Maplesoft
Mathematics • Modeling • Simulation

TIPS & TECHNIQUES
www.maplesoft.com

Classroom Tips and Techniques: Plotting Curves Defined Parametrically

Robert J. Lopez
Emeritus Professor of Mathematics and Maple Fellow
Maplesoft

Introduction

If the vector representation of a curve is considered to be a parametric representation, then (in Maple) there are at least three ways to graph a curve defined parametrically. A purely parametric representation (not using vectors) in 2D is graphed with the **plot** command; in 3D, with the **spacecurve** command. A vector representation in 2D is graphed with the **SpaceCurve**, or **PlotPositionVector** commands (all in the *VectorCalculus* packages); in 3D, with the **spacecurve** and the *VectorCalculus* commands. An integral curve of a vector field can be drawn with the **FlowLine** command in the *Student*[*VectorCalculus*] package.

In this article, we compare the options for graphing curves given parametrically in two or three dimensions.

Initializations

```
restart;
with(plots) :
with(DEtools) :
with(VectorCalculus ) :
BasisFormat(false ) :
interface (typesetting = extended ) :
```

The *plot* and *spacecurve* Commands

Figure 1 uses the **plot** command to graph the parametrically-given curve $x = \cos(t), y = \sin(t)$.

```
plot([cos(t), sin(t), t = 0 ..2 π],
    labels = [x, y])
```

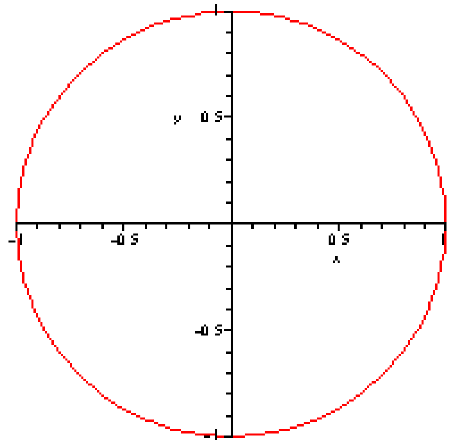


Figure 1 Parametric plot in 2D

The syntax for the **plot** command is a list of the form $[x(t), y(t), t = a..b]$, where, of course, t can be any parameter. The list must have these three elements, and the range must be inside the list.

Figure 2 uses the **spacecurve** command to graph one loop of the helix $x = \cos(t), y = \sin(t), z = t$.

$$spacecurve\left([\cos(t), \sin(t), t], t = 0..2\,\pi, color = red, axes = box, labels = [x, y, z]\right)$$

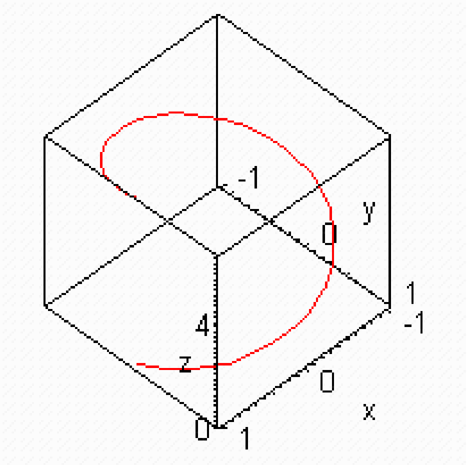


Figure 2 Parametric plot of a helix in 3D

The syntax for the **spacecurve** command is flexible. The list $[x(t), y(t), z(t)]$ may or may not include the range; the list of three functions can even be a vector!

In 2D, the **plot** command cannot provide a parametric plot of a curve given parametrically as a vector. Thus, a number of users have written their own parametric plotting functions that admit a vector in both two and three dimensions. Perhaps this observation led to the functionalities in the **PlotPositionVector**, **SpaceCurve**, and **FlowLine** commands in the *VectorCalculus* packages.

The *VectorCalculus* Commands

The original implementation of the *VectorCalculus* packages was based on the constructs of the *free vector* and the **VectorField**. In Cartesian coordinates, the point $\left(x_1, x_2, \ldots, x_n\right)$ is identified with the "free" vector $\langle x_1, x_2, \ldots, x_n\rangle$ whose tail is at the origin and whose head is at the point. This works fine in Cartesian coordinates where the basis vectors are constant. (It does not work well in nonCartesian coordinates where the basis vectors change from point to point.)

Figure 3 uses the **SpaceCurve** command to plot the (Cartesian) free vector

$V := \langle \cos(t), \sin(t) \rangle$

$$V := \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$$

$SpaceCurve\left(V, t = 0 .. 2\,\pi\right)$

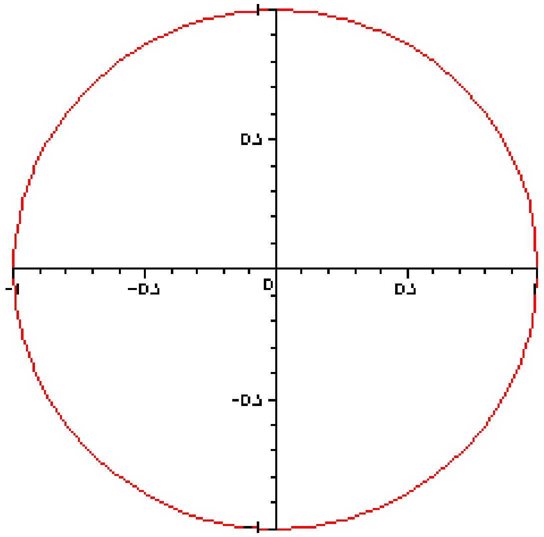


Figure 3 2D curve defined parametrically by a free vector, graphed with the **SpaceCurve** command

The **SpaceCurve** command could equally well have been applied to the free vector $\langle \cos(t), \sin(t), t \rangle$, thereby producing the graph of a helix. Thus, the distinction between dimensions has been eliminated.

Updates to the *VectorCalculus* packages introduced the construct of the **PositionVector** $\mathbf{R} = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k}$ for defining curves (and surfaces). The curve graphed in Figures 1 and 3 would be represented by

$\mathbf{R} := \mathit{PositionVector}([\cos(t), \sin(t)])$

$$R := \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix}$$

and would be graphed with the **PlotPositionVector** command as per Figure 4.

$\mathit{PlotPositionVector}\big(\mathbf{R}, t = 0..2\,\pi,$
$\quad \mathit{curveoptions} = [\mathit{color} = \mathit{red}]\big)$

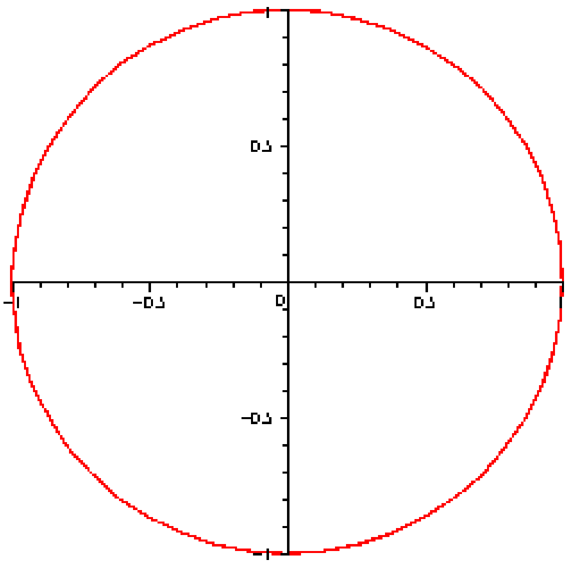


Figure 4 PositionVector graphed by **PlotPositionVector** command

An option to the **PlotPositionVector** command provides for plotting the arrows of a vector field, and other options provide for plotting the arrows of the tangent, principal normal, and derivative fields for the position vector. This is illustrated in Figure 5 where tangent vectors are shown in black, and normal vectors are shown in green.

$$
\mathit{PlotPositionVector}\left(\mathbf{R}, t = 0\,..2\,\pi,\ \mathit{curveoptions} = [\mathit{color} = \mathit{red}],\ \mathit{tangent} = \mathit{true}, \mathit{normal} = \mathit{true}\right)
$$

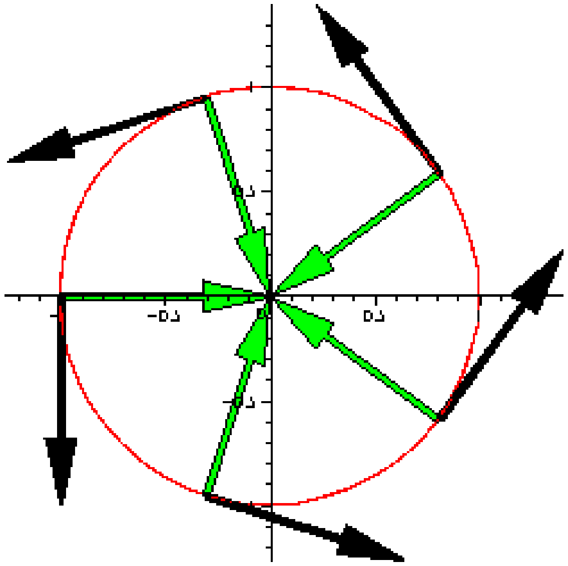


Figure 5 Drawing tangent and principal normal vectors with the **PlotPositionVector** command

The **VectorField** command defines a vector at each point of $\mathbb{R}^n$. Figure 6 superimposes ten vectors of the vector field

$$
\mathbf{F} := \mathit{VectorField}\left(\langle x^2 + y, x - y\rangle, \mathit{cartesian}[x, y]\right)
$$

$$
F := \begin{bmatrix} x^2 + y \\ x - y \end{bmatrix}
$$

on the curve **R** in Figure 4.

$$
\mathit{PlotPositionVector}\left(\mathbf{R}, t = 0\,..2\,\pi,\ \mathit{curveoptions} = [\mathit{color} = \mathit{red}, \mathit{scaling} = \mathit{constrained}],\ \mathit{vectorfield} = \mathbf{F},\ \mathit{vectorfieldoptions} = [\mathit{color} = \mathit{green}],\ \mathit{vectornum} = 10\right)
$$

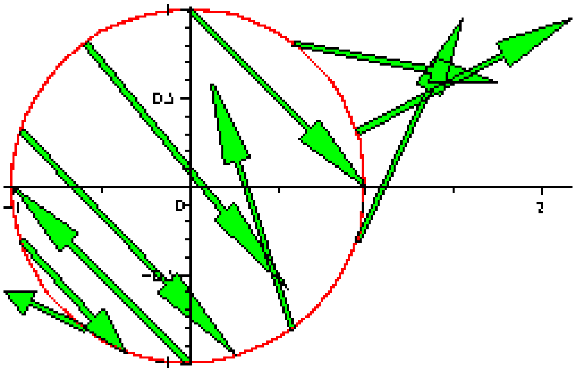


Figure 6 Arrows of the vector field **F** evaluated along the curve **R**

The integral of a vector field produces curves called *flow lines* along which the vectors of the field are tangent. (For example, the flow lines of an electric field are called *field lines*.) The **FlowLine** command in the *Student*[*VectorCalculus*] package will draw the flow line emanating from a given point, as illustrated for the field **F** and Figure 7.

```
Student[VectorCalculus]:-FlowLine(F, ⟨
    −3, 3⟩, fieldoptions = [grid = [10,
    10], arrows = medium, fieldstrength
    = fixed])
```

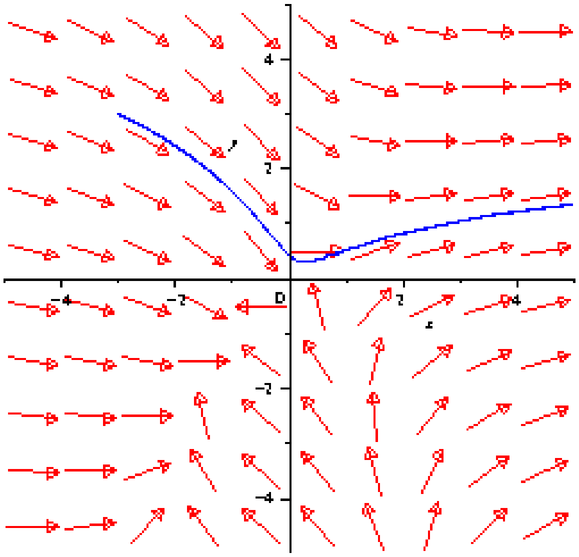


Figure 7 The flow line emanating from $(1, 1)$ for the field **F**

Unfortunately, it is not possible to ask the **FlowLine** command for more than one flow line in Maple 12. We hope to see this shortcoming corrected in a future release of Maple. The alternative is to write the differential equations

$$DEx := \dot{x}(t) = F_1\Big|_{x = x(t),\, y = y(t)};$$
$$DEy := \dot{y}(t) = F_2\Big|_{x = x(t),\, y = y(t)}$$

$$DEx := \dot{x}(t) = x(t)^2 + y(t)$$
$$DEy := \dot{y}(t) = x(t) - y(t)$$

and to integrate them with the **DEplot** command from the *DEtools* package. The result for the initial points

$$inits := [seq([0, 1, k], k = -4..4)]$$
$$inits := [[0, 1, -4], [0, 1, -3], [0, 1, -2], [0, 1, -1], [0, 1, 0], [0, 1, 1], [0, 1, 2], [0, 1, 3], [0, 1, 4]]$$

appears in Figure 8.

```
DEplot([DEx, DEy], [x(t), y(t)], t = 0 ..3,
    x =−2 ..5, y =−4 ..4, inits, dirgrid
     = [10, 10], arrows = medium, color
     = red, linecolor = black, thickness
     = 2)
```

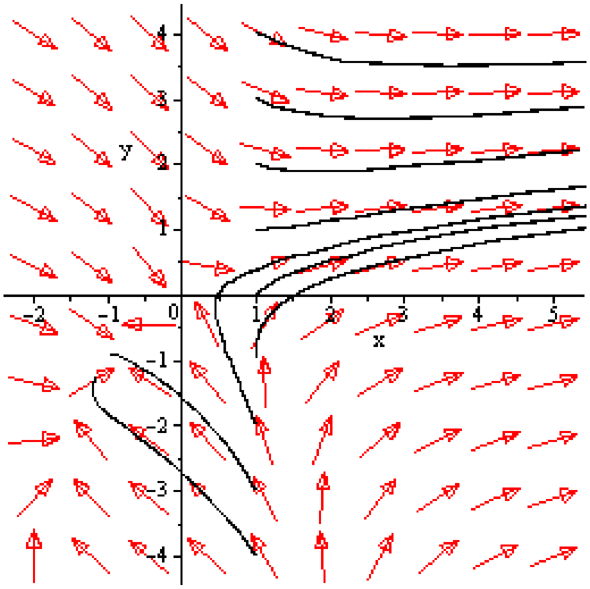


Figure 8 Integration of the field **F** by means of the **DEplot** command

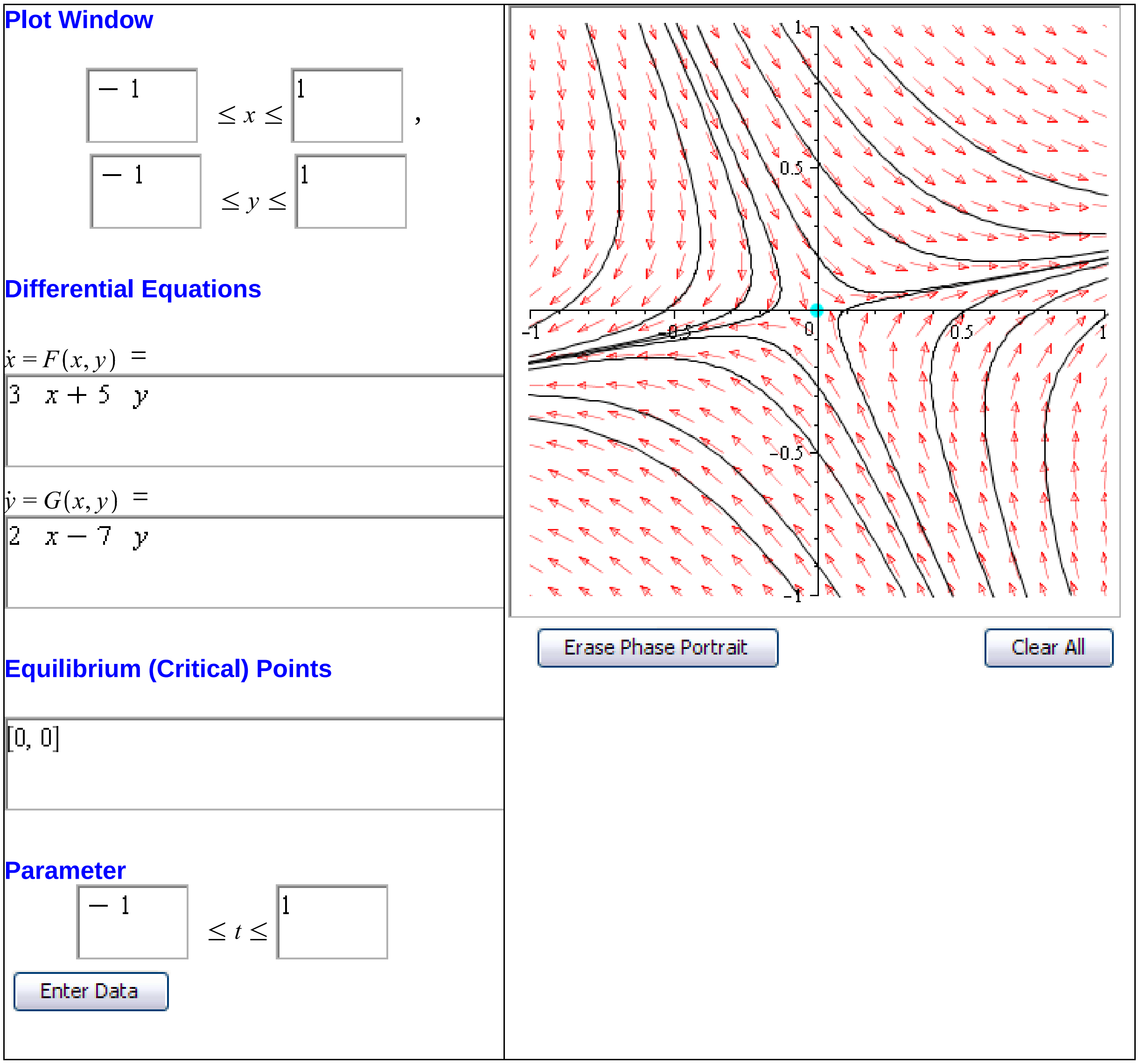
Task Templates

Maple 12 provides a task template for interactively generating orbits in the phase plane of the autonomous system

$$\dot{x}(t) = f(x(t), y(t))$$
$$\dot{y}(t) = g(x(t), y(t))$$

Tools/Tasks/Browse: Differential Equations/ODEs/Phase portrait - Autonomous Systems

Phase Portraits for Autonomous Systems



Plot Window

$-1 \le x \le 1$,

$-1 \le y \le 1$

Differential Equations

$\dot{x} = F(x,y) = 3x + 5y$

$\dot{y} = G(x,y) = 2x - 7y$

Equilibrium (Critical) Points

[0, 0]

Parameter

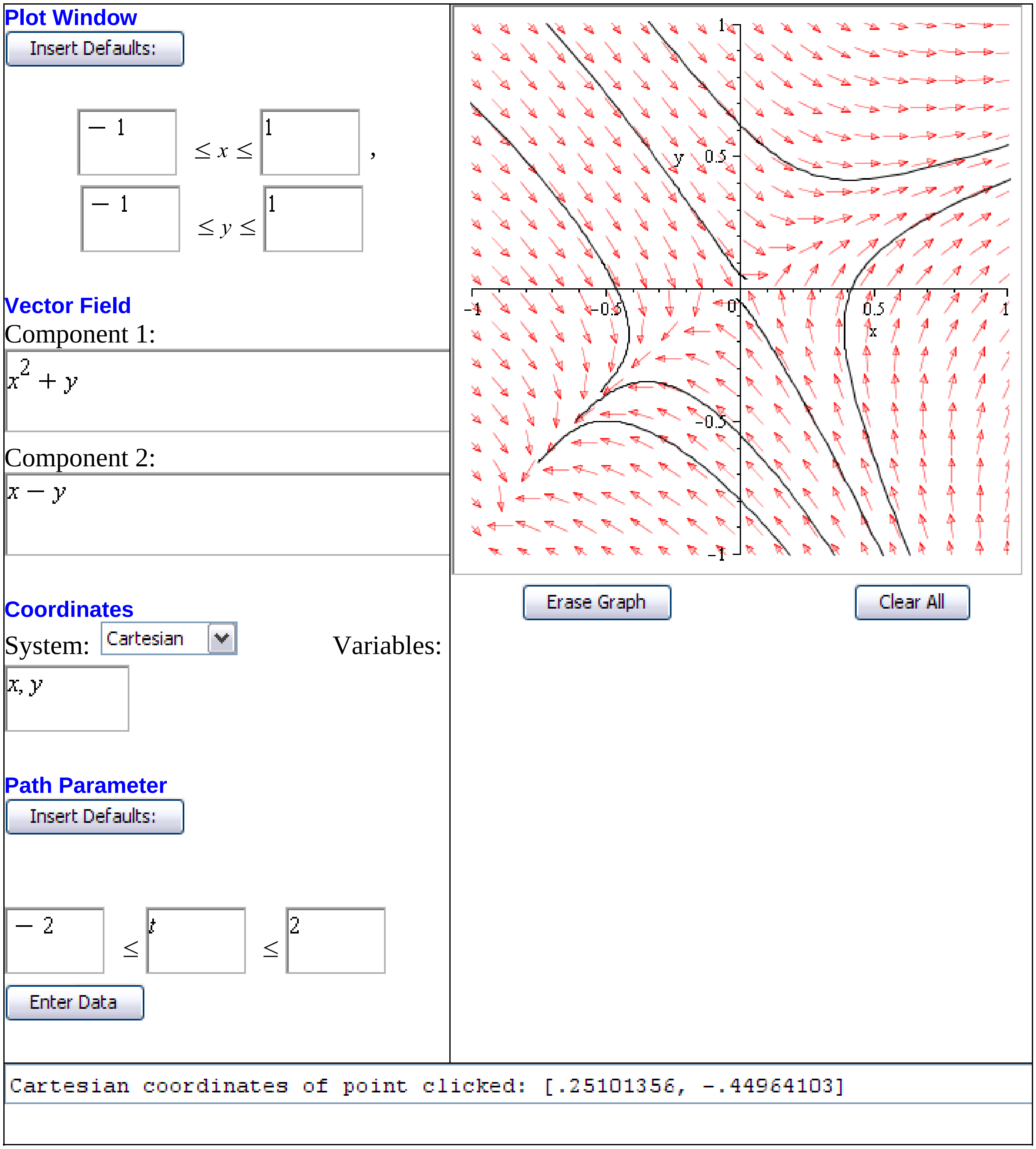
$-1 \le t \le 1$

Clicking on a point in the phase plane generates the orbit through that point.

Since graphing the flow lines of a planar vector field is equivalent to drawing the phase portrait of this same autonomous system, the next version of Maple will have an additional task template (available now from the author):

Tools/Tasks/Browse: Vector Calculus/Integrate Planar Vector Field

Integrate Planar Vector Field



Plot Window

$-1 \le x \le 1$,

$-1 \le y \le 1$

Vector Field

Component 1:

$x^2 + y$

Component 2:

$x - y$

Coordinates

System: Cartesian

Variables:

x, y

Path Parameter

$-2 \le t \le 2$

```
Cartesian coordinates of point clicked: [.25101356, -.44964103]
```

This task template is also based on the **DEplot** command from the DEtools package. However, the extra steps of converting the components of the vector field to the syntax of differential equations are hidden in the code behind the buttons.

Legal Notice: The copyright for this application is owned by Maplesoft. The application is intended to demonstrate the use of Maple to solve a particular problem. It has been made available for product evaluation purposes only and may not be used in any other context without