That's an excellent question. The code you have is a numerical implementation of a well-established mathematical procedure. The core idea behind calculating the largest Lyapunov exponent is to measure the average exponential rate at which two initially close trajectories diverge from each other.

The specific method implemented in the code is the **Benettin, et al. algorithm**. This iterative procedure is designed to overcome the problem of a perturbation vector growing too large, which would cause the two trajectories to no longer be considered "nearby."

Here is the step-by-step mathematical formalism of the calculation as it relates to your Python script:

1. The Core Principle

A positive Lyapunov exponent, λ, implies that the distance between two nearby trajectories, $\delta(t)$, grows exponentially over time. This relationship is expressed as:

$$||\delta(t)|| \approx ||\delta(0)||e^{\lambda t}$$

Rearranging for λ, we get:

$$\lambda = \lim_{t \to \infty} \frac{1}{t} \ln \left(\frac{||\delta(t)||}{||\delta(0)||} \right)$$

The numerical algorithm's job is to approximate this limit by repeatedly measuring the stretching of the perturbation vector and averaging the growth rate.

2. The Benettin, et al. Algorithm

The algorithm proceeds in a series of discrete steps, each of duration Δt. Let's denote the state of the system at time t_i as $\mathbf{x}_i$.

Step 1: Initialization
The process starts by defining two initial conditions that are infinitesimally close to each other. In the code, this corresponds to:

- A reference trajectory at `x0`: $\mathbf{x}_0$
- A perturbed trajectory at `x1`: $\mathbf{x}'_0 = \mathbf{x}_0 + \mathbf{d}_0$, where $\mathbf{d}_0$ is a small perturbation vector with magnitude $||\mathbf{d}_0|| = 10^{-9}$.

Step 2: Time Evolution (Integration)
For each time step i (from $i = 1$ to `n_steps`), the script integrates both trajectories forward over a small time interval `dt` (which is our Δt).
The `solve_ivp` function numerically solves the Lorenz system for both `x0` and `x1` to find their new positions at time $t_i = t_{i-1} + \Delta t$.

Step 3: Measuring Divergence
The new separation vector, $\mathbf{d}_i$, is calculated:

$$\mathbf{d}_i = \mathbf{x}'_i - \mathbf{x}_i$$

Its length, $||\mathbf{d}_i||$, which the code calls `d_new_norm`, is measured. The ratio of the new length to the old length, $||\mathbf{d}_i||/||\mathbf{d}_{i-1}||$, represents the stretching factor for this specific time step.

Step 4: Renormalization (The Crucial Step)

To prevent the perturbation from growing too large and leaving the "nearby" region of the phase space, the perturbation vector is scaled back to its original length, $||\mathbf{d}_0||$. This is done by calculating a new perturbation vector, $\mathbf{d}'_i$, with the same direction as $\mathbf{d}_i$ but with the initial magnitude:

$$\mathbf{d}'_i = \mathbf{d}_i \frac{||\mathbf{d}_0||}{||\mathbf{d}_i||}$$

In the code, this is implemented by the line `d0 = d_new * (d0_norm / d_new_norm)`. This new `d0` vector is then added to the new state `x0_new` to create the starting point for the next iteration.

Step 5: Accumulation of Growth

The logarithm of the stretching factor from Step 3, $\ln\left(\frac{||\mathbf{d}_i||}{||\mathbf{d}_{i-1}||}\right)$, is added to a running total (`lyapunov_sum`). The logarithm is used to convert exponential growth into a linear sum, which simplifies the averaging process later.

Step 6: Final Calculation

After completing all `n_steps`, the largest Lyapunov exponent is calculated by dividing the total sum of the logarithmic growth factors by the total time elapsed.

Step 4: Renormalization

This is the key step to keep the trajectories "nearby" and prevent the perturbation from becoming too large. The new perturbation vector is normalized back to the original magnitude, but its direction is preserved. This new, renormalized vector is $\mathbf{d}_1'$, which will be used for the next step of the iteration.

$$\mathbf{d}_1' = \mathbf{d}_1 \frac{||\mathbf{d}_0||}{||\mathbf{d}_1||}$$

The new perturbed state for the next step is then $\mathbf{x}_1' = \mathbf{x}_1 + \mathbf{d}_1'$.

Step 5: Accumulation

The logarithm of the growth factor is summed at each step to account for the multiplicative nature of the exponential growth. This process is repeated for N steps:

$$\text{Sum} = \sum_{i=1}^{N} \ln\left(\frac{||\mathbf{d}_i||}{||\mathbf{d}_{i-1}||}\right)$$

where $\mathbf{d}_i$ is the perturbation vector *before* renormalization at step i.

Step 6: Final Calculation

The largest Lyapunov exponent, λ_{max}, is the average of these logarithmic growth rates over the total time elapsed ($N \cdot \Delta t$):

$$\lambda_{max} = \frac{1}{N \cdot \Delta t} \sum_{i=1}^{N} \ln\left(\frac{||\mathbf{d}_i||}{||\mathbf{d}_{i-1}||}\right)$$

This final value provides a quantitative measure of the average rate of exponential divergence, which is the hallmark of chaos.