main

December 1, 2025

0.0.1 II.1

[25]:
```python
import math
import cmath
import matplotlib as mpl
import matplotlib.pyplot as plt
import sympy as sp
import numpy as np

TEAM_NUMBER: int = 3
a: int = TEAM_NUMBER # alias
j: complex = 1j # short hand
```

1 A

$$z(initial) = -2^{a} \times j^{4 \times a} \times |\frac{\sqrt{3} + j}{(\sqrt{3} - j)^3}|$$

substituted with `a = 3`

$$+z(initial) = -2^{3} \times j^{4 \times 3} \times |\frac{\sqrt{3} + j}{(\sqrt{3} - j)^3}|$$

1.1 Simplification

1.1.1 First term

$$-2^3 = -8$$

1.1.2 Second term

$$j^{12} = (j^4)^3 = 1^3 = 1$$

$$-8 \times 1 = -8$$

1.1.3 Third term

simplifying the fraction to polar form

$$z = \frac{\sqrt{3} + j}{(\sqrt{3} - j)^3}$$

$$z = |z|e^{j0}$$

Numerator

$$z(num) = \sqrt{3} + j$$

$$|z(num)| = \sqrt{(\sqrt{3})^2 + 1^2} = \sqrt{3+1} = \sqrt{4} = 2$$

$$\arg\left(z(num)\right) = \arctan\left(1/\sqrt{3}\right) = \frac{\pi}{6}$$

$$z(num) = 2e^{j\pi/6}$$

Denominator

$$z(den) = (\sqrt{3} - j)^3$$

$$z(base) = \sqrt{3} - j = 2e^{-j\pi/6}$$

$$z(den) = (2e^{-j\pi/6})^3 = 2^3 e^{j\times 3\times(-\pi/6)} = 8e^{-j\pi/2}$$

final fraction

$$|\frac{z(num)}{z(den)}| = |\frac{2e^{j\pi/6}}{8e^{-j\pi/2}}| = |\frac{2}{8}e^{j(\pi/6-(-\pi/2))}| = |\frac{1}{4}e^{j(2\pi/3)}|$$

Since the modulus of a complex number `re^j0` is r, the expression simplifies to `1/4`

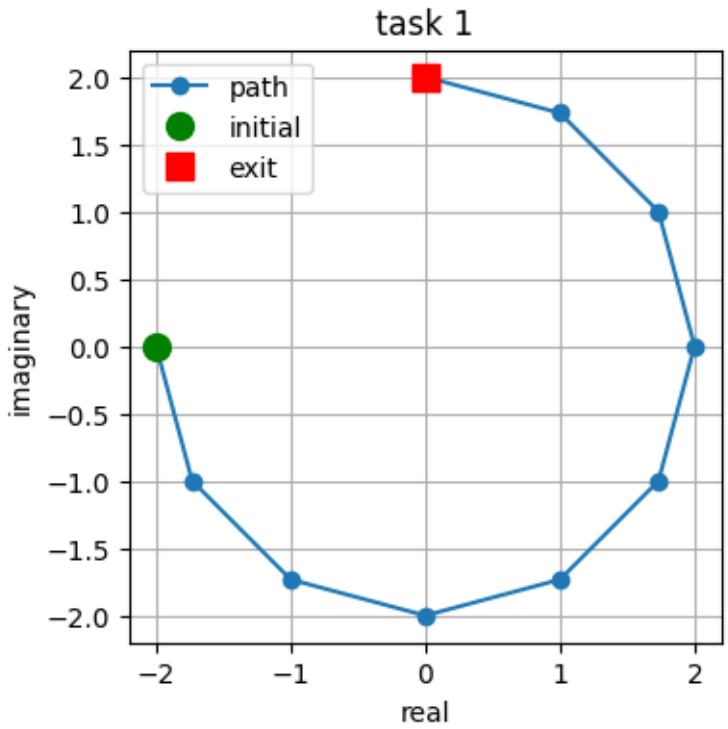
1.1.4 Simplified equation

$$z(initial) = (-8) \times (1) \times (\frac{1}{4}) = -2$$

[42]:
```
initial_position = -2.0 + 0j
current_position = initial_position
increment = (math.sqrt(3) + j) / 2
exit_position = 2j

x_array = [current_position.real]
y_array = [current_position.imag]

number_of_moves = 0
while not cmath.isclose(current_position, exit_position, rel_tol=1e-5):
    current_position *= increment
    x_array.append(current_position.real)
    y_array.append(current_position.imag)
```

```
    number_of_moves += 1

print(f'--- task A solution ---')
print(f'initial position (z_initial): {initial_position}')
print(f'exit position (z_exit): {exit_position}')
print(f'final position reached: {current_position.real:.2f} + {current_position.
  ↪imag:.2f}j')
print(f'minimum number of moves taken: {number_of_moves}')

plt.figure(figsize=(6, 4))
plt.plot(x_array, y_array, 'o-', label='path')
plt.plot(initial_position.real, initial_position.imag, 'go', markersize=10,␣
  ↪label='initial')
plt.plot(exit_position.real, exit_position.imag, 'rs', markersize=10,␣
  ↪label='exit')
plt.xlabel('real')
plt.ylabel('imaginary')
plt.title('task 1')
plt.legend()
plt.grid(True)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

```
--- task A solution ---
initial position (z_initial): (-2+0j)
exit position (z_exit): 2j
final position reached: -0.00 + 2.00j
minimum number of moves taken: 9
```



1.2 Explanation of solution

1.2.1 Initial

$$z(initial) = -2 = 2e^{j\pi}$$

1.2.2 Initial

$$z(exit) = 2j = 2e^{j\pi/2}$$

1.2.3 Multiplier

$$u = \frac{\sqrt{3}+j}{2} \times |u| = \sqrt{(\frac{\sqrt{3}}{2})^2 + *(\frac{1}{2})^2} = \sqrt{\frac{3}{4}+\frac{1}{4}} = 1$$

$$arg(u) = \arctan(\frac{1/2}{\sqrt{3}/2}) = \arctan(\frac{1}{\sqrt{3}}) = \pi/6$$

so

$$u = 1 \times e^{j\pi/6}$$

multiplying by u is a ration of

$$\pi/6rad = 30\deg$$

1.2.4 To solve

the minimum integer that solves

$$z(initial) \times u^n = zexit$$

$$2e^{j\pi} \times (e^{j\pi/6})^n = 2e^{j\pi/2}$$

Every `360deg` complex angles repeat, solve for angles of the modulo `2k{pi}`

$$\pi + n(\frac{\pi}{6}) = \frac{\pi}{2} + 2k\pi$$

solving for `n`

$$n(frac\pi 6) = -\frac{\pi}{2} + 2k\pi$$

$$n = -3 + 12k$$

the smallest possible integer for `n`. if `k = 0` then `n = -3` which is invalid. if `k = 1` then `n = -3 + 12 = 9` which is valid.

the minimum number of movements is `9`

2 B

2.1 Left door

$$y(left) = \lim_{x \to -a} a + (x+a)^{100} \times \frac{-1}{e^{(x+a)^2}}$$

$$y(left) = \lim_{x \to -3} 3 + (x+3)^{100} \times \frac{-1}{e^{(x+3)^2}}$$

$$y(left) = \lim_{u \to 0} (3 + u^{100} \times e^{-1/u^2})$$

as `u -> 0`, the term `e^(-1 / u^2)` term is exponential: reaching `0` much faster than `u^100` term which is polynomial; the limit of this product is `0`

2.2 Right door

$$y(right) = \lim_{x \to -a} \frac{e^{(x+a)}}{x + a \times e^{(x+a)}}$$

$$y(right) = \lim_{x \to -3} \frac{e^{(x+3)}}{x + 3 \times e^{(x+3)}}$$

$$y(right) = \lim_{u \to 0} \frac{e^{u}}{(u-3) + 3 \times e^{u}}$$

by direct substitution

$$y(right) = \frac{e^0}{(0-3)+3\times e^0} = \frac{1}{-3+3} = \frac{1}{0}$$

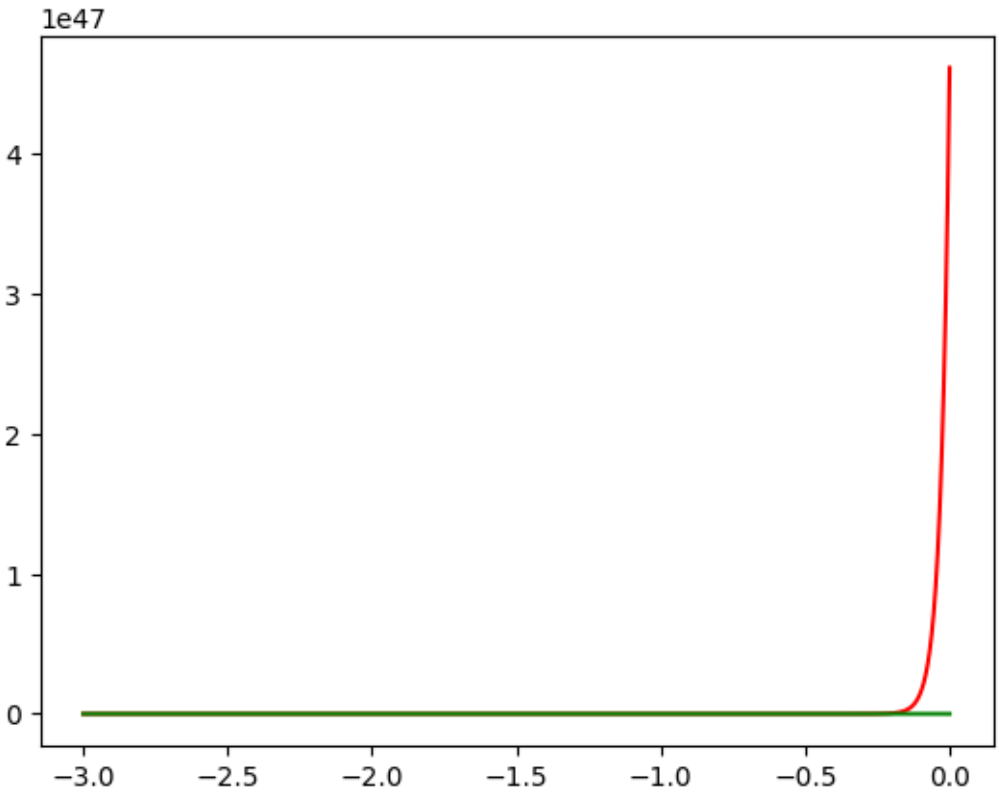
the limit is undefined and approaches **infinity**: this door will never open

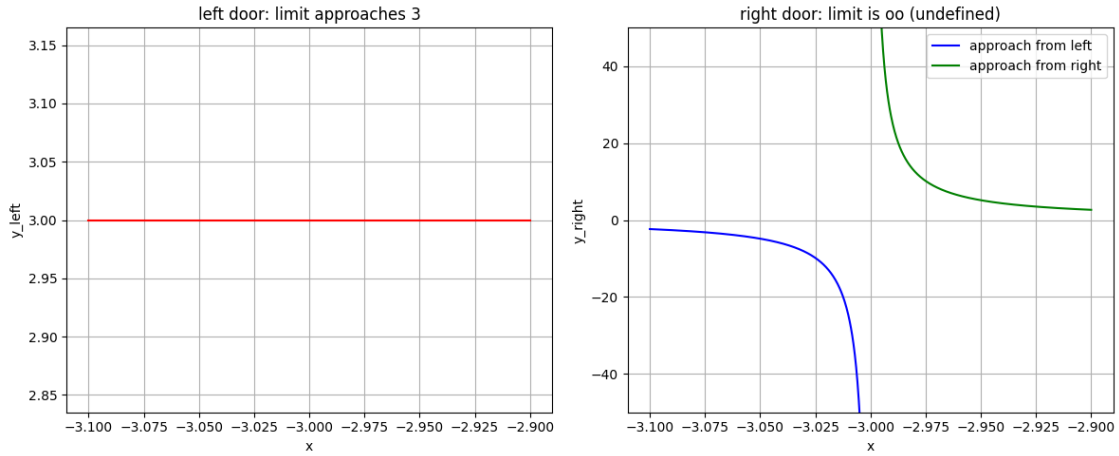
[43]:
```
x = np.linspace(0, -3, 10_000)

left = a + (x + a)**100 * np.e**(-1 / (x + a)**2)

right = (np.e**(x + a)) / (x + a * np.e ** (x + a))

plt.plot(x, left, 'r-')
plt.plot(x, right, 'g-')



x = sp.Symbol('x')
a_val = a # Use a = 3

y_left_expr = a_val + (x + a_val)**100 * sp.exp(-1 / (x + a_val)**2)
y_left_door = sp.limit(y_left_expr, x, -a_val)

y_right_expr = sp.exp(x + a_val) / (x + a_val * sp.exp(x + a_val))
y_right_door = sp.limit(y_right_expr, x, -a_val)

print(f"--- task B solution ---")
print(f"left door limit (y_left_door): {y_left_door}")
print(f"right door limit (y_right_door): {y_right_door}")

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

x_vals = np.linspace(-3.1, -2.9, 1000)
f_left = sp.lambdify(x, y_left_expr, 'numpy')
ax1.plot(x_vals, f_left(x_vals), 'r-')
ax1.set_title(f'left door: limit approaches {y_left_door}')
ax1.set_xlabel('x')
ax1.set_ylabel('y_left')
ax1.grid(True)

x_vals_l = np.linspace(-3.1, -3.0001, 500)
x_vals_r = np.linspace(-2.9999, -2.9, 500)
f_right = sp.lambdify(x, y_right_expr, 'numpy')
ax2.plot(x_vals_l, f_right(x_vals_l), 'b-', label='approach from left')
ax2.plot(x_vals_r, f_right(x_vals_r), 'g-', label='approach from right')
ax2.set_title(f'right door: limit is {y_right_door} (undefined)')
```

```
ax2.set_xlabel('x')
ax2.set_ylabel('y_right')
ax2.set_ylim(-50, 50) # Limit the view to show the asymptote
ax2.legend()
ax2.grid(True)

plt.tight_layout()
plt.show()
```

```
/tmp/ipykernel_77701/776809982.py:3: RuntimeWarning: divide by zero encountered
in divide
  left = a + (x + a)**100 * np.e**(-1 / (x + a)**2)
/tmp/ipykernel_77701/776809982.py:5: RuntimeWarning: divide by zero encountered
in divide
  right = (np.e**(x + a)) / (x + a * np.e ** (x + a))
```

```
--- task B solution ---
left door limit (y_left_door): 3
right door limit (y_right_door): oo
```





2.3 Explanation of solution

the left door will open in 3min. the right door will open in infinity minutes, this door will never open

Therefore the li6 must choose the left door

3 C

$$x \times e^{(y\times(x+a+1))} + (a+1) \times e^{(y\times(x+a+1))} = x + a$$

3.1 Solve for y

factor out the exponential term

$$x \times e^{(Y(X+a+1))} \times (x+a+1) = x + a$$

isolating the exponential term

$$e^{(y(x+a+1))} = \frac{x+a}{x+a+1}$$

take the natural log of both sides

$$y(x+a+1) = \ln\left(\frac{x+a}{x+a+1}\right)$$

solve for y

$$y = \frac{\ln\left(\frac{x+a}{x+a+1}\right)}{x+a+1}$$

3.2 Domain analysis

for y to be a real number, 2 conditions must be met

3.2.1 Condition 1, the denominator

$$den \neq 0$$

$$x + a + 1 \neq 0 \implies x + 4 \neq 0 \implies x \neq -4$$

3.2.2 Condition 2, the natural log

$$log_a rg > 0$$

$$\frac{x + a}{x + a + 1} > 0 \implies \frac{x + 3}{x + 4} > 0$$

cases of a fraction being both a numerator and denominator must have the same signs

Case 1

$$x + 3 > 0$$

and

$$x + 4 > 0 \implies > -3$$

and

$$x > -4 \implies x \neq -4$$

Case 2

$$x + 3 < 0$$

and

$$x + 4 < 0 \implies x < -3$$

and

$$x < -4 \implies x < -4$$

3.2.3 Conclusion

the unsafe domain: where y is a real number, is:

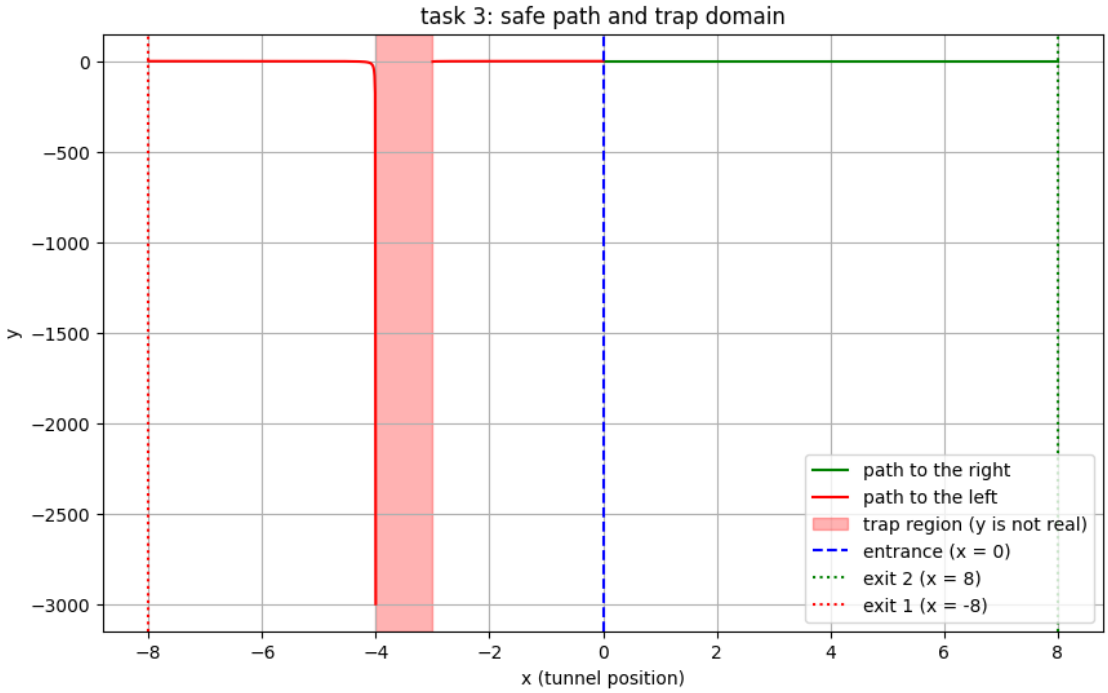
$$(-\infty, -4) \cup (-3, \infty)$$

this is the region that contains the traps

[41]:
```
a_val: int = a

def f(x, a) -> float:
    numerator = np.log10((x + a) / (x + a + 1))
    denominator = (x + a + 1)
    return numerator / denominator
```

```
x_right = np.linspace(0, 8, 400)
y_right = f(x_right, a_val)

x_left_safe = np.linspace(-2.999, 0, 400)
y_left_safe = f(x_left_safe, a_val)
x_left_danger = np.linspace(-8, -4.001, 400) # After the trap to the exit
y_left_danger = f(x_left_danger, a_val)

print(f"--- Task 3 Solution ---")
print(f"Function: y = log10((x+3)/(x+4)) / (x+4)")
print(f"Safe Domain: (-inf, -4) U (-3, +inf)")
print(f"Trap Region: [-4, -3]")

plt.figure(figsize=(10, 6))
plt.plot(x_right, y_right, 'g-', label='path to the right')
plt.plot(x_left_safe, y_left_safe, 'r-', label='path to the left')
plt.plot(x_left_danger, y_left_danger, 'r-')

plt.axvspan(-4, -3, color='red', alpha=0.3, label='trap region (y is not real)')
plt.axvline(x=0, color='blue', linestyle='--', label='entrance (x = 0)')
plt.axvline(x=8, color='green', linestyle=':', label='exit 2 (x = 8)')
plt.axvline(x=-8, color='red', linestyle=':', label='exit 1 (x = -8)')

plt.title("task 3: safe path and trap domain")
plt.xlabel("x (tunnel position)")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```

```
--- Task 3 Solution ---
Function: y = log10((x+3)/(x+4)) / (x+4)
Safe Domain: (-inf, -4) U (-3, +inf)
Trap Region: [-4, -3]
```



3.3 Explanation of solution

3.3.1 Path to the left

this path runs from `0` to `-8`, it would not be safe to cross: this path cross the trapped region of `-3` to `-4`

3.3.2 Path to the right

this path runs from `0` to `8`, it would be safe as the un trapped region of `-3` to `inf` contains this path in its entirety

Therefore the safe path is the **right path**

4 D

$$((p \to q) \to p) \to r$$

$$p : \wp(S) \equiv \{\emptyset, \{\alpha\}, \{\infty\}, \{\alpha, \infty\}\}$$

$$q : S \cap Q \cap \{+\infty\} \equiv \{\emptyset\}$$

$$r : \neg(\neg p)$$

$$S : \{\binom{3}{2} \times j, \lim_{x \to 0} \frac{1 \cdot x}{x^5}\} \cap \{\lim_{x \to 1} \frac{1}{(x-1)} + 1, \sqrt{-a}\}$$

$$Q : \{a \times e^{\ln j}\}\ T$$

4.1 Solving set `S`

4.1.1 First set

First part

$$\binom{3}{2} \times j = (3 + 2i) \times i = 3i + 2i^2 = -2 + 3i$$

Second part

$$\lim_{x \to 0} \frac{1 \cdot x}{x^5} = \lim_{x \to 0} \frac{0}{x^4} = +\infty$$

Third part

$$A = \{-2 + 3j, +\infty\}$$

4.1.2 Second set

First part limit does not exist: `x -> 1` is `-inf` and `x -> 1` is `inf`

$$\lim_{x \to 1}(\frac{1}{x-1} + 1) = \text{NaN}$$

Second part

$$\sqrt{-a} = \sqrt{-3} = j\sqrt{3}$$

Third part

$$B = \{\text{NaN}, j\sqrt{3}\}$$

4.1.3 Intersection

this is an empty set as there are no common elements in the intersection

$$S = A \cap B = \{-2 + 3j, +\infty\} \cap \{\text{NaN}, j\sqrt{3}\} = \emptyset$$

4.2 Solving set `Q`

4.2.1 First part

$$3 \times e^{\ln j} = 3 \times j = 3j$$

4.2.2 Second part

$$T(1.1) = 9$$

4.2.3 Third part

$$Q = C\ \{9\} = \{3j\}\ \{9\} = \{3j\}$$

4.3 Evaluation of `p`, `q`, and `r`

4.3.1 Proposition of `p`

$$q : S \cap Q \cap \{+\infty\} \equiv \{\emptyset\}$$

$$S \cap Q \cap \{+\infty\} = \emptyset \cap \{3j\} \cap \{+\infty\} = \emptyset$$

`s` claims equivalence to an empty set, which is not the same as a set of an empty set, which evaluates `s` to `FALSE`

4.3.2 Proposition of `r`

r is `¬(¬p)` which is double negation, since `p` is `FALSE`, `r` is also `FALSE`

4.4 Logical evaluation

`p` = `FALSE` `r` = `FALSE`

4.4.1 Step 1

$$(p \to q) \implies (F \to F) \implies T$$

4.4.2 Step 2

$$((p \to q) \to p) = (T \to F) = F$$

4.4.3 Step 3

$$((p \to q) \to p) \to r \implies (F \to F) \implies T$$

4.5 Solution

the final value is `TRUE`

[40]:
```
x = sp.Symbol('x')

limit_a2 = sp.limit(1 / x**4, x, 0)

limit_b1_left = sp.limit((1 / (x - 1)) + 1, x, 1, dir='-')
limit_b1_right = sp.limit((1 / (x - 1)) + 1, x, 1, dir='+')

print(f'--- task 4.2 limit verification ---')
print(f'limit for set a (lim x -> 0 1/x^4): {limit_a2}')
print(f'limit for set b (from left, x -> 1-): {limit_b1_left}')
print(f'limit for set b (from right, x -> 1+): {limit_b1_right}')
```

```
print(f'since the left and right limits are not equal, the limit at x=1 does␣
  ↪not exist (nan).')
```

```
--- task 4.2 limit verification ---
limit for set a (lim x -> 0 1/x^4): oo
limit for set b (from left, x -> 1-): -oo
limit for set b (from right, x -> 1+): oo
since the left and right limits are not equal, the limit at x=1 does not exist
(nan).
```

4.6 Explantation of solution

the left lever will open the door: the logical clause evaluates to TRUE Therefore Li6 must pull lever on the **left**