Problemstillingen

- Løse IVP
  - Eksponentiell vekst
  - Pendel
  - Dobbelpendel

Temaer i IN1910

- Fokus på struktur
  - Objektorientering, abstraction
  - Dataclasses (duck typing)
- Gradvis utvikling, lesbarhet
- Testing
  - Unit tests
  - Energibevaring

Løsning

```python
class DoublePendulum(ode.ODEModel):
    def __init__(
        self,
        L1: Optional[float] = 1.0,
        L2: Optional[float] = 1.0,
        g: Optional[float] = 9.81
    ):
        self.L1 = L1
        self.L2 = L2
        self.g = g

    @property
    def num_states(self) -> int:
        return 4

    def __call__(self, t: float, u: np.ndarray) -> np.ndarray:
        self.verify_number_of_inputs(u)
        _, omega1, _, omega2 = u
        dtheta1_dt = omega1
        domega1_dt = self.domega1_dt(u)
        dtheta2_dt = omega2
        domega2_dt = self.domega2_dt(u)
        return np.array([dtheta1_dt, domega1_dt, dtheta2_dt, domega2_dt])

    def domega1_dt(self, u: np.ndarray) -> float:
        theta1, omega1, theta2, omega2 = u
        del_th = theta2 - theta1
        L1, L2, g = self.L1, self.L2, self.g
        t1 = L1 * omega1**2 * np.sin(del_th) * np.cos(del_th)
        t2 = g * np.sin(theta2) * np.cos(del_th)
        t3 = L2 * omega2**2 * np.sin(del_th)
        t4 = - 2 * g * np.sin(theta1)
        num = t1 + t2 + t3 + t4
        denom = 2*L1 - L1*np.cos(del_th)**2
        return num/denom
```

```python
@pytest.mark.parametrize('L1', [1.0, 2.0, 3.0])
@pytest.mark.parametrize('L2', [1.0, 2.0, 3.0])
def test_solve_double_pendulum_function_zero_ic(L1, L2):
    u0 = np.zeros(4)
    T = 10.0
    model = DoublePendulum(L1=L1, L2=L2)
    sol = solve_double_pendulum(u0, T, pendulum=model)
    assert np.all(sol.theta1 == 0.0)
    assert np.all(sol.omega1 == 0.0)
    assert np.all(sol.theta2 == 0.0)
    assert np.all(sol.omega2 == 0.0)
    assert np.all(sol.x1 == 0.0)
    assert np.all(sol.y1 == -L1)
    assert np.all(sol.x2 == 0.0)
    assert np.all(sol.y2 == -L1 - L2)
```

Utfordringer

- Få eom’s riktig, refactor
- Nøyaktighet i solve_ivp
  - Problemer når veksten blir stor
  - Energibevaring
  - Velge en annen numerisk metode
  - Størrelse på tidssteg

Testing

- Unit tests
- Eksponentiell vekst er testet mot eksakt løsning
- Pendel testet mot eksakt løsning i spesialtilfeller
- Fysiske prinsipper
  - Energibevaring