Visualizing Systems of Differential Equations in 3-Dimensions

Yuanxun Bill Bao, Michael Monagan

ybb@sfu.ca, mmonagan@cecm.sfu.ca

Introduction

Systems of differential equations play an essential role in the study of dynamical systems and mathematical modeling. An important tool for studying systems of differential equations is the *direction field*. The direction field gives a graphical representation of the dynamics of the system without solving it analytically or numerically.

Consider the 3-Dimensional autonomous system:

$$x'(t) = f(x, y, z)$$
$$y'(t) = g(x, y, z)$$
$$z'(t) = h(x, y, z)$$

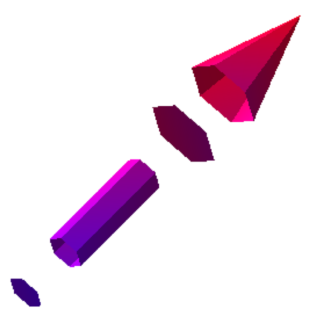
where $x, y, z \in \mathbb{R}$.

Given one point (x_i, y_i, z_i) in the domain, we can draw an arrow at this point in the direction $\langle f(x_i, y_i, z_i), g(x_i, y_i, z_i), h(x_i, y_i, z_i)\rangle$. Such an arrow is tangent to the solution curve at the point (x_i, y_i, z_i). The collection of all these arrows is often called the *direction field*. By following the arrows in the direction field, we can get a good approximation to the Initial Value Problem(s) as well as a qualitative visualization of the behavior of the system. The **DEplot** command in Maple can create a direction field of a 2-dimensional autonomous system with additional options if specified. In this poster, we extend this functionality to 3-dimensions and enhance the visualization by using animations with a computer demo.

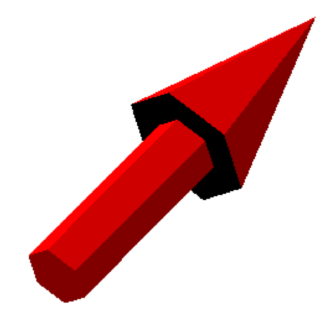
Design of Direction Vectors

The key elements in the construction of a direction field are designing a set of 3-D arrow objects that represent the direction field. The cost of computing these arrows needs to be carefully considered. As a result, we present the following 3 designs.

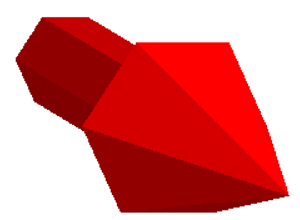
- **Hexagon-based Arrows**
  This type of arrow objects consists of a cylinder with a regular hexagon base as the arrow body and a hexagonal pyramid as the arrow head.



Decomposition of Hexagon-based Arrow

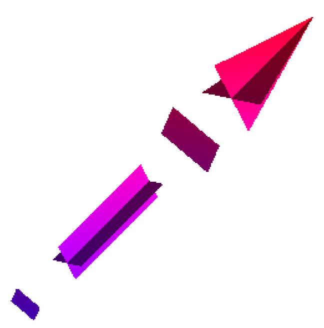


Hexagon-based Arrow

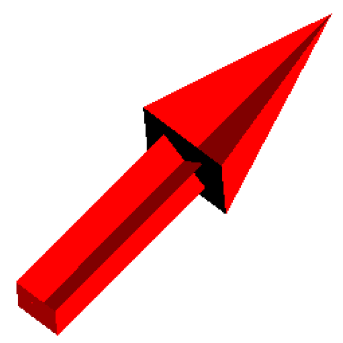


Hexagon-based Arrow

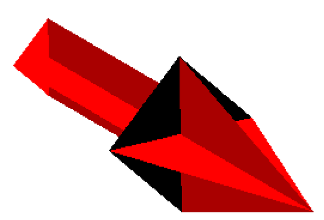
- **Square-based Arrows**
  We take rectangles and triangles intersecting orthogonally and attach them to a square base. The cost of computing arrows is reduced without losing too much visually.



Decomposition of Square-based Arrow

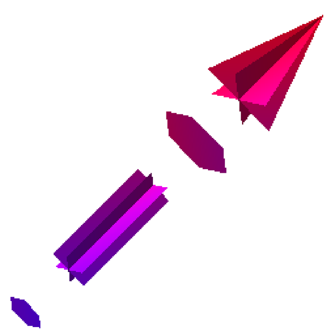


Square-based Arrow

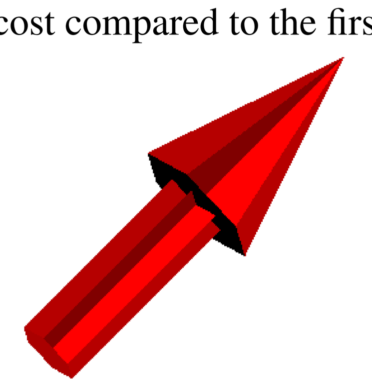


Square-based Arrow

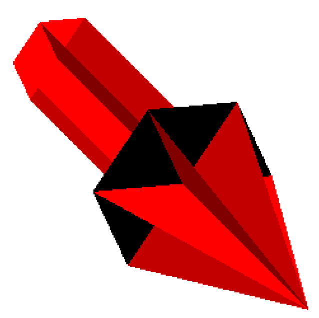
- **Hybrid Arrows**
  Combining the advantages from the above 2 types of arrow objects, we can easily design a hexagon-based arrow with a lower cost compared to the first type.



Decomposition of Hybrid Arrow



Hybrid Arrow



Hybrid Arrow

As mentioned earlier, costs of computing arrow objects require careful considerations. Below are 2 tables of costs comparison of different arrow objects.

Arrow Type	*# of Triangles*	*# of Rectangles*	*# of Hexagons*	*Total # of Points*
Hexagon-based	6	6	2	54
Square-based	2	4	0	22
Hybrid	3	3	2	33

Table 1: Costs Comparison of 3 Types of Arrow Objects

# of Arrows	*Square − based*	*Hybrid*	*Hexagon − based*
100	0.016	0.031	0.047
1000	0.266	0.375	0.453
5000	1.687	1.766	2.594
10000	3.078	4.172	5.125
50000	15.875	20.047	25.828
100000	31.844	40.562	51.641

Table 2: Timings (in CPU seconds) for generating arrows as Maple POLYGON objects

Direction Fields

- Consider the region $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ in 3-dimensional Cartesian coordinates.
- Let D be the set of all points in the region where a direction vector needs to be computed at that point. D can be constructed by either using a regular grid in 3-D or by randomly generating points in $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ in 3-D. Visually, random arrows look better, as shown in the example.
- For every point (x_i, y_i, z_i) in D, the corresponding direction vector is computed as

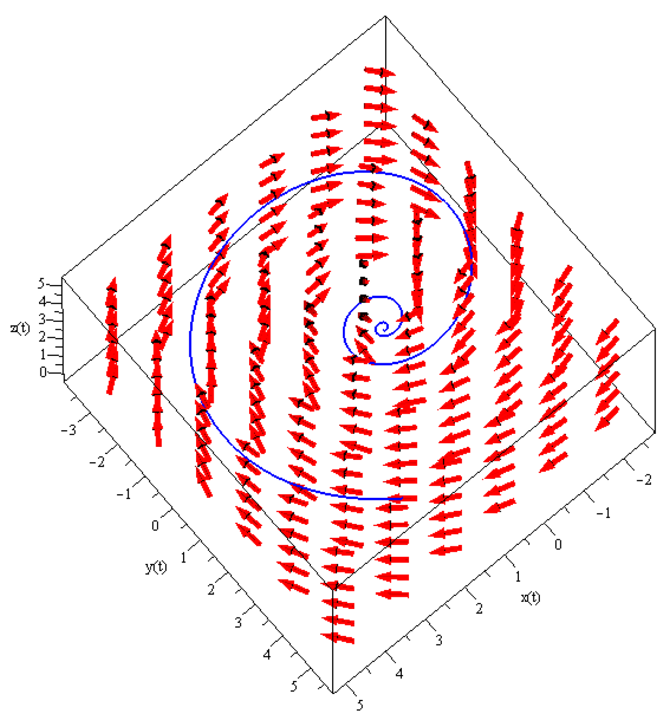
$$\langle f(x_i, y_i, z_i), g(x_i, y_i, z_i), h(x_i, y_i, z_i)\rangle$$

  In some systems, the magnitudes of direction vectors vary greatly. To avoid this, we scale the vectors to the same length.
- Repeatedly call the function that computes the direction vector at every point in D.

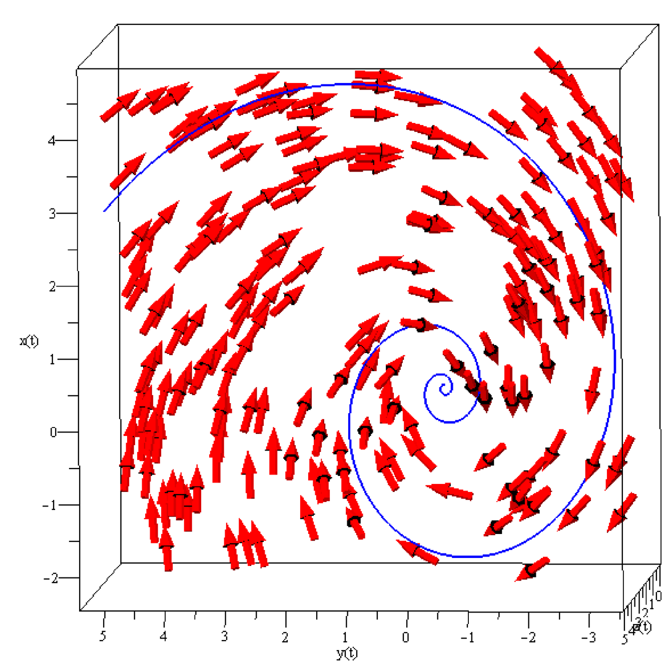
Example 1: Consider the system

$$x'(t) = -\tfrac{1}{4}x(t) + y(t)$$
$$y'(t) = -x(t) - \tfrac{1}{4}y(t)$$
$$z'(t) = -\tfrac{1}{4}z(t)$$
$$x(0) = 3, y(0) = 5, z(0) = 5$$

One can see that this system has a fixed point at $(0, 0, 0)$. Now let's try looking at the direction field.



Direction Field with Square-based Arrows on 3D grid

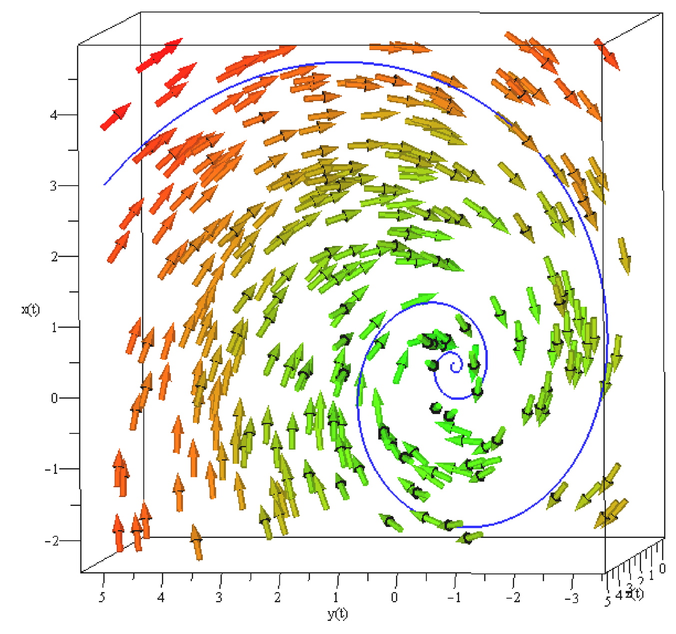


Direction Field with Random Hexagon-based Arrows

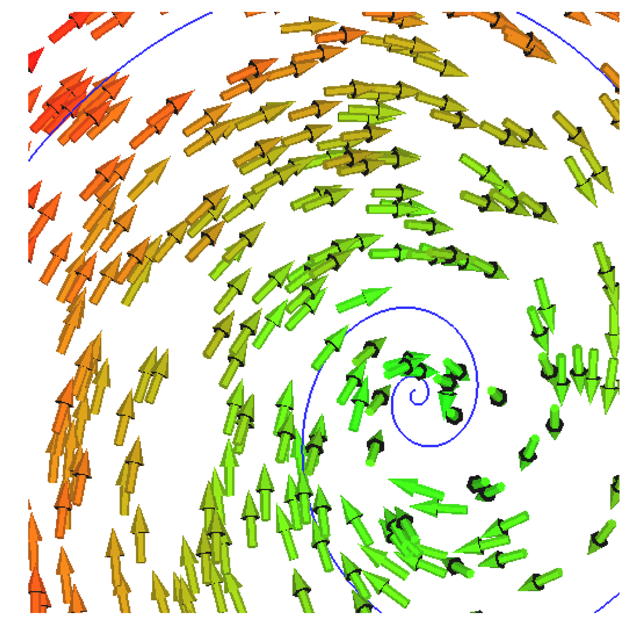
Speed of Systems

The drawback of scaling the magnitude of direction vectors to the same length is that we lose all information about the speed of the system. Speed information is important since it tells how "fast" at each point the system is moving. Furthermore, speed information can assist us in spotting fixed points.

One solution is to apply a color-scale to represent different speeds of direction vectors. For the system in Example 1, we will use red for fast moving vectors and green for slow moving vectors. The light green color indicates that the system slows down and approaches to a fixed point.



Direction Field with Random Hexagon-based Arrows



Direction Field with Random Hexagon-based Arrows (ZOOM-IN)

Applications to Dynamical Systems

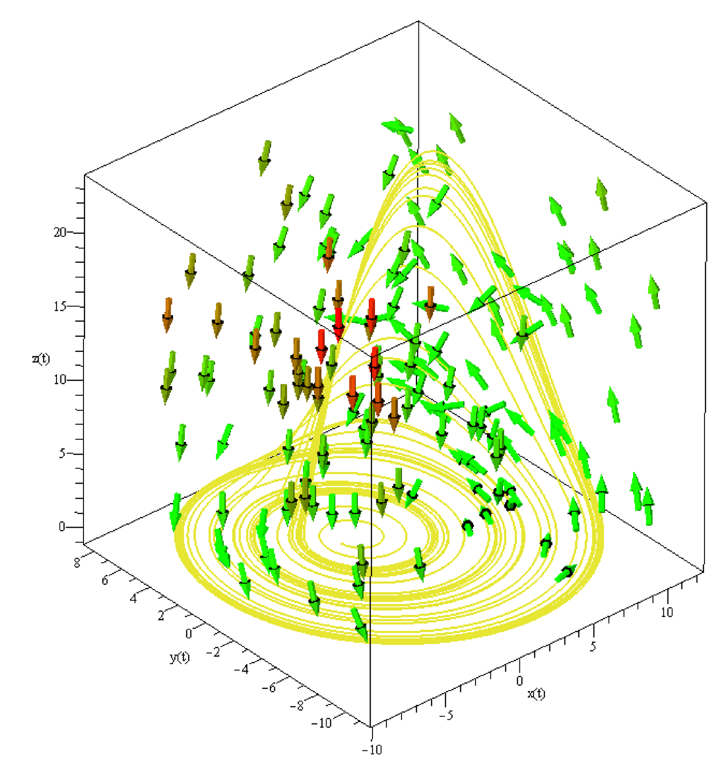
Many interesting systems are nonlinear and therefore are very hard to solve analytically. Examples include the Rössler System and the Lorenz System (the latter is better known as the "Butterfly Model").

Rössler System

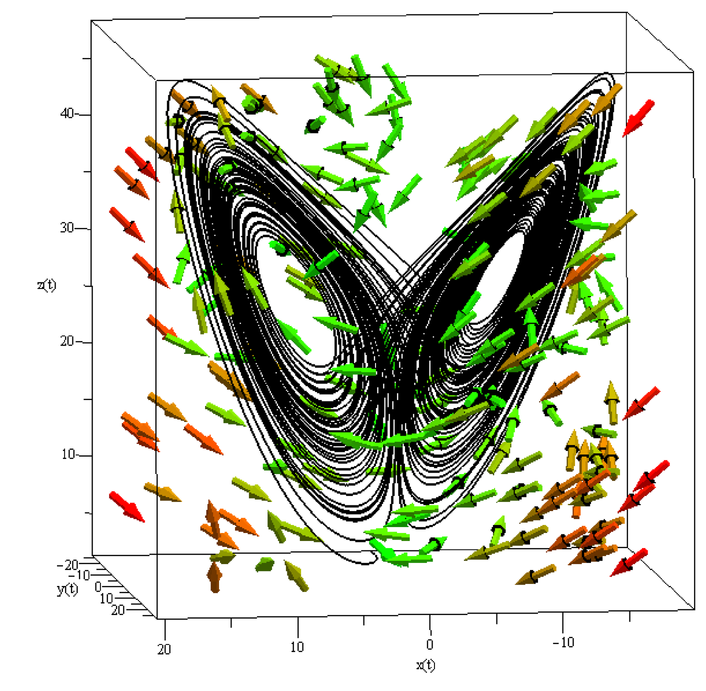
$$x'(t) = -y(t) - z(t)$$
$$y'(t) = x(t) + a \cdot y(t)$$
$$z'(t) = b + z(t) \cdot (x(t) - c)$$
$$a = 0.2, b = 0.2, c = 5.7$$
$$x(0) = 4, y(0) = 0, z(0) = 0$$

Lorenz System

$$x'(t) = \sigma \cdot (y(t) - x(t))$$
$$y'(t) = x(t) \cdot (\rho - z(t)) - y(t)$$
$$z'(t) = x(t) \cdot y(t) - \beta \cdot z(t)$$
$$\sigma = 10, \rho = 28, \beta = \tfrac{8}{3}$$
$$x(0) = 2, y(0) = 0, z(0) = 4$$



Rössler System



Lorenz System

References

[1] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron and P. DeMarco, *Maple 10 Advanced Programming Guide*. Maplesoft, Waterloo, Ontario, Canada, 2005.

[2] M. Ebrahimi, M. Monagan, New Options to Visualize Systems of Differential Equations in Maple, *Proceedings of the 2005 Maple Conference*. pp.260-271, Maplesoft, 2005