AP 8

Loewner differential equation

The Loewner differential equation is written as:

$$\frac{\partial g_t(z)}{\partial t} = \frac{2}{g_t(z) - \lambda(t)},$$

where:

- $g_t(z)$ is the Loewner chain,
- $\lambda(t)$ is the driving function (often continuous or piecewise).

For numerical purposes, this equation can be reformulated into a system of ODEs. Let's assume a toy driving function like $\lambda(t) = 2t$ to simplify implementation.

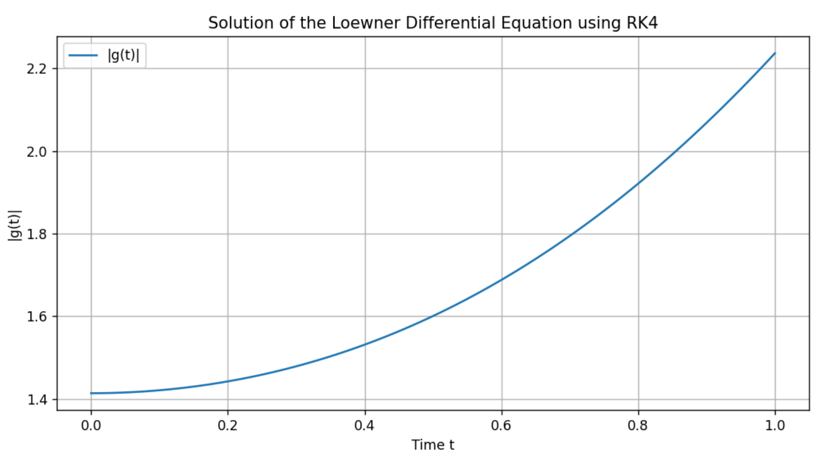
Rewriting:

$$g_t(z) = y(t), \quad \text{so the system becomes:} \quad \frac{dy}{dt} = \frac{2}{y - \lambda(t)}.$$

https://en.wikipedia.org/wiki/List_of_nonlinear_ordinary_differential_equations

Result:

The output visualizes the solution to the Loewner differential equation using the 4th-order Runge-Kutta method. The plot shows the magnitude of |g(t)| over the time interval from t = 0 to t = 1, divided into N = 100 steps. The y-axis represents the magnitude of g(t), and the x-axis represents time t. The plot helps to understand the behavior and stability of the solution over time, illustrating any trends or oscillations in the system's magnitude.



Code:

```
import numpy as np
import matplotlib.pyplot as plt

# driving function
def driving_function(t):
    return 2 * t

# Derivative function
def dg_dt(t, g):
    return driving_function(t)

# Runge-Kutta 4th order method
def runge_kutta_4(g0, t0, t_end, N):
    h = (t_end - t0) / N  # Step size
    t_values = np.linspace(t0, t_end, N + 1)
    g_values = np.zeros(N + 1, dtype=complex)

    g_values[0] = g0

    for i in range(N):
        t = t_values[i]
        g = g_values[i]

        k1 = h * dg_dt(t, g)
        k2 = h * dg_dt(t + h / 2, g + k1 / 2)
        k3 = h * dg_dt(t + h / 2, g + k2 / 2)
        k4 = h * dg_dt(t + h, g + k3)

        g_values[i + 1] = g + (k1 + 2 * k2 + 2 * k3 + k4) / 6

    return t_values, g_values

# Initial conditions
t0 = 0
g0 = 1 + 1j

# Time interval and step size
t_end = 1
N = 100
```

```python
# call RK4
time, g_values = runge_kutta_4(g0, t0, t_end, N)

# Plot results
plt.figure(figsize=(10, 5))
plt.plot(time, np.abs(g_values), label='|g(t)|')
plt.title('Solution of the Loewner Differential Equation using RK4')
plt.xlabel('Time t')
plt.ylabel('|g(t)|')
plt.legend()
plt.grid(True)
plt.show()
```