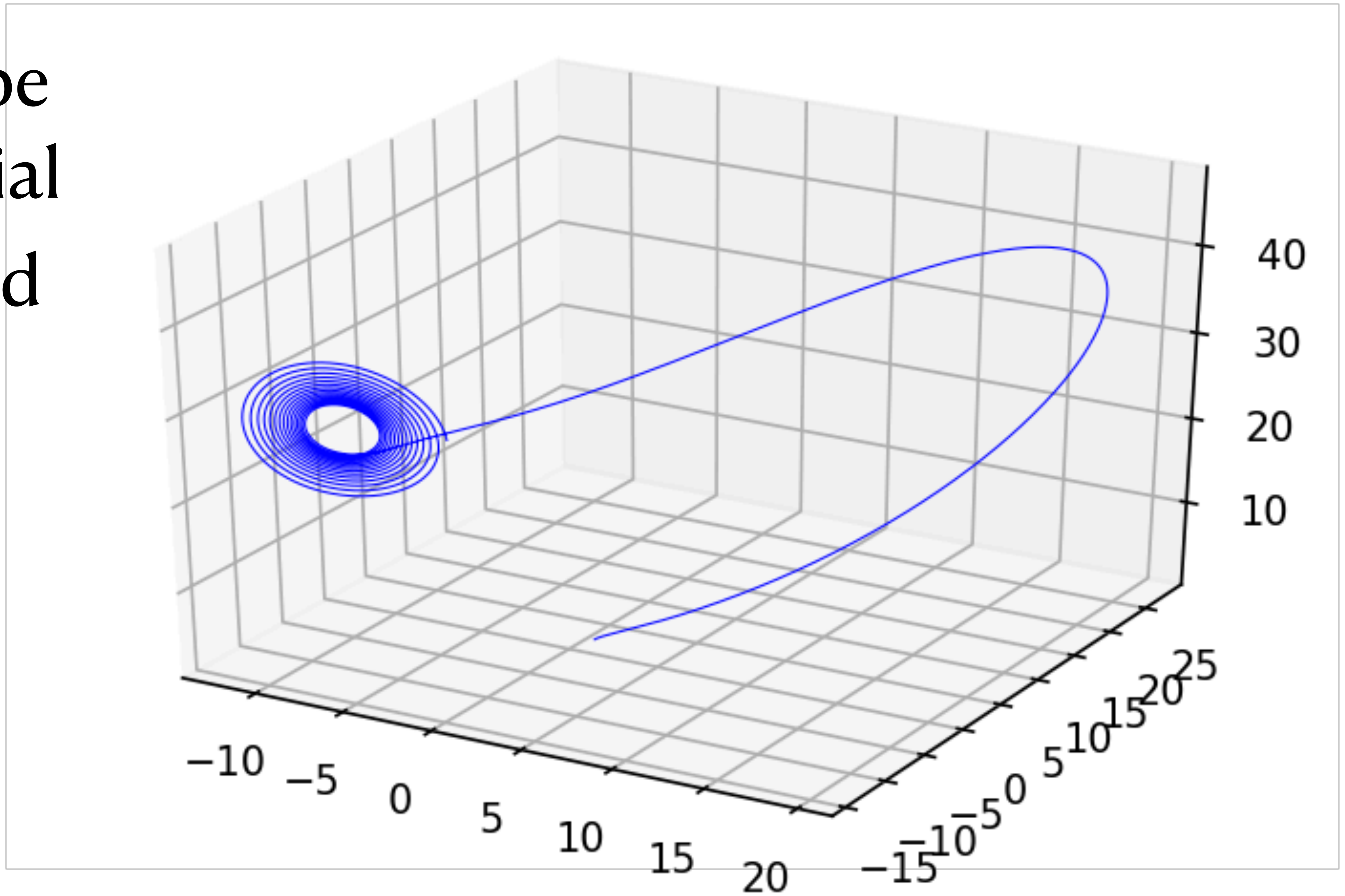
Differentiation of ODE-based functions

ODE-defined functions

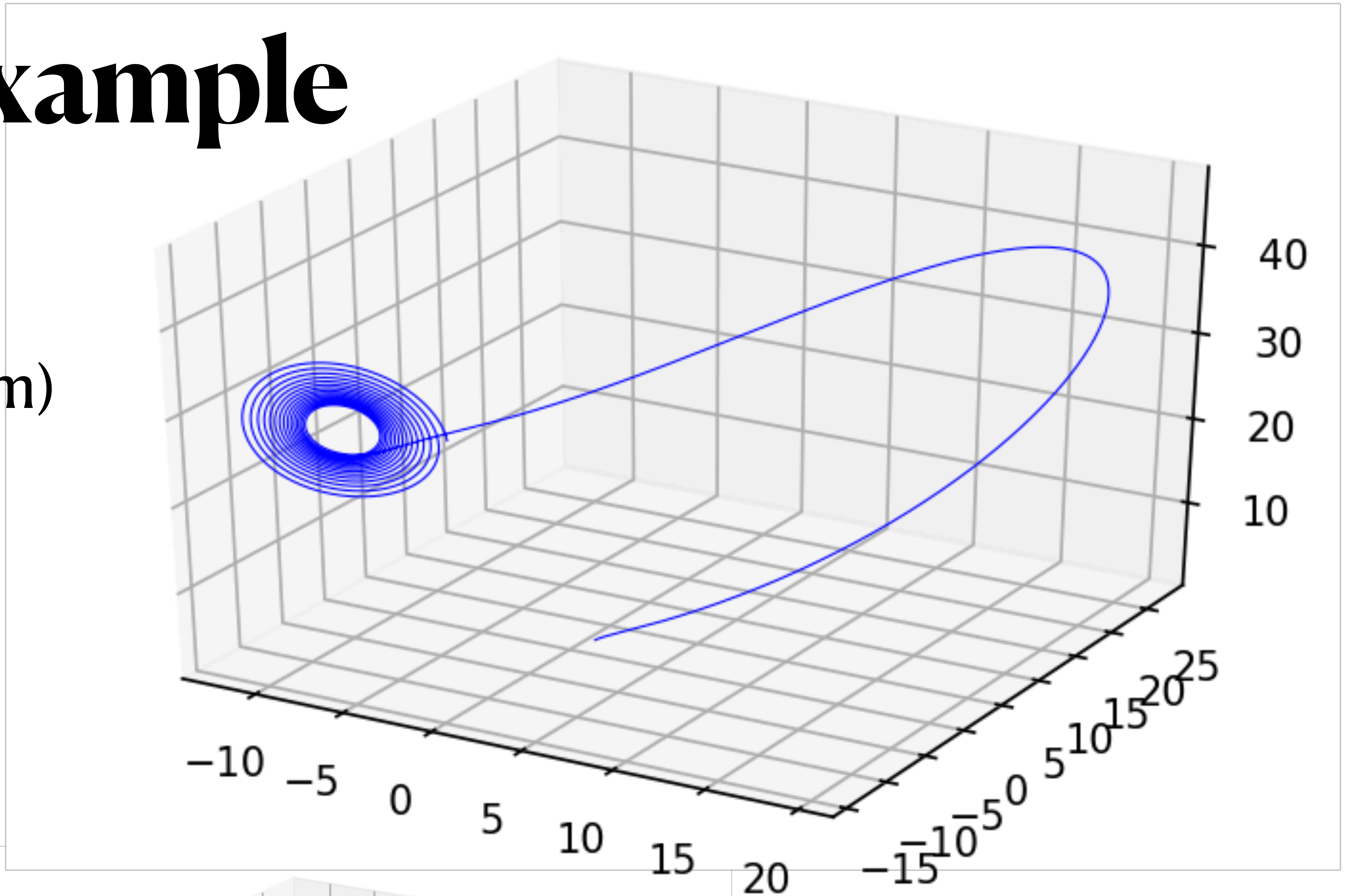
- Instead of a fixed-value system, a layer can be modelled implicitly by an ordinary differential equation with function $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ and an initial value $y_0 \in \mathbb{R}^n$

$$\dot{y}(t) = f(t, y(t))$$
$$y(0) = y_0$$

- We can use any ODE solver to solve $y(t)$ for future values of t, for instance Euler integration, or Runge-Kutta



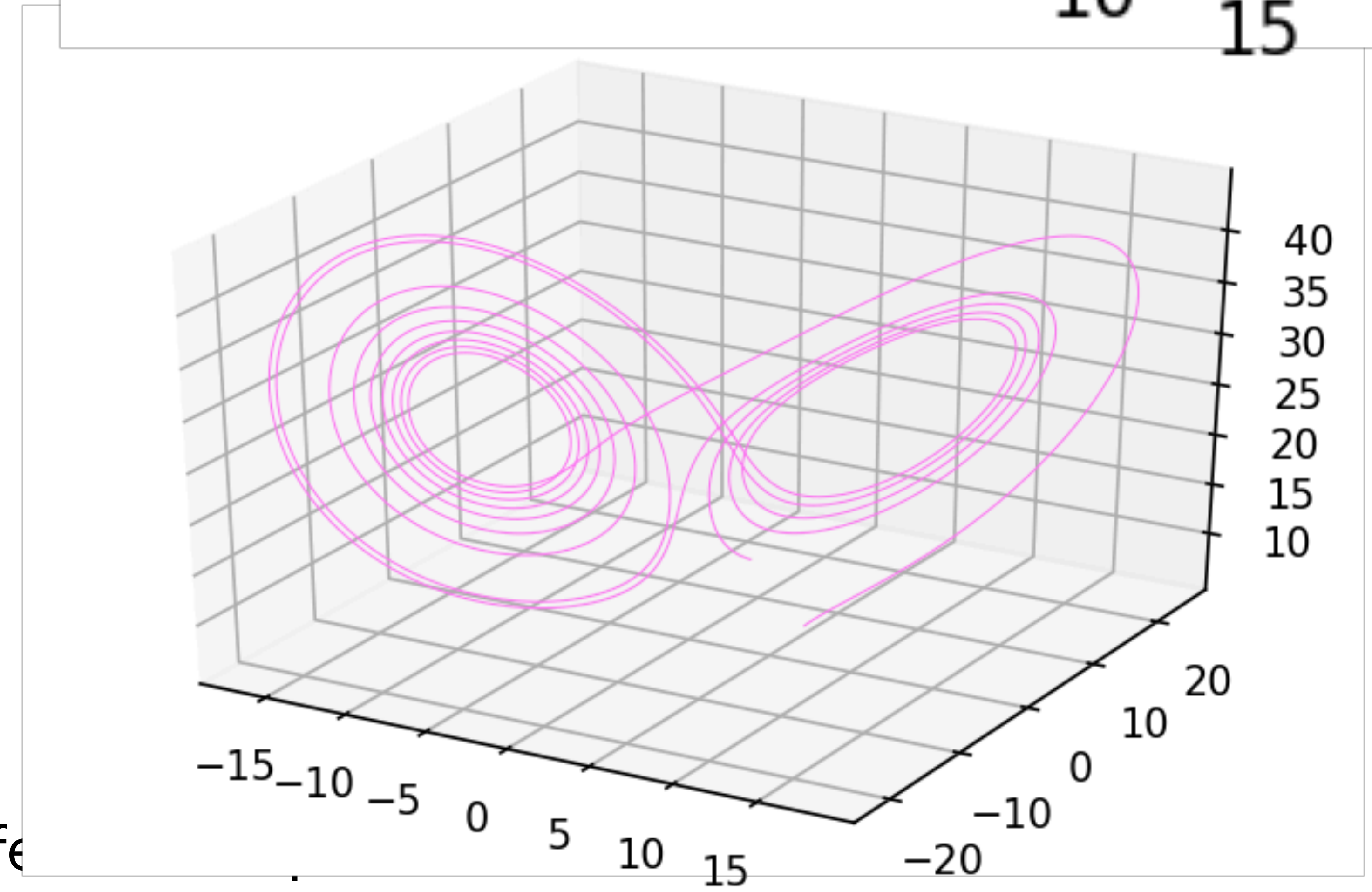
Example

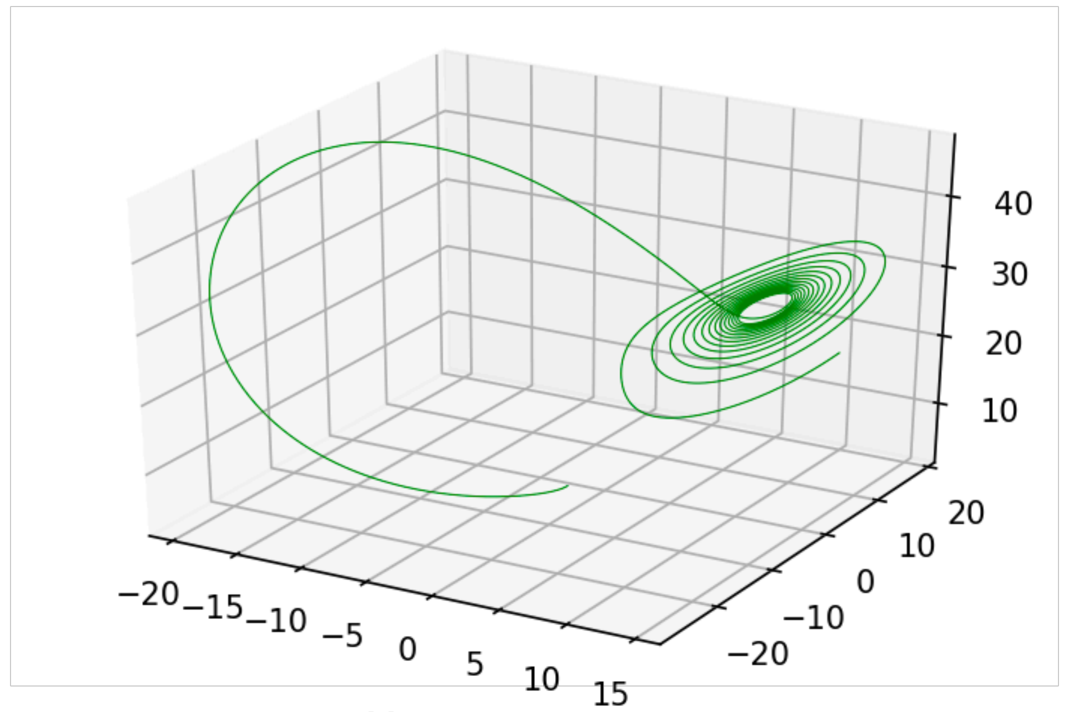


- Say we have the ODE (Lorentz system)

$$\partial_t y(t, x, y, z) = \begin{bmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{bmatrix}$$

- Forward propagating, we can have various trajectories





Differentiating through an ODE

- We can differentiate via all steps in the sequence, but we would have the same problems with memory and instability as with fixed-point layers
- Instead, we can follow a similar route with the implicit function theorem
-

JVP for ODE-based layers

- For the model: $\partial_t y(t, a, b) = f(t, y(t, a, b), a)$ with initial conditions: $y(0, a, b) = 0$
- We are interested in how much the solution to the ODE function will change if we nudge parameters a, b
- We first need to compute the derivative with respect to parameters a

$$\partial_a \Big(\partial_t y(t, a, b) \Big) = \partial_a \Big(f(t, y(t, a, b), a) \Big) = \partial_a f(t, y, a) + \partial_y f(t, y, a) \partial_a y(t, a, b) \Rightarrow$$

$$\partial_t \underbrace{\partial_a y(t, a, b)}_{z(t,a,b)} = \partial_a f(t, y, a) + \partial_y f(t, y, a) \partial_a y(t, a, b)$$

$$\partial_t z(t, a, b) = \partial_a f(t, y, a) + \partial_y f(t, y, a) z(t, a, b)$$

JVP for ODE-based layers

- That is, to compute how much a nudge in the parameters affects the gradient we must solve another ODE

$$\begin{bmatrix} \partial_t \ y(t, a, b) \\ \partial_t \ z(t, a, b) \end{bmatrix} = \begin{bmatrix} f(t, y(t, a, b), a) \\ \partial_a f(t, y, a) + \partial_y f(t, y, a) z(t, a, b) \end{bmatrix}$$

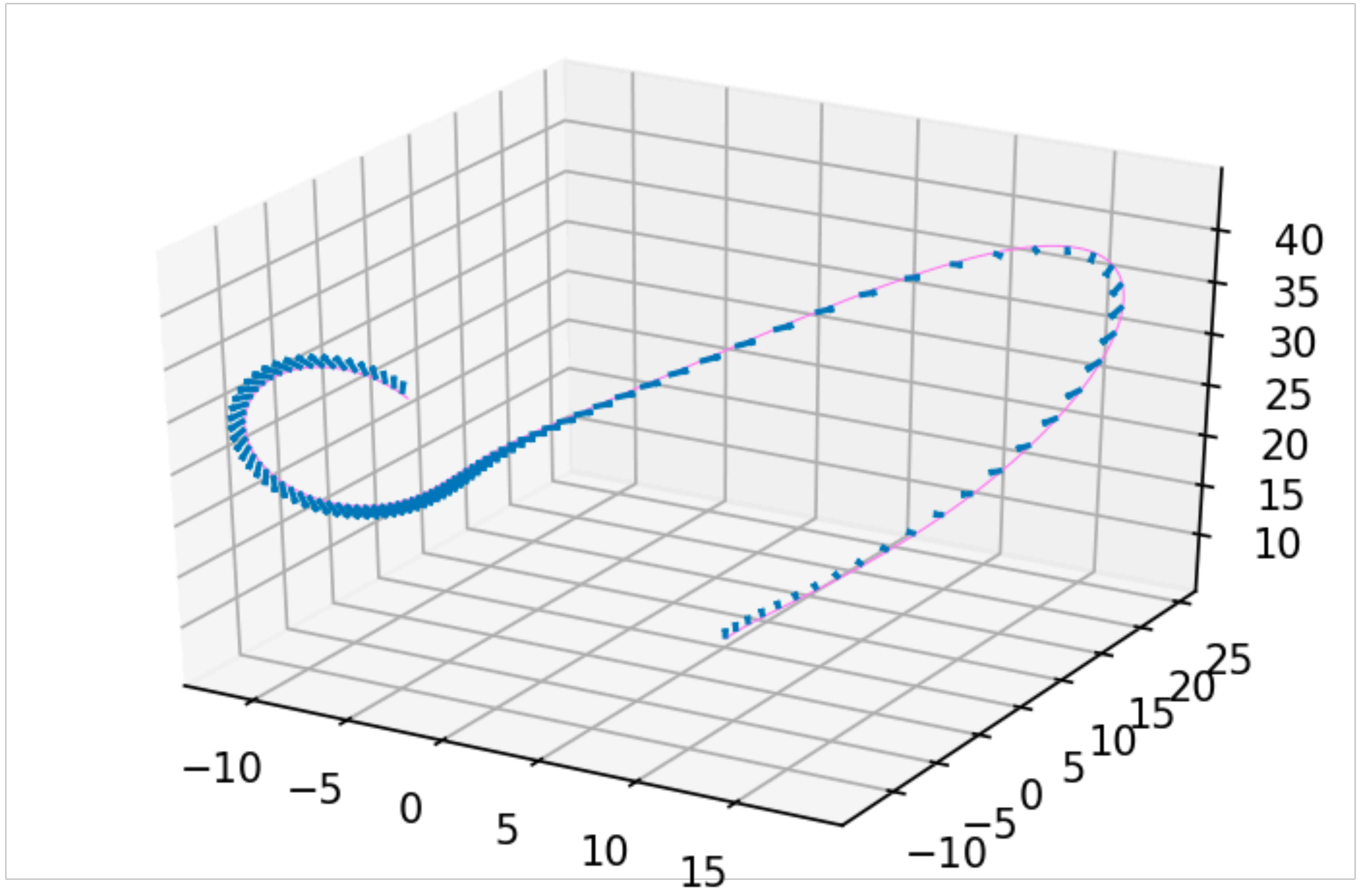
With initial conditions

$$\begin{bmatrix} y(0, a, b) \\ z(0, a, b) \end{bmatrix} = \begin{bmatrix} b \\ \Delta b \end{bmatrix}$$

JVP on our example

- Let's say we want to see how the Lorentz trajectory will change when we nudge the third dimension of the output

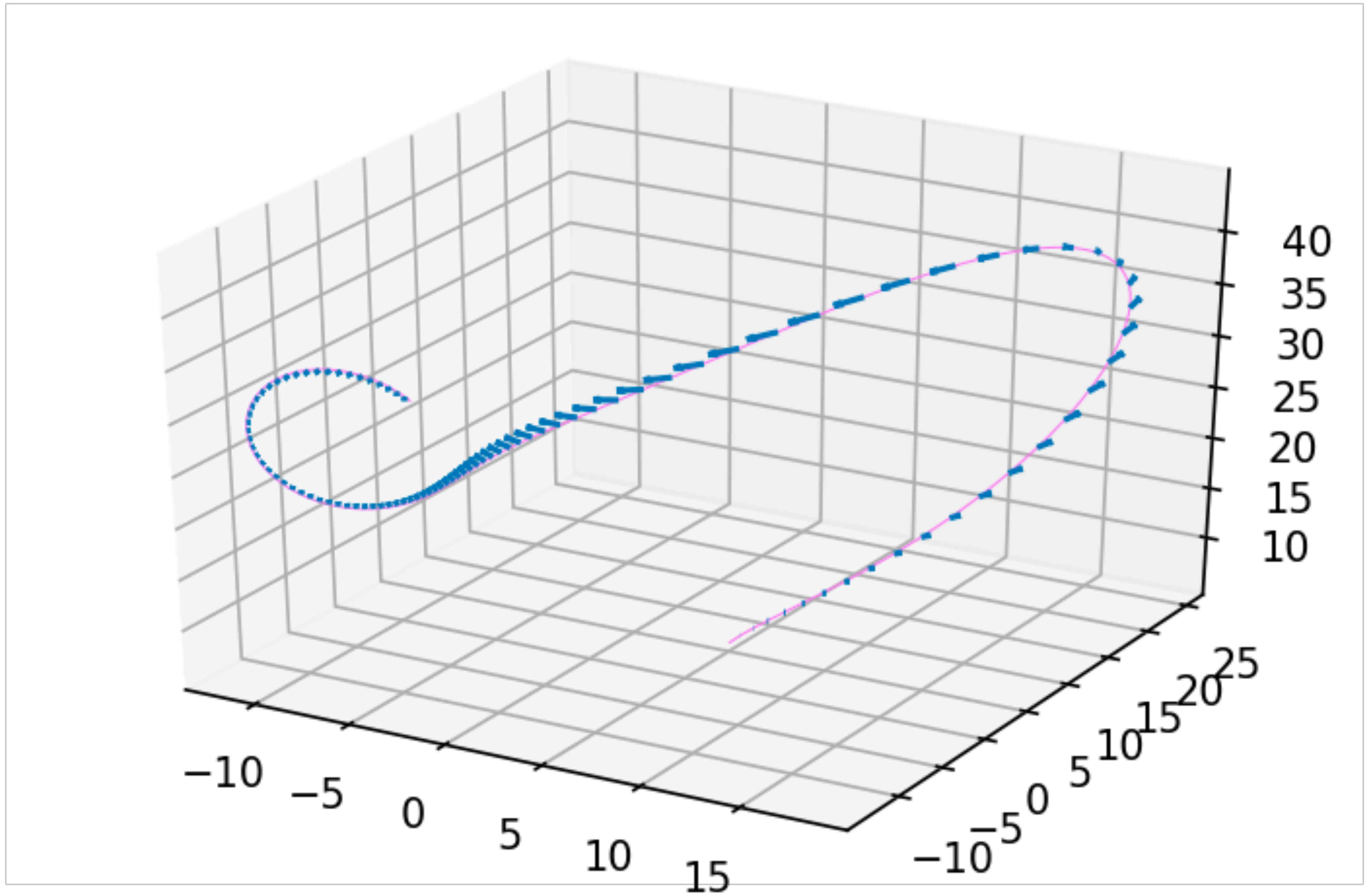
$$\partial_t y(t, x, y, z) = \begin{bmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ {\color{red} xy - \beta z} \end{bmatrix}$$



JVP on our example

- Or if we nudge the parameter σ

$$\partial_t y(t, x, y, z) = \begin{bmatrix} \color{red}{\sigma}(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{bmatrix}$$



JVP on our example

- Or β

$$\partial_t y(t, x, y, z) = \begin{bmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{bmatrix}$$