Created in COMSOL Multiphysics 6.1

Lorenz Attractor

Introduction

The Lorenz system is a system of ordinary differential equations (the Lorenz equations) first studied by Edward N. Lorenz. For certain parameter values and initial conditions, the system of ODEs has chaotic solutions. The solution is then a so-called strange attractor called the *Lorenz attractor*, discovered by Lorenz in 1962.

Model Definition

The Lorenz equations were developed as a simplified mathematical model for atmospheric convection. They are a system of three coupled ODEs with three states (degrees of freedom), u, v, and w:

$$\begin{cases} \dfrac{du}{dt} = a(v-u) \\ \dfrac{dv}{dt} = u(c-w)-v \\ \dfrac{dw}{dt} = uv-bw \end{cases} \tag{1}$$

Here, the parameters a, b, and c are generally positive scalar numbers. Not all solutions are chaotic, but Lorenz found that the values 10, 8/3, and 28, respectively, give a Lorenz system with a chaotic behavior.

TABLE 1: PARAMETER VALUES FOR A LORENZ ATTRACTOR.

PARAMETER	VALUE
a	10
b	8/3
c	28

For the parameter values in Table 1, the solution to this system of equations approaches a geometrical object in the phase space defined by the solution coordinates (u, v, w) called a strange attractor or the *Lorenz attractor*. It can be shown that this object is not of a standard geometrical topology, like a disk (with dimension 2) or a line (with dimension 1), but another strange topology with a fractal dimension larger than 2. Moreover, almost all perturbations to the solution grow exponentially fast with time. As an illustration of this behavior, the model starts from an initial solution close to the attractor and studies how a very small perturbation to this initial data grows. More precisely, the initial solution used here is $(u, v, w) = (-10, -4.45, 35.1)$, which in a second perturbed case is changed to $(u,$

v, w) = (−10+pert, −4.45+pert, 35.1+pert), where pert = 10^{-11}. To ascertain that the difference between the unperturbed and perturbed problems is larger than the numerical errors, the tolerances used when solving the model are very strict. A default absolute tolerance factor of 0.1 is used, meaning that the absolute tolerance is 0.1 times the relative tolerance TOL used in the model.

Results and Discussion

First consider the numerical errors. Table 2 shows the results for the solution (u, v, w) for two unperturbed simulations with TOL = 10^{-15} and TOL = 10^{-16}, respectively, at the times $t = 20$, 25, 30, and 35. It is clear that for $t = 20$ all 5 displayed digits of the result agree, a strong indication of a highly accurate solution. In contrast, already at $t = 30$ only two digits remain correct. Both the numerical errors and the perturbations will grow over time; eventually no digits will be correct with this method and these tolerances.

TABLE 2: UNPERTURBED SOLUTIONS FOR DIFFERENT TIMES.

TOL	t	u	v	w
10^{-15}	20	1.0748	-2.9811	26.380
10^{-15}	25	7.9602	13.758	14.996
10^{-15}	30	3.7537	4.1334	20.640
10^{-15}	35	-8.1947	-4.9358	30.479
10^{-16}	20	1.0748	-2.9811	26.380
10^{-16}	25	7.9595	13.757	14.994
10^{-16}	30	3.7181	4.1095	20.566
10^{-16}	35	3.5213	1.4743	24.833

Figure 1 visualizes how the difference between the unperturbed and perturbed problems grows with time. The plot shows the absolute difference between the two solutions for the most accurate (tight) tolerance value as functions of time together with an estimate for the average growth rate given by $e^{\lambda t}$, where $\lambda = 0.9$, the maximal *Lyapunov exponent* for this model according to the literature. Note that the plot uses a vertical axis with a logarithmic scale. As the plot shows, the growth of the difference is close to the correct value until it reaches O(1), after which it does not grow much further. The reason is that all solutions stay close to the attractor, so the difference is bounded.

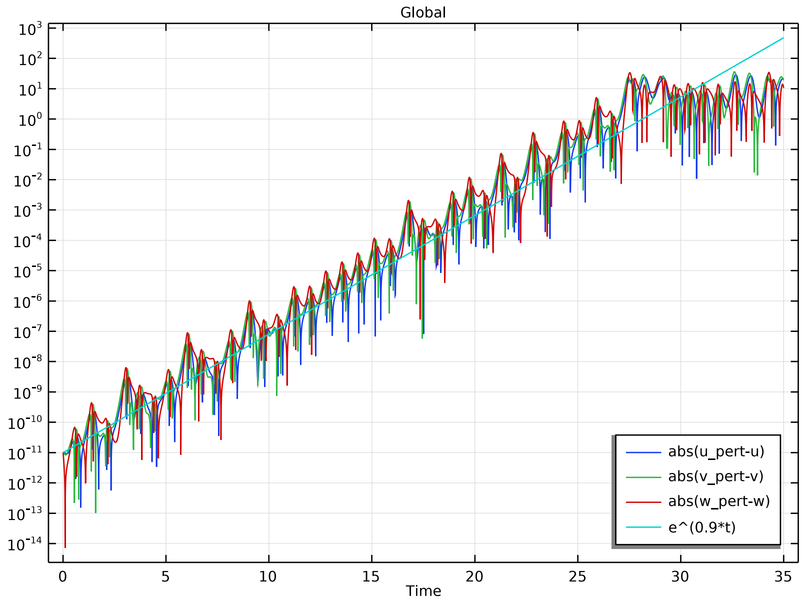

Figure 1: Differences between the unperturbed and perturbed solutions as functions of time. The straight line shows the average growth rate estimated as $\exp(\lambda t)$, *where* $\lambda = 0.9$.

With the chosen set of parameter values, the Lorenz system behaves as a Lorenz attractor. The point trajectories plot in Figure 2 of the phase space shows the “butterfly” or “figure eight” characteristic of this attractor.

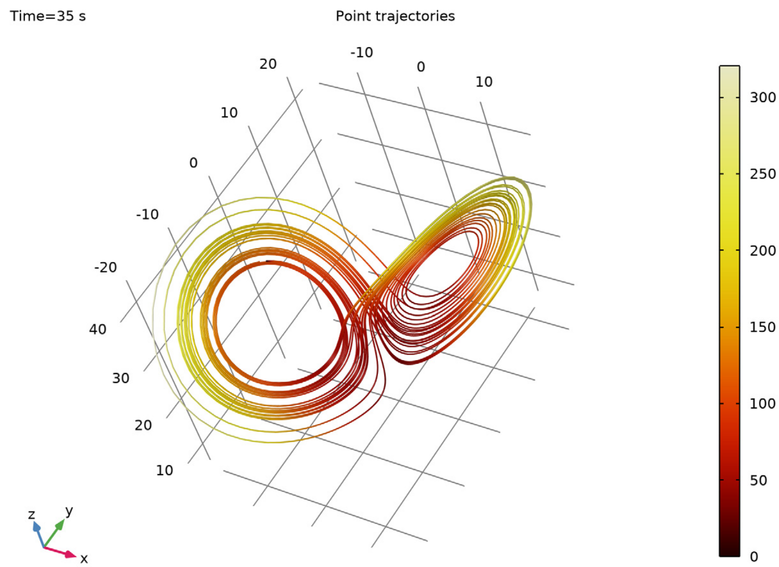

Figure 2: The typical pattern for a Lorenz attractor visualized using a Point Trajectories plot of the phase space.

Notes About the COMSOL Implementation

Some key points for settings up, solving, and visualizing the results for a Lorenz attractor in COMSOL Multiphysics:

- Using the Global ODEs and DAEs interface, it is straightforward to enter the ODEs that define the Lorenz system in Equation 1.
- For a fast solution to a set of coupled ODEs, the explicit Dormand-Prince time-stepping method is efficient. By adjusting the relative tolerance, you can control the accuracy of the solution.
- The `withsol()` operator and a Global Plot is used to generate the plot in Figure 1.
- A Point Trajectories plot in the phase space for the states show the typical “butterfly” or “figure eight” pattern shown in Figure 2.

Reference

1. http://en.wikipedia.org/wiki/Lorenz_system

Application Library path: `COMSOL_Multiphysics/Equation_Based/lorenz_attractor`

Modeling Instructions

From the **File** menu, choose **New**.

NEW

In the **New** window, click **Model Wizard**.

MODEL WIZARD

1. In the **Model Wizard** window, click **3D**.
2. In the **Select Physics** tree, select **Mathematics>ODE and DAE Interfaces>Global ODEs and DAEs (ge)**.
3. Click **Add**.
4. Click **Study**.
5. In the **Select Study** tree, select **General Studies>Time Dependent**.
6. Click **Done**.

ROOT

The ODEs here are dimensionless, so do not use any unit system.

1. In the **Model Builder** window, click the root node.
2. In the root node's **Settings** window, locate the **Unit System** section.
3. From the **Unit system** list, choose **None**.

GLOBAL DEFINITIONS

The parameters `a`, `b`, and `c` are the system parameters for the Lorenz system of ODEs. The default values of 10, 8/3, and 28, respectively, are the values that Lorenz used.

Parameters 1

1. In the **Model Builder** window, under **Global Definitions** click **Parameters 1**.

2 In the **Settings** window for **Parameters**, locate the **Parameters** section.

3 In the table, enter the following settings:

Name	Expression	Value	Description
a	10	10	System parameter
b	8/3	2.6667	System parameter
c	28	28	System parameter
pert	0	0	Initial data perturbation
TOL	1e-6	1E-6	Solver Tolerance
Tfinal	35	35	End time for simulation

GLOBAL ODES AND DAES (GE)

Use a Global ODEs and DAEs interface to enter the three ODEs that define a Lorenz system. Choose initial data close to the strange attractor to study its sensitivity. The perturbation is added to all the components.

Global Equations 1

1 In the **Model Builder** window, under **Component 1 (comp1)>Global ODEs and DAEs (ge)** click **Global Equations 1**.

2 In the **Settings** window for **Global Equations**, locate the **Global Equations** section.

3 In the table, enter the following settings:

Name	f(u,ut,utt,t)	Initial value (u_0)	Initial value (u_t0)
u	ut-a*(v-u)	-10+pert	0
v	vt-c*u+v+u*w	-4.45+pert	0
w	wt-u*v+b*w	35.1+pert	0

STUDY 1

For an accurate solution, use the explicit Dormand-Prince 5 time-stepping method. The tolerance for this method will be systematically changed.

Solution 1 (sol1)

In the **Study** toolbar, click **Show Default Solver**.

Step 1: Time Dependent

1 In the **Model Builder** window, under **Study 1** click **Step 1: Time Dependent**.

2 In the **Settings** window for **Time Dependent**, locate the **Study Settings** section.

3 Click **Range**.

4 In the **Range** dialog box, type `0.01` in the **Step** text field.

5 In the **Stop** text field, type `Tfinal`.

6 Click **Replace**.

7 In the **Settings** window for **Time Dependent**, locate the **Study Settings** section.

8 From the **Tolerance** list, choose **User controlled**.

9 In the **Relative tolerance** text field, type `TOL`.

Solution 1 (sol1)

1 In the **Model Builder** window, expand the **Study 1>Solver Configurations>Solution 1 (sol1)** node, then click **Time-Dependent Solver 1**.

2 In the **Settings** window for **Time-Dependent Solver**, click to expand the **Time Stepping** section.

3 From the **Solver type** list, choose **Explicit**.

4 From the **Method** list, choose **Runge–Kutta**.

5 From the **Runge–Kutta method** list, choose **Dormand–Prince 5**.

6 In the **Model Builder** window, expand the **Study 1>Solver Configurations>Solution 1 (sol1)>Time-Dependent Solver 1** node, then click **Direct**.

7 In the **Settings** window for **Direct**, locate the **General** section.

8 From the **Solver** list, choose **PARDISO**.

This solver is more efficient for small problems with a large number of time steps.

Parametric Sweep

Use a **Parametric Sweep** to study the effect of a small perturbation to the initial data and to changes in a very strict solver tolerance.

1 In the **Study** toolbar, click **Parametric Sweep**.

2 In the **Settings** window for **Parametric Sweep**, locate the **Study Settings** section.

3 Click **Add** twice.

4 In the table, enter the following settings:

Parameter name	Parameter value list
pert (Initial data perturbation)	1e-11 0
TOL (Solver Tolerance)	1e-15 1e-16

5 From the **Sweep type** list, choose **All combinations**.

6 In the **Study** toolbar, click **Compute**.

RESULTS

1D Plot Group 1

Use a **Global** plot to visualize how the solution changes due to the small perturbation and add an exponential growth corresponding to the largest Lyapunov exponent 0.9 reported in the literature for comparison. Note how the growth stops when the deviation is of O(1); this is caused by the attraction effect. The deviations cannot be larger, since all solutions are close to the attractor.

1 In the **Settings** window for **1D Plot Group**, locate the **Data** section.

2 From the **Parameter selection (pert)** list, choose **From list**.

3 In the **Parameter values (pert)** list, select **1E-11**.

4 From the **Parameter selection (TOL)** list, choose **From list**.

5 In the **Parameter values (TOL)** list, select **1E-16**.

Global 1

1 In the **Model Builder** window, expand the **1D Plot Group 1** node, then click **Global 1**.

2 In the **Settings** window for **Global**, locate the **y-Axis Data** section.

3 In the table, enter the following settings:

Expression	Description
abs(u-withsol('sol2',u,setval(pert,0), setval(TOL,TOL),setval(t,t)))	abs(u_pert-u)
abs(v-withsol('sol2',v,setval(pert,0), setval(TOL,TOL),setval(t,t)))	abs(v_pert-v)
abs(w-withsol('sol2',w,setval(pert,0), setval(TOL,TOL),setval(t,t)))	abs(w_pert-w)
pert*exp(0.9*t)	e^(0.9*t)

4 Click to expand the **Legends** section. From the **Legends** list, choose **Manual**.

5 In the table, enter the following settings:

Legends
abs(u_pert-u)
abs(v_pert-v)
abs(w_pert-w)
e^(0.9*t)

6 Locate the **x-Axis Data** section. From the **Axis source data** list, choose **Time**.

1D Plot Group 1

1 In the **Model Builder** window, click **1D Plot Group 1**.

2 In the **1D Plot Group 1** toolbar, click **Plot**.

3 Click the **y-Axis Log Scale** button in the **Graphics** toolbar.

4 In the **Settings** window for **1D Plot Group**, locate the **Legend** section.

5 From the **Position** list, choose **Lower right**.

Compare with the plot in Figure 1.

Use a **Point Trajectories** plot to visualize the solution to the Lorenz system as point trajectories as shown in Figure 2.

3D Plot Group 2

In the **Home** toolbar, click **Add Plot Group** and choose **3D Plot Group**.

Point Trajectories 1

1 In the **3D Plot Group 2** toolbar, click **More Plots** and choose **Point Trajectories**.

2 In the **Settings** window for **Point Trajectories**, locate the **Trajectory Data** section.

3 In the **X-expression** text field, type `u`.

4 In the **Y-expression** text field, type `v`.

5 In the **Z-expression** text field, type `w`.

6 Locate the **Coloring and Style** section. Find the **Line style** subsection. From the **Type** list, choose **Tube**.

7 Select the **Radius scale factor** check box. In the associated text field, type `0.1`.

Color Expression 1

1 Right-click **Point Trajectories 1** and choose **Color Expression**.

2 In the **Settings** window for **Color Expression**, locate the **Coloring and Style** section.

3 Click **Change Color Table**.

4 In the **Color Table** dialog box, select **Thermal>ThermalDark** in the tree.

5 Click **OK**.

6 In the **Settings** window for **Color Expression**, locate the **Expression** section.

7 In the **Expression** text field, type `sqrt(ut^2+vt^2+wt^2)`.

8 In the **3D Plot Group 2** toolbar, click **Plot**.

By clicking in the **Graphics** window and dragging you can explore the attractor from different viewpoints.

The following steps generate an animation of the Lorenz attractor.

Animation 1

1 In the **Results** toolbar, click **Animation** and choose **Player**.

2 In the **Settings** window for **Animation**, click **Show Frame**.

3 Locate the **Scene** section. From the **Subject** list, choose **3D Plot Group 2**.

4 Locate the **Frames** section. In the **Number of frames** text field, type `200`.

5 Click the **Play** button in the **Graphics** toolbar.

Next, evaluate the unperturbed solution at a selection of times and study how accurate it is by comparing the values for the two tolerances used. Note how several digits are the same for moderate time values, while no digits are the same at the final time.

State variables

1 In the **Model Builder** window, expand the **Results>Derived Values** node, then click **Global Evaluation 1**.

2 In the **Settings** window for **Global Evaluation**, locate the **Data** section.

3 From the **Parameter selection (pert)** list, choose **From list**.

4 In the **Parameter values (pert)** list, select **0**.

5 From the **Time selection** list, choose **Interpolated**.

6 In the **Times (s)** text field, type `20,25,30,35`.

7 Click **Evaluate**.

The results should agree with those displayed in Table 2.

8 In the **Label** text field, type `State variables`.