt{C}t}+1)}$ , as  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ . The approach and reasoning are correct, and the Python implementation accurately reflects the derived solution.

**Attempt 2** (2025-02-11T21:53:15.547333+00:00)

**Detailed Solution:** To solve the given differential equation, we first write it in the standard form:

$$\dot{N} = C - C_A N^2.$$

This is a first-order nonlinear differential equation. To solve it, we can use the method of separation of variables.

First, we separate the variables:

$$\frac{dN}{dt} = C - C_A N^2.$$

We move all terms involving  $N$  to the left-hand side and all terms involving  $t$  to the right-hand side:

$$\frac{dN}{C - C_A N^2} = dt.$$

Now, we integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To evaluate the integral on the left-hand side, we can use the following substitution:

$$u = \sqrt{\frac{C_A}{C}} N.$$

Then,

$$du = \sqrt{\frac{C_A}{C}} dN.$$

Substituting this into the integral, we get:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \int \frac{du}{1 - u^2}.$$

The integral on the right-hand side can be evaluated using the following formula:

$$\int \frac{du}{1 - u^2} = \frac{1}{2} \ln \left| \frac{1 + u}{1 - u} \right|.$$

Substituting back  $u = \sqrt{\frac{C_A}{C}} N$ , we get:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{2\sqrt{CC_A}} \ln \left| \frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} \right|.$$

Evaluating the integral on the right-hand side, we get:

$$\frac{1}{2\sqrt{CC_A}} \ln \left| \frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} \right| = t + K,$$

where  $K$  is the constant of integration.

To find the value of  $K$ , we use the initial condition  $N(0) = 0$ :

$$\frac{1}{2\sqrt{CC_A}} \ln \left| \frac{1 + \sqrt{\frac{C_A}{C}} \cdot 0}{1 - \sqrt{\frac{C_A}{C}} \cdot 0} \right| = 0 + K.$$

Simplifying, we get:

$$K = 0.$$

Therefore, the solution to the differential equation is:

$$\frac{1}{2\sqrt{CC_A}} \ln \left| \frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} \right| = t.$$

To find  $N(t)$ , we can solve this equation for  $N$ :

$$\ln \left| \frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} \right| = 2\sqrt{CC_A} t.$$

Taking the exponential of both sides, we get:

$$\left| \frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} \right| = e^{2\sqrt{CC_A} t}.$$

Since  $N(t) \geq 0$ , we can remove the absolute value sign:

$$\frac{1 + \sqrt{\frac{C_A}{C}} N}{1 - \sqrt{\frac{C_A}{C}} N} = e^{2\sqrt{CC_A} t}.$$

Solving for  $N$ , we get:

$$N(t) = \frac{1}{\sqrt{\frac{C_A}{C}}} \frac{e^{2\sqrt{CC_A} t} - 1}{e^{2\sqrt{CC_A} t} + 1}.$$

Simplifying, we get:

$$N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A} t).$$

Final Answer:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    """
    Calculate the number of dark matter particles in an astrophysical body as a
    function of time.

    Args:
    C (float): The capture rate of dark matter in the astrophysical body.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate the number of dark matter particles.

    Returns:
    float: The number of dark matter particles in the astrophysical body at time t.
    """
```

```

if C_A == 0:
    return C * t
else:
    return math.sqrt(C / C_A) * math.tanh(math.sqrt(C * C_A) * t)

```

**Verification Results:**

**Auto verification result** (['2025-02-18T05:49:51.945798+00:00']) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** (['2025-02-12T03:11:07.590759+00:00']) Grade: A

Notes: The submitted solution correctly applies the method of separation of variables to solve the differential equation. The integration process is accurately carried out, and the substitution used is appropriate. The final expression for  $N(t)$  is equivalent to the standard solution, as  $N(t) = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{CC_A}t)$  is mathematically equivalent to the standard solution's form. The solution process includes all necessary steps and reasoning, and the final answer is correctly implemented in the Python function.

**Attempt 3** (2025-02-11T21:53:53.518318+00:00)

**Detailed Solution:** To solve the given differential equation, we start by separating the variables.

$$\dot{N} = C - C_A N^2$$

We rewrite the equation as:

$$\frac{dN}{dt} = C - C_A N^2$$

Separating the variables, we get:

$$\frac{dN}{C - C_A N^2} = dt$$

Now, we integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

To solve the integral on the left-hand side, we use the following substitution:

$$u = \sqrt{\frac{C_A}{C}} N$$

$$du = \sqrt{\frac{C_A}{C}} dN$$

$$dN = \sqrt{\frac{C}{C_A}} du$$

Substituting these expressions into the integral, we get:

$$\int \frac{\sqrt{\frac{C}{C_A}} du}{1 - u^2} = \int dt$$

$$\frac{1}{\sqrt{C_A C}} \tanh^{-1}(u) = t + K$$

where  $K$  is the constant of integration.

Now, we substitute back  $u = \sqrt{\frac{C_A}{C}} N$ :

$$\frac{1}{\sqrt{C_A C}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + K$$

Using the initial condition  $N(0) = 0$ , we find that  $K = 0$ .

$$\frac{1}{\sqrt{C_A C}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t$$

Now, we solve for  $N$ :

$$\tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = \sqrt{C_A C} t$$

$$\sqrt{\frac{C_A}{C}} N = \tanh(\sqrt{C_A C} t)$$

$$N = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t)$$

Final Answer:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles in an astrophysical
    body
    as a function of time, given the capture rate C and the dark matter annihilation
    rate
    per effective volume C_A.

    Parameters:
    C (float): The capture rate of dark matter in an astrophysical body.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): Time.

    Returns:
    float: The number of dark matter particles in the astrophysical body at time t.
    """
    if C_A == 0:
        return C * t
    else:
        return math.sqrt(C / C_A) * math.tanh(math.sqrt(C_A * C) * t)
```

Verification Results:

Auto verification result (['2025-02-18T05:49:51.946245+00:00']) Status: **Correct**

Evaluations:

Evaluated by gpt-4o (['2025-02-12T03:11:15.288428+00:00']) Grade: A

Notes: The submitted solution correctly solves the differential equation by separating variables and integrating both sides. The substitution  $u = \sqrt{\frac{C_A}{C}} N$  is used to simplify the integral, leading to the expression  $\frac{1}{\sqrt{C_A C}} \tanh^{-1} \left( \sqrt{\frac{C_A}{C}} N \right) = t + K$ . The initial condition  $N(0) = 0$  is applied to find  $K = 0$ . Solving for  $N$ , the solution  $N = \sqrt{\frac{C}{C_A}} \tanh(\sqrt{C_A C} t)$  is obtained, which is mathematically equivalent to the standard solution. The Python function correctly implements this solution, accounting for the special case when  $C_A = 0$ . The approach and final answer are both correct and equivalent to the standard solution.

Attempt 4 (2025-02-11T21:54:34.179900+00:00)

**Detailed Solution:** To solve the given differential equation, we first write it in a standard form:

$$\dot{N} = C - C_A N^2$$

This is a first-order nonlinear differential equation. To solve it, we can use the method of separation of variables.

First, we separate the variables:

$$\frac{dN}{dt} = C - C_A N^2$$
$$\frac{dN}{C - C_A N^2} = dt$$

Now, we integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

To integrate the left-hand side, we can use the following substitution:

$$u = C - C_A N^2$$
$$du = -2C_A N dN$$

However, this substitution does not directly apply. Instead, we can use the following approach:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{\sqrt{CC_A}} \int \frac{dN}{\sqrt{\frac{C}{C_A} - N^2}}$$
$$= \frac{1}{\sqrt{CC_A}} \arcsin\left(\sqrt{\frac{C_A}{C}} N\right)$$

Now, we can write:

$$\frac{1}{\sqrt{CC_A}} \arcsin\left(\sqrt{\frac{C_A}{C}} N\right) = t + K$$

where  $K$  is the constant of integration.

To find  $K$ , we use the initial condition  $N(0) = 0$ :

$$\frac{1}{\sqrt{CC_A}} \arcsin\left(\sqrt{\frac{C_A}{C}} \cdot 0\right) = 0 + K$$
$$K = 0$$

Now, we can write:

$$\frac{1}{\sqrt{CC_A}} \arcsin\left(\sqrt{\frac{C_A}{C}} N\right) = t$$

To find  $N(t)$ , we can rearrange the equation:

$$\arcsin\left(\sqrt{\frac{C_A}{C}} N\right) = \sqrt{CC_A} t$$
$$\sqrt{\frac{C_A}{C}} N = \sin(\sqrt{CC_A} t)$$
$$N(t) = \sqrt{\frac{C}{C_A}} \sin(\sqrt{CC_A} t)$$

Final Answer:

```

import math

def answer(C: float, C_A: float, t: float) -> float:
    """
    Calculate the number of dark matter particles in an astrophysical body as a
    function of time.

    Args:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): Time.

    Returns:
    float: The number of dark matter particles.
    """
    # Calculate the number of dark matter particles
    N = math.sqrt(C / C_A) * math.sin(math.sqrt(C * C_A) * t)
    return N

```

#### Verification Results:

**Auto verification result** ([`'2025-02-18T05:49:51.946676+00:00'`]) Status: **Incorrect**

#### Evaluations:

**Evaluated by gpt-4o** ([`'2025-02-12T03:11:21.168398+00:00'`]) Grade: C

Notes: The submitted solution attempts to solve the differential equation using separation of variables and integration, but it incorrectly applies the integration technique. The standard solution correctly uses partial fraction decomposition to integrate the expression, leading to a logarithmic solution. The submitted solution incorrectly uses an arcsine function, which is not equivalent to the standard solution's logarithmic form. The final expression for  $N(t)$  in the submitted solution is not equivalent to the standard solution's expression. The approach shows some understanding of separation of variables but fails in the integration step and final expression derivation.

#### Attempt 5 (2025-02-11T21:55:25.406207+00:00)

**Detailed Solution:** To solve the given differential equation, we start by separating the variables:

$$\dot{N} = C - C_A N^2$$

We rewrite the equation as:

$$\frac{dN}{dt} = C - C_A N^2$$

Separating the variables, we get:

$$\frac{dN}{C - C_A N^2} = dt$$

Now, we integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

To solve the integral on the left-hand side, we use partial fractions. Let's assume:

$$\frac{1}{C - C_A N^2} = \frac{A}{\sqrt{C/C_A} - N} + \frac{B}{\sqrt{C/C_A} + N}$$

Solving for A and B, we get:

$$A = \frac{1}{2\sqrt{C/C_A}}$$

$$B = \frac{1}{2\sqrt{C/C_A}}$$

Now, we can rewrite the integral as:

$$\int \left( \frac{A}{\sqrt{C/C_A - N}} + \frac{B}{\sqrt{C/C_A + N}} \right) dN = \int dt$$

Solving the integral, we get:

$$-A \ln \left| \sqrt{C/C_A} - N \right| + B \ln \left| \sqrt{C/C_A} + N \right| = t + K$$

where K is the constant of integration.

Using the initial condition  $N(0) = 0$ , we get:

$$-A \ln \left| \sqrt{C/C_A} \right| + B \ln \left| \sqrt{C/C_A} \right| = K$$

Simplifying, we get:

$$K = 0$$

Now, we can rewrite the equation as:

$$-A \ln \left| \sqrt{C/C_A} - N \right| + B \ln \left| \sqrt{C/C_A} + N \right| = t$$

Substituting the values of A and B, we get:

$$-\frac{1}{2\sqrt{C/C_A}} \ln \left| \sqrt{C/C_A} - N \right| + \frac{1}{2\sqrt{C/C_A}} \ln \left| \sqrt{C/C_A} + N \right| = t$$

Simplifying, we get:

$$\ln \left| \frac{\sqrt{C/C_A} + N}{\sqrt{C/C_A} - N} \right| = 2\sqrt{C/C_A} t$$

Taking the exponential of both sides, we get:

$$\frac{\sqrt{C/C_A} + N}{\sqrt{C/C_A} - N} = e^{2\sqrt{C/C_A} t}$$

Solving for N, we get:

$$N = \sqrt{\frac{C}{C_A}} \frac{e^{2\sqrt{C/C_A} t} - 1}{e^{2\sqrt{C/C_A} t} + 1}$$

This is the final symbolic mathematical expression.

Now, let's convert this expression into Python code:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    """
    Calculate the number of dark matter particles in an astrophysical body as a
    function of time.

    Args:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): Time.

    Returns:
    float: The number of dark matter particles.
    """
```

```
# Calculate the square root of C/C_A
sqrt_C_div_C_A = math.sqrt(C / C_A)

# Calculate the exponential term
exp_term = math.exp(2 * sqrt_C_div_C_A * t)

# Calculate the number of dark matter particles
N = sqrt_C_div_C_A * (exp_term - 1) / (exp_term + 1)

return N
```

This code implements the function answer that takes the capture rate  $C$ , the annihilation rate  $C_A$ , and time  $t$  as input and returns the number of dark matter particles  $N$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.947142+00:00]) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:11:27.267451+00:00]) Grade: A

Notes: The submitted solution correctly follows the process of separating variables and integrating both sides of the differential equation. The use of partial fraction decomposition is consistent with the standard solution, and the integration steps are correctly executed. The final expression for  $N$  matches the standard solution, showing that the solution is physically equivalent. The Python code provided correctly implements the derived formula. Overall, the solution demonstrates a clear understanding of the mathematical concepts involved.



## 2.2.8 Model: Qwen/Qwen2.5-7B-Instruct

Attempt 1 (2025-02-11T08:18:43.532156+00:00)

**Detailed Solution:** To solve the given problem, we need to find the function  $N(t)$  that satisfies the differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understand the Problem The given differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

This is a separable differential equation. We can separate the variables  $N$  and  $t$  as follows:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 2: Integrate Both Sides To integrate the left-hand side, we use partial fraction decomposition. The integrand can be written as:

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C} \cdot \frac{1}{1 - \left(\sqrt{\frac{C_A}{C}} N\right)^2}$$

Let  $k = \sqrt{\frac{C_A}{C}}$ . Then the integral becomes:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \frac{1}{C} \int \frac{dN}{1 - k^2 N^2}$$

The integral of  $\frac{1}{1 - k^2 N^2}$  is:

$$\int \frac{dN}{1 - k^2 N^2} = \frac{1}{2k} \ln \left| \frac{1 + kN}{1 - kN} \right| + C_1$$

Thus, we have:

$$\frac{1}{C} \cdot \frac{1}{2k} \ln \left| \frac{1 + kN}{1 - kN} \right| = t + C_2$$

Multiplying both sides by  $2kC$ :

$$\ln \left| \frac{1 + kN}{1 - kN} \right| = 2kCt + C_3$$

Exponentiating both sides:

$$\left| \frac{1 + kN}{1 - kN} \right| = e^{2kCt + C_3} = A e^{2kCt}$$

where  $A = e^{C_3}$ .

Step 3: Apply Initial Condition Using the initial condition  $N(0) = 0$ :

$$\left| \frac{1 + k \cdot 0}{1 - k \cdot 0} \right| = A \implies A = 1$$

Thus, we have:

$$\left| \frac{1 + kN}{1 - kN} \right| = e^{2kCt}$$

Since  $N$  is positive for  $t > 0$ , we can drop the absolute value:

$$\frac{1 + kN}{1 - kN} = e^{2kCt}$$

Solving for  $N$ :

$$\begin{aligned} 1 + kN &= e^{2kCt} (1 - kN) \\ 1 + kN &= e^{2kCt} - kN e^{2kCt} \\ kN + kN e^{2kCt} &= e^{2kCt} - 1 \end{aligned}$$

$$kN(1 + e^{2kCt}) = e^{2kCt} - 1$$

$$N = \frac{e^{2kCt} - 1}{k(e^{2kCt} + 1)}$$

Substituting back  $k = \sqrt{\frac{C_A}{C}}$ :

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}Ct} - 1}{\sqrt{\frac{C_A}{C}}(e^{2\sqrt{\frac{C_A}{C}}Ct} + 1)}$$

Step 4: Implement the Solution in Python

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    N = (math.exp(2 * k * C * t) - 1) / (k * (math.exp(2 * k * C * t) + 1))
    return N
```

This function `answer(C, C_A, t)` computes the number of dark matter particles  $N(t)$  at time  $t$  given the capture rate  $C$  and the annihilation rate  $C_A$ .

**Verification Results:**

**Auto verification result** ([`'2025-02-18T05:49:51.935212+00:00'`]) Status: **Correct**

**Evaluations:**

**Evaluated by gpt-4o** ([`'2025-02-12T03:08:08.874491+00:00'`]) Grade: A

Notes: The submitted solution correctly follows the standard solution's approach by separating variables and integrating both sides of the differential equation. The use of partial fraction decomposition and integration is accurate. The solution correctly applies the initial condition to determine the constant of integration. The final expression for  $N(t)$  is mathematically equivalent to the standard solution's result, as both expressions represent the same physical behavior of the system. The Python implementation is a correct translation of the mathematical solution into code. Overall, the solution demonstrates a clear understanding of the problem and provides a valid and equivalent result.

**Attempt 2** (2025-02-11T08:19:05.519758+00:00)

**Detailed Solution:** To solve the given problem, we need to find the function  $N(t)$  that satisfies the differential equation

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understanding the Problem The given differential equation is a first-order nonlinear ordinary differential equation (ODE). We need to solve it to find  $N(t)$ .

Step 2: Devise a Plan To solve the ODE, we can use separation of variables. The equation can be rewritten as

$$\frac{dN}{dt} = C - C_A N^2.$$

Separating the variables, we get

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 3: Carry Out the Plan We integrate both sides of the equation. The left-hand side requires partial fraction decomposition. Let's rewrite the integral:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To integrate the left-hand side, we use the substitution  $N = \sqrt{\frac{C}{C_A}} \tan(\theta)$ . Then  $dN = \sqrt{\frac{C}{C_A}} \sec^2(\theta) d\theta$ . Substituting this into the integral, we get

$$\int \frac{\sqrt{\frac{C}{C_A}} \sec^2(\theta) d\theta}{C - C_A \left(\sqrt{\frac{C}{C_A}} \tan(\theta)\right)^2} = \int dt.$$

Simplifying the denominator,

$$\int \frac{\sqrt{\frac{C}{C_A}} \sec^2(\theta) d\theta}{C - C \tan^2(\theta)} = \int dt.$$

$$\int \frac{\sqrt{\frac{C}{C_A}} \sec^2(\theta) d\theta}{C(1 - \tan^2(\theta))} = \int dt.$$

$$\int \frac{\sqrt{\frac{C}{C_A}} \sec^2(\theta) d\theta}{C \cos^2(\theta)} = \int dt.$$

$$\int \frac{\sqrt{\frac{C}{C_A}} d\theta}{C} = \int dt.$$

$$\frac{1}{\sqrt{CC_A}} \int d\theta = \int dt.$$

$$\frac{1}{\sqrt{CC_A}} \theta = t + C_1.$$

Since  $N = \sqrt{\frac{C}{C_A}} \tan(\theta)$ , we have  $\theta = \tan^{-1}\left(\frac{N\sqrt{C_A}}{\sqrt{C}}\right)$ . Substituting back, we get

$$\frac{1}{\sqrt{CC_A}} \tan^{-1}\left(\frac{N\sqrt{C_A}}{\sqrt{C}}\right) = t + C_1.$$

Using the initial condition  $N(0) = 0$ , we find  $C_1 = 0$ . Thus,

$$\tan^{-1}\left(\frac{N\sqrt{C_A}}{\sqrt{C}}\right) = \sqrt{CC_A}t.$$

Taking the tangent of both sides,

$$\frac{N\sqrt{C_A}}{\sqrt{C}} = \tan(\sqrt{CC_A}t).$$

Solving for  $N$ ,

$$N = \sqrt{\frac{C}{C_A}} \tan(\sqrt{CC_A}t).$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \tan(\sqrt{CC_A}t).$$

Python Code:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return math.sqrt(C / C_A) * math.tan(math.sqrt(C * C_A) * t)
```

This function implements the solution  $N(t) = \sqrt{\frac{C}{C_A}} \tan(\sqrt{CC_A}t)$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.935621+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:08:20.611180+00:00]) Grade: C

Notes: The submitted solution shows an understanding of solving a first-order nonlinear ODE using separation of variables and substitution. However, the final answer  $N(t) = \sqrt{\frac{C}{C_A}} \tan(\sqrt{CC_A}t)$  is incorrect compared to the standard solution  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t}-1)}{(e^{2\sqrt{C}t}+1)}$ . The approach using trigonometric substitution is valid, but it leads to a different form that does not match the expected solution. The error lies in the integration and substitution steps, which result in a tangent function rather than the hyperbolic tangent form that matches the standard solution. The submitted solution does not reach the correct steady state behavior as  $t \rightarrow \infty$ .

**Attempt 3** (2025-02-11T08:19:28.132683+00:00)

**Detailed Solution:** To solve the given problem, we need to find the function  $N(t)$  that satisfies the differential equation

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understanding the Problem The given differential equation is a first-order nonlinear ordinary differential equation (ODE). The unknown function is  $N(t)$ , and the constants are  $C$  and  $C_A$ .

Step 2: Devise a Plan To solve this ODE, we can use separation of variables. We rewrite the equation as

$$\frac{dN}{dt} = C - C_A N^2.$$

Separating the variables, we get

$$\frac{dN}{C - C_A N^2} = dt.$$

Step 3: Carry Out the Plan We integrate both sides of the equation. The left-hand side requires partial fraction decomposition. We rewrite the integrand as

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C}N^2)} = \frac{1}{C} \cdot \frac{1}{1 - \left(\frac{\sqrt{C_A}}{\sqrt{C}}N\right)^2}.$$

Let  $k = \sqrt{\frac{C_A}{C}}$ . Then the integral becomes

$$\int \frac{dN}{1 - k^2 N^2} = \int dt.$$

Using the standard integral

$$\int \frac{dx}{1 - x^2} = \frac{1}{2} \ln \left| \frac{1+x}{1-x} \right| + C,$$

we get

$$\frac{1}{2k} \ln \left| \frac{1+kN}{1-kN} \right| = t + C.$$

Applying the initial condition  $N(0) = 0$ , we find  $C = 0$ . Thus,

$$\frac{1}{2k} \ln \left| \frac{1+kN}{1-kN} \right| = t.$$

Exponentiating both sides, we obtain

$$\left| \frac{1+kN}{1-kN} \right| = e^{2kt}.$$

Since  $N$  is positive and small initially, we can drop the absolute value:

$$\frac{1+kN}{1-kN} = e^{2kt}.$$

Solving for  $N$ , we get

$$1 + kN = e^{2kt}(1 - kN).$$

Rearranging terms, we have

$$\begin{aligned}1 + kN &= e^{2kt} - kNe^{2kt}, \\kN + kNe^{2kt} &= e^{2kt} - 1, \\N(k + ke^{2kt}) &= e^{2kt} - 1, \\N &= \frac{e^{2kt} - 1}{k(1 + e^{2kt})}.\end{aligned}$$

Substituting back  $k = \sqrt{\frac{C_A}{C}}$ , we get

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}.$$

Final Answer:

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}.$$

Python Code Implementation:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    N = (math.exp(2 * k * t) - 1) / (k * (1 + math.exp(2 * k * t)))
    return N
```

This code defines the function answer that computes  $N(t)$  given the constants  $C$  and  $C_A$  and the time  $t$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.936056+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:08:25.775208+00:00']) Grade: A

Notes: The submitted solution correctly applies separation of variables and partial fraction decomposition to solve the differential equation. The integration process is accurate, and the final expression for  $N(t)$  is mathematically equivalent to the standard solution. The solution process includes all necessary steps and reasoning, and the Python code implementation correctly reflects the derived formula. The approach and final answer are consistent with the standard solution, demonstrating a clear understanding of the problem.

**Attempt 4** (2025-02-11T08:22:42.258405+00:00)

**Attempt 5** (2025-02-11T08:25:56.410649+00:00)

**Detailed Solution:** To solve the given problem, we need to solve the differential equation:

$$\dot{N} = C - C_A N^2$$

with the initial condition  $N(0) = 0$ .

Step 1: Understand the Problem The differential equation is a first-order nonlinear ordinary differential equation (ODE). The unknown is  $N(t)$ , and the given data are the constants  $C$  and  $C_A$ .

Step 2: Devise a Plan To solve this ODE, we can use separation of variables. We rewrite the equation as:

$$\frac{dN}{dt} = C - C_A N^2$$

Separating the variables, we get:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3: Carry Out the Plan We integrate both sides of the equation. The left side requires a partial fraction decomposition. Let's integrate:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

We can rewrite the denominator as:

$$C - C_A N^2 = C\left(1 - \frac{C_A}{C} N^2\right) = C(1 - kN^2)$$

where  $k = \frac{C_A}{C}$ . Thus, the integral becomes:

$$\int \frac{dN}{C(1 - kN^2)} = \int dt$$

This can be solved using the standard integral:

$$\int \frac{dN}{1 - kN^2} = \frac{1}{2\sqrt{k}} \ln \left| \frac{1 + \sqrt{k}N}{1 - \sqrt{k}N} \right|$$

So, we have:

$$\frac{1}{2\sqrt{k}} \ln \left| \frac{1 + \sqrt{k}N}{1 - \sqrt{k}N} \right| = t + C_1$$

Using the initial condition  $N(0) = 0$ , we find  $C_1 = 0$ . Thus:

$$\frac{1}{2\sqrt{k}} \ln \left| \frac{1 + \sqrt{k}N}{1 - \sqrt{k}N} \right| = t$$

Exponentiating both sides, we get:

$$\left| \frac{1 + \sqrt{k}N}{1 - \sqrt{k}N} \right| = e^{2\sqrt{k}t}$$

Since  $N(0) = 0$ , we can drop the absolute value:

$$\frac{1 + \sqrt{k}N}{1 - \sqrt{k}N} = e^{2\sqrt{k}t}$$

Solving for  $N$ :

$$1 + \sqrt{k}N = e^{2\sqrt{k}t}(1 - \sqrt{k}N)$$

$$1 + \sqrt{k}N = e^{2\sqrt{k}t} - \sqrt{k}N e^{2\sqrt{k}t}$$

$$1 + \sqrt{k}N + \sqrt{k}N e^{2\sqrt{k}t} = e^{2\sqrt{k}t}$$

$$\sqrt{k}N(1 + e^{2\sqrt{k}t}) = e^{2\sqrt{k}t} - 1$$

$$N = \frac{e^{2\sqrt{k}t} - 1}{\sqrt{k}(1 + e^{2\sqrt{k}t})}$$

Substituting back  $k = \frac{C_A}{C}$ :

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}$$

Simplifying:



















$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}} + \sqrt{\frac{C_A}{C}}e^{2\sqrt{\frac{C_A}{C}}t}}$$

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}$$

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}} + \sqrt{\frac{C_A}{C}}e^{2\sqrt{\frac{C_A}{C}}t}}$$

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}$$

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}} + \sqrt{\frac{C_A}{C}}e^{2\sqrt{\frac{C_A}{C}}t}}$$

$$N(t) = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}$$

$$N(t) =$$

Verification Results:

**Auto verification result** ([\*2025-02-18T05:49:51.936121+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T03:08:41.905733+00:00']) Grade: A

Notes: The submitted solution correctly applies separation of variables to solve the differential equation  $\dot{N} = C - C_A N^2$ . The integration process is consistent with the standard solution, using partial fraction decomposition and the standard integral for  $\int \frac{dN}{1-kN^2}$ . The final expression for  $N(t)$  is mathematically equivalent to the standard solution,  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$ , after accounting for the substitution  $k = \frac{C_A}{C}$ . The solution process is thorough and demonstrates a correct understanding of the mathematical concepts involved.

## 2.2.9 Model: meta-llama/Meta-Llama-3.1-8B-Instruct

Attempt 1 (2025-02-11T07:54:12.291617+00:00)

**Detailed Solution:** Problem Solution

Understand the Problem

We are given a Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2.$$

The initial condition is  $N(0) = 0$ . We need to find  $N(t)$  as a function of time.

Devise a Plan

To solve this differential equation, we can use the method of separation of variables. We will first separate the variables  $N$  and  $t$ , and then integrate both sides to obtain the solution.

Carry Out the Plan

We start by separating the variables:

$$\frac{dN}{dt} = C - C_A N^2.$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt.$$

Now, we can integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To evaluate the left-hand side, we can use a substitution:

$$u = C_A N^2,$$

$$du = 2C_A N dN.$$

We can rewrite the left-hand side as:

$$\frac{1}{2C_A} \int \frac{du}{C - u} = \int dt.$$

Evaluating the integral, we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - u}{C} \right| = t + K,$$

where  $K$  is the constant of integration.

Substituting back  $u = C_A N^2$ , we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - C_A N^2}{C} \right| = t + K.$$

Now, we can solve for  $N$ :

$$\ln \left| \frac{C - C_A N^2}{C} \right| = 2C_A(t + K).$$

Taking the exponential of both sides, we get:

$$\frac{C - C_A N^2}{C} = e^{2C_A(t+K)}.$$

Simplifying, we get:

$$C - C_A N^2 = C e^{2C_A(t+K)}.$$

Now, we can solve for  $N$ :

$$C_A N^2 = C - C e^{2C_A(t+K)}.$$

Dividing both sides by  $C_A$ , we get:

$$N^2 = \frac{C - C e^{2C_A(t+K)}}{C_A}.$$

Taking the square root of both sides, we get:

$$N = \pm \sqrt{\frac{C - C e^{2C_A(t+K)}}{C_A}}.$$



Since  $N(0) = 0$ , we know that  $N$  must be positive. Therefore, we can take the positive square root:

$$N = \sqrt{\frac{C - Ce^{2C_A(t+K)}}{C_A}}.$$

Now, we can use the initial condition  $N(0) = 0$  to find the value of  $K$ :

$$0 = \sqrt{\frac{C - Ce^{2C_A K}}{C_A}}.$$

Squaring both sides, we get:

$$0 = C - Ce^{2C_A K}.$$

Solving for  $K$ , we get:

$$K = -\frac{1}{2C_A} \ln\left(\frac{C}{C}\right) = -\frac{1}{2C_A} \ln(1) = 0.$$

Therefore, the solution is:

$$N = \sqrt{\frac{C - Ce^{2C_A t}}{C_A}}.$$

Convert to Python Code

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) at time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.

    Returns:
    float: The number of dark matter particles N(t) at time t.
    """
    return np.sqrt(C - C * np.exp(2 * C_A * t)) / np.sqrt(C_A)
```

Execute the Code

You can execute the code using a Python interpreter or a Jupyter notebook. The function `answer(C, C_A, t)` will return the value of  $N(t)$  at time  $t$ .

**Verification Results:**

**Auto verification result** (['2025-02-18T05:49:51.932693+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** (['2025-02-12T03:07:35.349975+00:00']) Grade: C

Notes: The submitted solution attempts to solve the differential equation using separation of variables, which is a valid approach. However, there are significant issues in the integration process and the final expression for  $N(t)$ . The substitution  $u = C_A N^2$  is incorrect for the given integral, leading to an incorrect expression for  $N(t)$ . The standard solution correctly uses partial fraction decomposition to integrate the left-hand side, resulting in a different form for  $N(t)$ . The submitted solution's final expression  $N = \sqrt{\frac{C - Ce^{2C_A t}}{C_A}}$  does not match the standard solution's result and is not equivalent. The approach shows some understanding but contains major flaws in execution.

**Attempt 2** (2025-02-11T07:54:44.649053+00:00)

**Detailed Solution:** Step 1: Understand the Problem We are given a Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body. The equation is  $\dot{N} = C - C_A N^2$ , where  $C$  is the capture rate of dark matter and  $C_A$  is the dark matter annihilation rate per effective volume. We are asked to find  $N(t)$  as a function of time, given the initial condition  $N(0) = 0$ .

Step 2: Devise a Plan To solve this problem, we will first use mathematical reasoning and symbolic calculations to find the solution to the differential equation. We will then convert the final mathematical expression into Python code.

Step 3: Carry Out the Plan The given differential equation is a Riccati equation, which can be solved by making a substitution. Let  $N = \frac{1}{u}$ . Then,  $\dot{N} = -\frac{\dot{u}}{u^2}$ . Substituting these expressions into the original differential equation, we get:

$$-\frac{\dot{u}}{u^2} = C - C_A \left(\frac{1}{u}\right)^2$$

Simplifying and rearranging, we get:

$$\dot{u} = -Cu + C_A$$

This is a first-order linear differential equation, which can be solved using an integrating factor. The integrating factor is  $e^{\int C dt} = e^{Ct}$ . Multiplying both sides of the differential equation by the integrating factor, we get:

$$\frac{d}{dt}(e^{Ct}u) = C_A e^{Ct}$$

Integrating both sides with respect to  $t$ , we get:

$$e^{Ct}u = \frac{C_A}{C}e^{Ct} + K$$

where  $K$  is a constant of integration. Dividing both sides by  $e^{Ct}$ , we get:

$$u = \frac{C_A}{C} + Ke^{-Ct}$$

Substituting back  $N = \frac{1}{u}$ , we get:

$$N = \frac{C}{C_A + CKe^{-Ct}}$$

Step 4: Apply the Initial Condition We are given the initial condition  $N(0) = 0$ . Substituting  $t = 0$  into the expression for  $N$ , we get:

$$0 = \frac{C}{C_A + CK}$$

Solving for  $K$ , we get:

$$K = -\frac{C}{C_A}$$

Substituting this value of  $K$  back into the expression for  $N$ , we get:

$$N = \frac{C}{C_A - \frac{CC_A}{C_A}e^{-Ct}}$$

Simplifying, we get:

$$N = \frac{C}{\frac{C_A^2}{C_A}e^{-Ct}}$$

$$N = \frac{C_A}{C_A e^{-Ct}}$$

$$N = \frac{C_A}{C_A} e^{Ct}$$

$$N = e^{Ct}$$

However, this solution is not correct. Let's go back to the previous step and re-evaluate the solution.

$$N = \frac{C}{C_A + CK e^{-Ct}}$$

Substituting  $K = -\frac{C}{C_A}$ , we get:

$$N = \frac{C}{C_A - \frac{CC_A}{C_A} e^{-Ct}}$$

Simplifying, we get:

$$N = \frac{C}{C_A - CC_A e^{-Ct}}$$

$$N = \frac{C}{C_A(1 - C e^{-Ct})}$$

$$N = \frac{1}{1 - C e^{-Ct}}$$

This is the correct solution.

Step 5: Convert the Solution to Python Code

```
def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) as a function
    of time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.

    Returns:
    float: The number of dark matter particles N(t).
    """
    return 1 / (1 - C * np.exp(-C * t))
```

Note: The np module is used for the exp function, which is a part of the NumPy library. If you don't have NumPy installed, you can install it using pip: pip install numpy.

Step 6: Execute the Python Code To execute the Python code, you can save it to a file (e.g., `dark_matter.py`) and run it using Python: `python dark_matter.py`. You can also use an online Python interpreter or a Jupyter notebook to execute the code.

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) as a function
    of time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.

    Returns:
    float: The number of dark matter particles N(t).
    """
    return 1 / (1 - C * np.exp(-C * t))
```

Test the function

```
C = 1.0
C_A = 2.0
t = 1.0
```

```
result = answer(C, C_A, t)
print(result)
```

This will output the number of dark matter particles at time  $t$ , given the capture rate  $C$  and annihilation rate  $C_A$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.933388+00:00\*]) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([\*2025-02-12T03:07:42.153022+00:00\*]) Grade: D

Notes: The submitted solution contains significant errors and does not match the standard solution. The standard solution correctly solves the differential equation using integration by quadrature and arrives at the expression  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t}-1)}{(e^{2\sqrt{C}t}+1)}$ . The submitted solution incorrectly applies a substitution and arrives at an incorrect expression  $N = \frac{1}{1-Ce^{-Ct}}$ , which is not equivalent to the standard solution. The approach taken in the submitted solution is fundamentally flawed, leading to an incorrect final answer.

**Attempt 3** (2025-02-11T07:55:08.192764+00:00)

**Detailed Solution:** Problem Solution

Step 1: Understand the Problem We are given a Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2.$$

The initial condition is  $N(0) = 0$ . We need to find  $N(t)$  as a function of time.

Step 2: Devise a Plan To solve this differential equation, we can use the method of separation of variables. We will first separate the variables  $N$  and  $t$ , and then integrate both sides to obtain the solution.

Step 3: Carry Out the Plan We start by separating the variables:

$$\frac{dN}{dt} = C - C_A N^2.$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt.$$

Now, we can integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To evaluate the left-hand side, we can use the substitution  $u = C_A N^2$ , which gives  $du = 2C_A N dN$ . We can then rewrite the integral as:

$$\frac{1}{2C_A} \int \frac{du}{C - u} = t + K,$$

where  $K$  is the constant of integration.

Evaluating the integral, we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - u}{C - u_0} \right| = t + K,$$

where  $u_0 = C_A N_0^2 = 0$ .

Substituting back  $u = C_A N^2$ , we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - C_A N^2}{C} \right| = t + K.$$

Now, we can solve for  $N$ :

$$\ln \left| \frac{C - C_A N^2}{C} \right| = 2C_A(t + K).$$

Taking the exponential of both sides, we get:

$$\frac{C - C_A N^2}{C} = e^{2C_A(t+K)}.$$

Simplifying, we get:

$$C_A N^2 = C - C e^{2C_A(t+K)}.$$

Dividing both sides by  $C_A$ , we get:

$$N^2 = \frac{C}{C_A} - \frac{C}{C_A} e^{2C_A(t+K)}.$$

Taking the square root of both sides, we get:

$$N = \pm \sqrt{\frac{C}{C_A} - \frac{C}{C_A} e^{2C_A(t+K)}}.$$

Since  $N(0) = 0$ , we know that  $N$  must be positive, so we can discard the negative solution.

Step 4: Simplify the Solution We can simplify the solution by combining the constants:

$$N = \sqrt{\frac{C}{C_A} (1 - e^{2C_A t})}.$$

Step 5: Convert to Python Code We can convert the solution to Python code as follows:

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    Calculate the number of dark matter particles N(t) as a function of time t.

    Parameters:
    C (float): Capture rate of dark matter.
    C_A (float): Dark matter annihilation rate per effective volume.
    t (float): Time.

    Returns:
    float: Number of dark matter particles N(t).
    """
    return np.sqrt(C / C_A * (1 - np.exp(2 * C_A * t)))
```

This code defines a function `answer` that takes the capture rate  $C$ , annihilation rate  $C_A$ , and time  $t$  as input and returns the number of dark matter particles  $N(t)$ . The function uses the `numpy` library to evaluate the exponential function.

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.933763+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:07:48.654935+00:00']) Grade: C

Notes: The submitted solution attempts to solve the differential equation using separation of variables, which is a valid approach. However, there are significant issues in the integration process. The substitution  $u = C_A N^2$  is incorrect for the given integral, leading to an incorrect expression for  $N(t)$ . The standard solution correctly uses partial fraction decomposition to integrate the left-hand side, resulting in a different form of the solution. The final expression for  $N(t)$  in the submitted solution does not match the standard solution, indicating a fundamental error in the integration process. Additionally, the boundary condition  $N(0) = 0$  is not correctly applied to determine the constant of integration  $K$ .

**Attempt 4** (2025-02-11T07:55:27.330735+00:00)

**Detailed Solution: Problem Solution**

Understand the Problem

We are given a Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2.$$

The initial condition is  $N(0) = 0$ . We need to find  $N(t)$  as a function of time.

Devise a Plan

To solve this differential equation, we can use the method of separation of variables. We will rearrange the equation to separate the variables  $N$  and  $t$ , and then integrate both sides.

Carry Out the Plan

We start by rearranging the equation:

$$\dot{N} = C - C_A N^2 \Rightarrow \frac{dN}{dt} = C - C_A N^2.$$

Next, we separate the variables:

$$\frac{dN}{C - C_A N^2} = dt.$$

Now, we can integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To evaluate the left-hand side, we can use a substitution:

$$u = C_A N^2 \Rightarrow du = 2C_A N dN.$$

However, we need to be careful with the constant of integration. We can rewrite the integral as:

$$\frac{1}{\sqrt{C_A}} \int \frac{dN}{\sqrt{C - C_A N^2}} = \int dt.$$

Now, we can evaluate the integral:

$$\frac{1}{\sqrt{C_A}} \arcsin\left(\frac{N}{\sqrt{\frac{C}{C_A}}}\right) = t + C_1.$$

We can now solve for  $N$ :

$$N = \sqrt{\frac{C}{C_A}} \sin(\sqrt{C_A}(t + C_1)).$$

Using the initial condition  $N(0) = 0$ , we can find the value of  $C_1$ :

$$0 = \sqrt{\frac{C}{C_A}} \sin(\sqrt{C_A}C_1) \Rightarrow C_1 = \frac{n\pi}{\sqrt{C_A}}.$$

We can now write the final expression for  $N(t)$ :

$$N(t) = \sqrt{\frac{C}{C_A}} \sin\left(\sqrt{C_A}t + \frac{n\pi}{\sqrt{C_A}}\right).$$

Convert to Python Code

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) as a function
    of time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.
```

```

Returns:
float: The number of dark matter particles N(t).
"""
# Define the constants
n = 0 # We can choose any integer value for n

# Calculate N(t)
N = np.sqrt(C / C_A) * np.sin(np.sqrt(C_A) * t + n * np.pi / np.sqrt(C_A))

return N

```

Final Answer

The final answer is the Python function `answer(C, C_A, t)` which calculates the number of dark matter particles  $N(t)$  as a function of time.

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.934251+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:07:54.233511+00:00']) Grade: D

Notes: The submitted solution is fundamentally flawed. The standard solution involves integrating the differential equation using partial fraction decomposition and logarithmic integration, leading to a hyperbolic tangent function for the solution. The submitted solution incorrectly applies a trigonometric substitution and arrives at a sine function, which is not equivalent to the standard solution. The approach does not match the correct method for solving the given differential equation, and the final expression for  $N(t)$  is incorrect. Additionally, the Python code provided is based on this incorrect solution.

**Attempt 5** (2025-02-11T07:55:57.179369+00:00)

**Detailed Solution:** Step 1: Understand the Problem We are given a Boltzmann equation governing the number  $N$  of dark matter particles in an astrophysical body:

$$\dot{N} = C - C_A N^2.$$

The initial condition is  $N(0) = 0$ . We need to find  $N(t)$  as a function of time.

Step 2: Devise a Plan To solve this differential equation, we can use the method of separation of variables. We will first separate the variables  $N$  and  $t$ , then integrate both sides to obtain the solution.

Step 3: Carry Out the Plan We start by separating the variables:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Now, we can integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

To evaluate the left-hand side, we can use a substitution:

$$u = C_A N^2$$

$$du = 2C_A N dN$$

$$\frac{1}{2C_A} \int \frac{du}{C - u} = \int dt$$

Evaluating the integral, we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - u}{C} \right| = t + K$$

where  $K$  is a constant of integration. Substituting back  $u = C_A N^2$ , we get:

$$\frac{1}{2C_A} \ln \left| \frac{C - C_A N^2}{C} \right| = t + K$$

Now, we can solve for  $N$ :

$$\begin{aligned} \ln \left| \frac{C - C_A N^2}{C} \right| &= 2C_A t + 2C_A K \\ \frac{C - C_A N^2}{C} &= e^{2C_A t + 2C_A K} \\ C - C_A N^2 &= C e^{2C_A t + 2C_A K} \\ C_A N^2 &= C - C e^{2C_A t + 2C_A K} \\ N^2 &= \frac{C}{C_A} - \frac{C}{C_A} e^{2C_A t + 2C_A K} \end{aligned}$$

Since  $N(0) = 0$ , we can find  $K$  by substituting  $t = 0$ :

$$\begin{aligned} 0^2 &= \frac{C}{C_A} - \frac{C}{C_A} e^{2C_A(0) + 2C_A K} \\ 0 &= \frac{C}{C_A} - \frac{C}{C_A} e^{2C_A K} \\ e^{2C_A K} &= 1 \\ 2C_A K &= 0 \\ K &= 0 \end{aligned}$$

Substituting  $K = 0$  back into the equation for  $N^2$ , we get:

$$N^2 = \frac{C}{C_A} - \frac{C}{C_A} e^{2C_A t}$$

Taking the square root of both sides, we get:

$$N = \pm \sqrt{\frac{C}{C_A} - \frac{C}{C_A} e^{2C_A t}}$$

Since  $N$  represents the number of dark matter particles, it must be non-negative. Therefore, we take the positive square root:

$$N = \sqrt{\frac{C}{C_A} - \frac{C}{C_A} e^{2C_A t}}$$

Step 4: Convert to Python Code

```
import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) at time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.

    Returns:
    float: The number of dark matter particles N(t) at time t.
    """
    return np.sqrt(C / C_A - C / C_A * np.exp(2 * C_A * t))
```

Final Answer: The final answer is:



```

import numpy as np

def answer(C: float, C_A: float, t: float) -> float:
    """
    This function calculates the number of dark matter particles N(t) at time t.

    Parameters:
    C (float): The capture rate of dark matter.
    C_A (float): The dark matter annihilation rate per effective volume.
    t (float): The time at which to calculate N.

    Returns:
    float: The number of dark matter particles N(t) at time t.
    """
    return np.sqrt(C / C_A - C / C_A * np.exp(2 * C_A * t))

```

#### Verification Results:

Auto verification result (['2025-02-18T05:49:51.934765+00:00']) Status: **Incorrect**

#### Evaluations:

Evaluated by gpt-4o (['2025-02-12T03:08:01.278403+00:00']) Grade: C

Notes: The submitted solution attempts to solve the differential equation using separation of variables, which is a valid approach. However, there are significant issues in the integration process and the final expression for  $N(t)$ . The substitution  $u = C_A N^2$  is incorrect for the given integral, leading to an incorrect expression for  $N(t)$ . The correct integration involves partial fraction decomposition as shown in the standard solution.

The final expression  $N = \sqrt{\frac{C}{C_A} - \frac{C}{C_A} e^{2C_A t}}$  does not match the standard solution  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$ . The approach shows some understanding but contains major flaws in execution.

## 2.2.10 Model: Qwen/Qwen2.5-72B-Instruct

Attempt 1 (2025-02-11T19:54:13.629579+00:00)

**Detailed Solution:** Step 1: Understand the Problem

We are given the following differential equation:

$$\dot{N} = C - C_A N^2$$

where: -  $\dot{N}$  is the time derivative of  $N$ , -  $C$  is the capture rate of dark matter, -  $C_A$  is the dark matter annihilation rate per effective volume, -  $N(t)$  is the number of dark matter particles at time  $t$ .

The initial condition is:

$$N(0) = 0$$

We need to find  $N(t)$  as a function of time.

Step 2: Devise a Plan

To solve this differential equation, we will: 1. Separate the variables. 2. Integrate both sides. 3. Apply the initial condition to find the constant of integration. 4. Solve for  $N(t)$ .

Step 3: Carry Out the Plan

Step 3.1: Separate the Variables

The given differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Integrate both sides with respect to their respective variables:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side is a standard integral. We can use partial fraction decomposition to solve it. Let:

$$C - C_A N^2 = C \left( 1 - \frac{C_A}{C} N^2 \right)$$

Let  $k = \sqrt{\frac{C_A}{C}}$ . Then:

$$C - C_A N^2 = C(1 - k^2 N^2)$$

The integral becomes:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \int dt$$

Using the partial fraction decomposition:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{2k} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right)$$

Thus:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \frac{1}{2kC} \int \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) dN$$

Integrating both terms:

$$\frac{1}{2kC} (\ln|1 - kN| - \ln|1 + kN|) = t + C_1$$

Simplify the logarithms:

$$\frac{1}{2kC} \ln \left| \frac{1 - kN}{1 + kN} \right| = t + C_1$$

Step 3.3: Apply the Initial Condition

At  $t = 0$ ,  $N = 0$ :

$$\frac{1}{2kC} \ln \left| \frac{1 - 0}{1 + 0} \right| = 0 + C_1$$

$$\frac{1}{2kC} \ln 1 = C_1$$

$$C_1 = 0$$

So the equation becomes:

$$\frac{1}{2kC} \ln \left| \frac{1 - kN}{1 + kN} \right| = t$$

Step 3.4: Solve for  $N(t)$

Multiply both sides by  $2kC$ :

$$\ln \left| \frac{1 - kN}{1 + kN} \right| = 2kCt$$

Exponentiate both sides:

$$\left| \frac{1 - kN}{1 + kN} \right| = e^{2kCt}$$

Since  $kN$  is small for small  $t$ , we can drop the absolute value:

$$\frac{1 - kN}{1 + kN} = e^{2kCt}$$

Solve for  $N$ :

$$1 - kN = e^{2kCt}(1 + kN)$$

$$1 - kN = e^{2kCt} + kNe^{2kCt}$$

$$1 - e^{2kCt} = kN(1 + e^{2kCt})$$

$$N = \frac{1 - e^{2kCt}}{k(1 + e^{2kCt})}$$

Substitute  $k = \sqrt{\frac{C_A}{C}}$ :

$$N(t) = \frac{1 - e^{2\sqrt{\frac{C_A}{C}}Ct}}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}Ct})}$$

Simplify:

$$N(t) = \frac{1 - e^{2\sqrt{C_A C}t}}{\sqrt{C_A}(1 + e^{2\sqrt{C_A C}t})}$$

Final Answer:

$$N(t) = \frac{1 - e^{2\sqrt{C_A C}t}}{\sqrt{C_A}(1 + e^{2\sqrt{C_A C}t})}$$

Step 4: Convert to Python Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    numerator = 1 - math.exp(2 * k * C * t)
    denominator = k * (1 + math.exp(2 * k * C * t))
    N = numerator / denominator
    return N
```

This Python function `answer` takes the capture rate  $C$ , the annihilation rate  $C_A$ , and time  $t$  as inputs and returns the number of dark matter particles  $N(t)$  at time  $t$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.940806+00:00]) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:09:52.348953+00:00]) Grade: C

Notes: The submitted solution shows a basic understanding of the problem and attempts to solve the differential equation by separation of variables and integration. However, there are significant issues in the integration process and the final expression for  $N(t)$ . The standard solution correctly uses partial fraction decomposition and arrives at a solution involving hyperbolic functions, while the submitted solution incorrectly simplifies the logarithmic expression and does not match the standard solution's form. The final answer provided in the submitted solution is not equivalent to the standard solution's answer, indicating a major flaw in the integration or algebraic manipulation steps.

**Attempt 2** (2025-02-11T19:55:21.580605+00:00)

**Detailed Solution:** Step-by-Step Solution

1. Understand the Problem - **Unknown:** The function  $N(t)$  that describes the number of dark matter particles as a function of time. - **Given Data:** - The capture rate  $C$ . - The annihilation rate per effective volume  $C_A$ . - The initial condition  $N(0) = 0$ . - **Conditions:** - The differential equation governing  $N(t)$  is:

$$\dot{N} = C - C_A N^2.$$

2. Devise a Plan - **Step 1:** Solve the differential equation  $\dot{N} = C - C_A N^2$ . - **Step 2:** Apply the initial condition  $N(0) = 0$  to find the particular solution. - **Step 3:** Convert the final symbolic expression into Python code.

3. Carry Out the Plan

Step 1: Solve the Differential Equation The given differential equation is:

$$\dot{N} = C - C_A N^2.$$

This is a separable differential equation. We can rewrite it as:

$$\frac{dN}{C - C_A N^2} = dt.$$

Integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt.$$

To solve the integral on the left-hand side, we use partial fraction decomposition. Let:

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C(1 - k^2 N^2)},$$

where  $k^2 = \frac{C_A}{C}$ .

The integral becomes:

$$\int \frac{dN}{1 - k^2 N^2} = \int dt.$$

The integral on the left-hand side is a standard form:

$$\int \frac{dN}{1 - k^2 N^2} = \frac{1}{2k} \ln \left| \frac{1 + kN}{1 - kN} \right| + C_1.$$

Thus, we have:

$$\frac{1}{2k} \ln \left| \frac{1 + kN}{1 - kN} \right| = t + C_2.$$

Multiply both sides by  $2k$ :

$$\ln \left| \frac{1 + kN}{1 - kN} \right| = 2kt + C_3,$$

where  $C_3 = 2kC_2$ .

Exponentiate both sides:

$$\left| \frac{1 + kN}{1 - kN} \right| = e^{2kt + C_3} = Ae^{2kt},$$

where  $A = e^{C_3}$ .

Since  $A$  is a positive constant, we can write:

$$\frac{1 + kN}{1 - kN} = Ae^{2kt}.$$

Solve for  $N$ :

$$\begin{aligned}1 + kN &= Ae^{2kt}(1 - kN). \\1 + kN &= Ae^{2kt} - Ake^{2kt}N. \\1 + kN + Ake^{2kt}N &= Ae^{2kt}. \\1 + kN(1 + Ae^{2kt}) &= Ae^{2kt}. \\kN(1 + Ae^{2kt}) &= Ae^{2kt} - 1. \\N &= \frac{Ae^{2kt} - 1}{k(1 + Ae^{2kt})}.\end{aligned}$$

Step 2: Apply the Initial Condition Given  $N(0) = 0$ :

$$0 = \frac{A - 1}{k(1 + A)}.$$

This implies:

$$A - 1 = 0 \implies A = 1.$$

Thus, the solution simplifies to:

$$N = \frac{e^{2kt} - 1}{k(1 + e^{2kt})}.$$

Substitute  $k = \sqrt{\frac{C_A}{C}}$ :

$$N = \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\sqrt{\frac{C_A}{C}}t})}.$$

Simplify the expression:

$$N = \frac{C}{C_A} \frac{e^{2\sqrt{\frac{C_A}{C}}t} - 1}{e^{2\sqrt{\frac{C_A}{C}}t} + 1}.$$

Step 3: Convert to Python Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    numerator = math.exp(2 * k * t) - 1
    denominator = math.exp(2 * k * t) + 1
    N = (C / C_A) * (numerator / denominator)
    return N
```

Final Answer:

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    k = math.sqrt(C_A / C)
    numerator = math.exp(2 * k * t) - 1
    denominator = math.exp(2 * k * t) + 1
    N = (C / C_A) * (numerator / denominator)
    return N
```

Verification Results:

Auto verification result ([\*2025-02-18T05:49:51.941468+00:00\*]) Status: **Incorrect**

Evaluations:

**Evaluated by gpt-4o** ([2025-02-12T03:10:00.515933+00:00]) Grade: A

Notes: The submitted solution correctly solves the differential equation  $\dot{N} = C - C_A N^2$  using separation of variables and partial fraction decomposition. The integration process and application of the initial condition  $N(0) = 0$  are correctly handled. The final expression for  $N(t)$  is mathematically equivalent to the standard solution:  $N = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{(e^{2\sqrt{C}t} - 1)}{(e^{2\sqrt{C}t} + 1)}$ . The Python code provided correctly implements this solution. The approach and reasoning are sound, and the solution is physically equivalent to the standard answer.

**Attempt 3** (2025-02-11T19:56:38.431425+00:00)

**Detailed Solution:** Step-by-Step Solution

1. Understand the Problem We are given the following differential equation:

$$\dot{N} = C - C_A N^2$$

where  $N(t)$  is the number of dark matter particles at time  $t$ ,  $C$  is the capture rate, and  $C_A$  is the annihilation rate per effective volume. The initial condition is  $N(0) = 0$ .

2. Devise a Plan To solve this differential equation, we will: 1. Separate variables. 2. Integrate both sides.

3. Apply the initial condition to find the constant of integration. 4. Solve for  $N(t)$ .

3. Carry Out the Plan

Step 1: Separate Variables The differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

Separate the variables:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 2: Integrate Both Sides Integrate both sides:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side is a standard integral. We can use partial fractions to simplify it. Let:

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C(1 - k^2 N^2)}$$

where  $k^2 = \frac{C_A}{C}$ .

Using partial fractions:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{2k} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right)$$

Thus, the integral becomes:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{2kC} \int \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) dN$$

Integrate:

$$\frac{1}{2kC} \left( -\frac{1}{k} \ln|1 - kN| + \frac{1}{k} \ln|1 + kN| \right) = t + C_1$$

Simplify:

$$\frac{1}{2k^2C} \ln \left| \frac{1 + kN}{1 - kN} \right| = t + C_1$$

Step 3: Apply the Initial Condition At  $t = 0$ ,  $N = 0$ :

$$\frac{1}{2k^2C} \ln \left| \frac{1 + k \cdot 0}{1 - k \cdot 0} \right| = 0 + C_1$$

$$\frac{1}{2k^2C} \ln 1 = C_1$$
$$C_1 = 0$$

So the equation becomes:

$$\frac{1}{2k^2C} \ln \left| \frac{1+kN}{1-kN} \right| = t$$

Step 4: Solve for  $N(t)$  Multiply both sides by  $2k^2C$ :

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2k^2Ct$$

Exponentiate both sides:

$$\left| \frac{1+kN}{1-kN} \right| = e^{2k^2Ct}$$

Since  $kN$  is small for small  $t$ , we can drop the absolute value:

$$\frac{1+kN}{1-kN} = e^{2k^2Ct}$$

Solve for  $N$ :

$$\begin{aligned} 1+kN &= e^{2k^2Ct}(1-kN) \\ 1+kN &= e^{2k^2Ct} - kNe^{2k^2Ct} \\ 1+kN + kNe^{2k^2Ct} &= e^{2k^2Ct} \\ kN(1+e^{2k^2Ct}) &= e^{2k^2Ct} - 1 \\ N &= \frac{e^{2k^2Ct} - 1}{k(1+e^{2k^2Ct})} \end{aligned}$$

Recall  $k^2 = \frac{C_A}{C}$ , so  $k = \sqrt{\frac{C_A}{C}}$ :

$$\begin{aligned} N &= \frac{e^{2\frac{C_A}{C}Ct} - 1}{\sqrt{\frac{C_A}{C}}(1+e^{2\frac{C_A}{C}Ct})} \\ N &= \frac{e^{2C_A t} - 1}{\sqrt{\frac{C_A}{C}}(1+e^{2C_A t})} \\ N &= \frac{C}{\sqrt{C C_A}} \frac{e^{2C_A t} - 1}{1+e^{2C_A t}} \\ N &= \sqrt{\frac{C}{C_A}} \frac{e^{2C_A t} - 1}{1+e^{2C_A t}} \end{aligned}$$

Final Answer:

$$N(t) = \sqrt{\frac{C}{C_A}} \frac{e^{2C_A t} - 1}{1+e^{2C_A t}}$$

Python Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    numerator = math.exp(2 * C_A * t) - 1
    denominator = 1 + math.exp(2 * C_A * t)
    N = math.sqrt(C / C_A) * (numerator / denominator)
    return N
```

This Python function answer takes the capture rate  $C$ , the annihilation rate  $C_A$ , and time  $t$  as inputs and returns the number of dark matter particles  $N(t)$  at time  $t$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.941900+00:00\*]) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:10:07.363348+00:00]) Grade: A

Notes: The submitted solution correctly follows the separation of variables and integration steps to solve the differential equation. The use of partial fractions to integrate  $\frac{1}{C-C_A N^2}$  is appropriate and matches the standard solution's approach. The final expression for  $N(t)$  is mathematically equivalent to the standard solution:  $N(t) = \frac{\sqrt{C}}{\sqrt{C_A}} \frac{e^{2C_A t} - 1}{1 + e^{2C_A t}}$ . The solution process includes all necessary steps and correctly applies the initial condition to find the integration constant. The Python code provided also correctly implements the derived formula. Overall, the solution demonstrates a clear understanding of the problem and the correct application of mathematical techniques.

**Attempt 4** (2025-02-11T19:57:51.944991+00:00)

**Detailed Solution:** Step 1: Understand the Problem

We are given the following differential equation:

$$\dot{N} = C - C_A N^2$$

where  $N(t)$  is the number of dark matter particles at time  $t$ ,  $C$  is the capture rate, and  $C_A$  is the dark matter annihilation rate per effective volume. The initial condition is  $N(0) = 0$ .

Step 2: Devise a Plan

To solve this differential equation, we will: 1. Separate the variables. 2. Integrate both sides. 3. Apply the initial condition to find the constant of integration. 4. Solve for  $N(t)$ .

Step 3: Carry Out the Plan

Step 3.1: Separate the Variables

The given differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Integrate both sides with respect to their respective variables:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side is a standard integral. We can use partial fraction decomposition to simplify it. Let:

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C(1 - k^2 N^2)}$$

where  $k^2 = \frac{C_A}{C}$ .

Using partial fractions, we have:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{2k} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right)$$

Thus, the integral becomes:

$$\int \frac{dN}{C - C_A N^2} = \frac{1}{2kC} \int \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) dN$$

Integrating both terms:

$$\frac{1}{2kC} \left( -\frac{1}{k} \ln|1 - kN| + \frac{1}{k} \ln|1 + kN| \right) = t + C_1$$

Simplify:

$$\frac{1}{2k^2C} (\ln|1 + kN| - \ln|1 - kN|) = t + C_1$$

Using the properties of logarithms:

$$\frac{1}{2k^2C} \ln \left| \frac{1 + kN}{1 - kN} \right| = t + C_1$$



Step 3.3: Apply the Initial Condition

At  $t = 0$ ,  $N = 0$ :

$$\frac{1}{2k^2C} \ln \left| \frac{1+0}{1-0} \right| = 0 + C_1$$
$$\frac{1}{2k^2C} \ln 1 = C_1$$
$$C_1 = 0$$

Thus, the equation simplifies to:

$$\frac{1}{2k^2C} \ln \left| \frac{1+kN}{1-kN} \right| = t$$

Step 3.4: Solve for  $N(t)$

Multiply both sides by  $2k^2C$ :

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2k^2Ct$$

Exponentiate both sides:

$$\left| \frac{1+kN}{1-kN} \right| = e^{2k^2Ct}$$

Since  $kN$  is small for small  $t$ , we can drop the absolute value:

$$\frac{1+kN}{1-kN} = e^{2k^2Ct}$$

Solve for  $N$ :

$$1+kN = e^{2k^2Ct}(1-kN)$$
$$1+kN = e^{2k^2Ct} - kNe^{2k^2Ct}$$
$$1+kN + kNe^{2k^2Ct} = e^{2k^2Ct}$$
$$kN(1 + e^{2k^2Ct}) = e^{2k^2Ct} - 1$$
$$N = \frac{e^{2k^2Ct} - 1}{k(1 + e^{2k^2Ct})}$$

Recall that  $k^2 = \frac{C_A}{C}$ , so  $k = \sqrt{\frac{C_A}{C}}$ :

$$N(t) = \frac{e^{2\frac{C_A}{C}Ct} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\frac{C_A}{C}Ct})}$$
$$N(t) = \frac{e^{2C_A t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2C_A t})}$$

Final Answer:

$$N(t) = \frac{C}{C_A} \tanh(C_A t)$$

Step 4: Convert to Python Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return (C / C_A) * math.tanh(C_A * t)
```

This Python function implements the final solution for  $N(t)$  as a function of time  $t$ , given the capture rate  $C$  and the annihilation rate  $C_A$ .

**Verification Results:**

**Auto verification result** ([\*2025-02-18T05:49:51.942294+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:10:13.706518+00:00]) Grade: C

Notes: The submitted solution shows a basic understanding of solving the differential equation by separation of variables and integration. However, there are significant issues in the final expression for  $N(t)$ . The standard solution involves expressing the integral in terms of logarithms and solving for  $N(t)$  using exponential functions, leading to a hyperbolic tangent form. The submitted solution incorrectly simplifies the expression to  $N(t) = \frac{C}{C_A} \tanh(C_A t)$ , which does not match the standard solution. Additionally, there are errors in the integration process and the handling of constants. The approach has some merit, but the final result is incorrect.

**Attempt 5** (2025-02-11T19:59:02.880495+00:00)

**Detailed Solution:** Step 1: Understand the Problem

We are given the following differential equation:

$$\dot{N} = C - C_A N^2$$

where  $N(t)$  is the number of dark matter particles at time  $t$ ,  $C$  is the capture rate, and  $C_A$  is the dark matter annihilation rate per effective volume. The initial condition is  $N(0) = 0$ .

Step 2: Devise a Plan

To solve this differential equation, we will: 1. Separate the variables. 2. Integrate both sides. 3. Apply the initial condition to find the constant of integration. 4. Solve for  $N(t)$ .

Step 3: Carry Out the Plan

Step 3.1: Separate the Variables

The given differential equation is:

$$\frac{dN}{dt} = C - C_A N^2$$

We can rewrite this as:

$$\frac{dN}{C - C_A N^2} = dt$$

Step 3.2: Integrate Both Sides

Integrate both sides with respect to their respective variables:

$$\int \frac{dN}{C - C_A N^2} = \int dt$$

The left-hand side is a standard integral. We can use partial fraction decomposition to solve it. Let:

$$\frac{1}{C - C_A N^2} = \frac{1}{C(1 - \frac{C_A}{C} N^2)} = \frac{1}{C(1 - k^2 N^2)}$$

where  $k^2 = \frac{C_A}{C}$ .

Using partial fractions:

$$\frac{1}{1 - k^2 N^2} = \frac{1}{2k} \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right)$$

Thus, the integral becomes:

$$\int \frac{dN}{C(1 - k^2 N^2)} = \frac{1}{2kC} \int \left( \frac{1}{1 - kN} + \frac{1}{1 + kN} \right) dN$$

Integrating both terms:

$$\frac{1}{2kC} \left( -\frac{1}{k} \ln|1 - kN| + \frac{1}{k} \ln|1 + kN| \right) = t + C_1$$

Simplify:

$$\frac{1}{2k^2 C} \ln \left| \frac{1 + kN}{1 - kN} \right| = t + C_1$$

Step 3.3: Apply the Initial Condition

At  $t = 0$ ,  $N = 0$ :

$$\frac{1}{2k^2 C} \ln \left| \frac{1 + k \cdot 0}{1 - k \cdot 0} \right| = 0 + C_1$$

$$\frac{1}{2k^2C} \ln 1 = C_1$$

$$C_1 = 0$$

Thus, the equation simplifies to:

$$\frac{1}{2k^2C} \ln \left| \frac{1+kN}{1-kN} \right| = t$$

Step 3.4: Solve for  $N(t)$

Multiply both sides by  $2k^2C$ :

$$\ln \left| \frac{1+kN}{1-kN} \right| = 2k^2Ct$$

Exponentiate both sides:

$$\left| \frac{1+kN}{1-kN} \right| = e^{2k^2Ct}$$

Since  $kN$  is small for small  $t$ , we can drop the absolute value:

$$\frac{1+kN}{1-kN} = e^{2k^2Ct}$$

Solve for  $N$ :

$$1+kN = e^{2k^2Ct}(1-kN)$$

$$1+kN = e^{2k^2Ct} - kNe^{2k^2Ct}$$

$$1+kN + kNe^{2k^2Ct} = e^{2k^2Ct}$$

$$kN(1 + e^{2k^2Ct}) = e^{2k^2Ct} - 1$$

$$N = \frac{e^{2k^2Ct} - 1}{k(1 + e^{2k^2Ct})}$$

Recall that  $k^2 = \frac{C_A}{C}$ , so  $k = \sqrt{\frac{C_A}{C}}$ :

$$N(t) = \frac{e^{2\frac{C_A}{C}Ct} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2\frac{C_A}{C}Ct})}$$

$$N(t) = \frac{e^{2C_A t} - 1}{\sqrt{\frac{C_A}{C}}(1 + e^{2C_A t})}$$

Final Answer:

$$N(t) = \frac{C}{C_A} \tanh(C_A t)$$

Step 4: Convert to Python Code

```
import math

def answer(C: float, C_A: float, t: float) -> float:
    return (C / C_A) * math.tanh(C_A * t)
```

This Python function implements the final solution for  $N(t)$ .

**Verification Results:**

**Auto verification result** ([2025-02-18T05:49:51.942685+00:00']) Status: **Incorrect**

**Evaluations:**

**Evaluated by gpt-4o** ([2025-02-12T03:10:19.780635+00:00']) Grade: C

Notes: The submitted solution shows a basic understanding of the problem and attempts to solve the differential equation by separation of variables and integration. However, there are significant issues in the integration process and the final expression for  $N(t)$ . The standard solution involves a hyperbolic tangent function, while the submitted solution incorrectly derives an expression involving  $\tanh(C_A t)$ . The integration steps in the submitted solution are flawed, particularly in the handling of partial fractions and the final algebraic manipulation. The final answer does not match the standard solution, indicating a major flaw in the approach.