Chapter 3

The Art of Phugoid

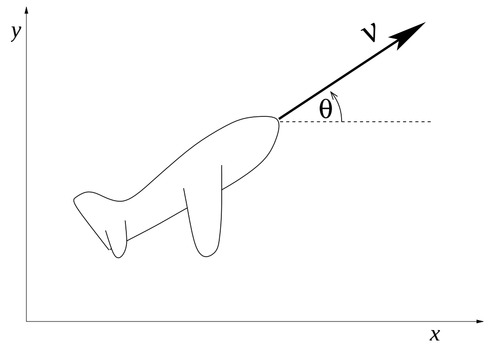
In this chapter, we will explore some aspects of mathematical model of glider flight. This model is called Lanchester's *phugoid theory*[1], developed by Frederick Lanchester [Lan] at beginning of the twentieth century. While this model has its drawbacks, it still is used today to explain oscillations and stalls in airplane flight.

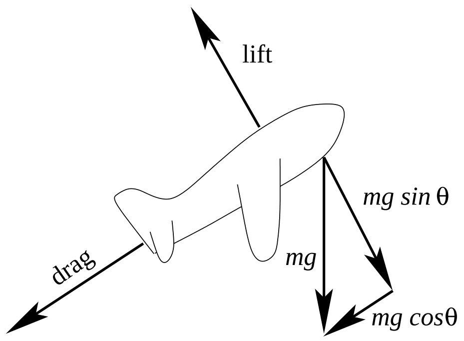
Since this model, like many models, is a set of differential equations, we will need some results which are typically covered in a course on differential equations. We will cover the relevant material briefly here, but readers needing a more in-depth treatment are encouraged to look in a text on ordinary differential equations, such as [BDH], [HW], or [EP]. Of these, the approach in [BDH] is perhaps closest to the one presented here.

1 The Phugoid model

The Phugoid model is a system of two nonlinear differential equations in a frame of reference relative to the plane. Let $v(t)$ be the speed the plane is moving forward at time t, and $\theta(t)$ be the angle the nose makes with the horizontal. As is common, we will suppress the functional notation and just write v when we mean $v(t)$, but it is important to remember that v and θ are functions of time.

[1]Folklore has it that this name was chosen by Lanchester because he wanted a classically-based name for his new theory of oscillations occuring during flight. Since the Greek root *phug* ($\phi\nu\gamma\eta$, pronounced "fyoog") as well as the Latin root *fug*- both correspond to the English word "flight", he decided on the name "phugoid". Unfortunately, in both Greek and Latin, this means "flight" as in "run away" instead of what birds and airplanes do— the same root gives rise to the words "fugitive" and "centrifuge". The Latin for flight in appropriate sense is *volatus*; the Greek word is *potê* ($\pi o\tau\eta$).





If we apply Newton's second law of motion (force = mass × acceleration) and examine the major forces acting on the plane, we see easily the force acting in the forward direction of the plane is

$$m\frac{dv}{dt} = -mg\sin\theta - \text{drag}.$$

This matches with our intuition: When θ is negative, the nose is pointing down and the plane will accelerate due to gravity. When $\theta > 0$, the plane must fight against gravity.

In the normal direction, we have centripetal force, which is often expressed as mv^2/r, where r is the instantaneous radius of curvature. After noticing that that $\frac{d\theta}{dt} = v/r$, this can be expressed as $v\frac{d\theta}{dt}$, giving

$$mv\frac{d\theta}{dt} = -mg\cos\theta + \text{lift}.$$

Experiments show that both drag and lift are proportional to v^2, and we can choose our units to absorb most of the constants. Thus, the equations simplify to the system

$$\frac{dv}{dt} = -\sin\theta - Rv^2 \qquad \frac{d\theta}{dt} = \frac{v^2 - \cos\theta}{v}$$

which is what we will use henceforth. Note that we must always have $v > 0$.

It is also common to use the notation $\dot{v}$ for $\frac{dv}{dt}$ and $\dot{\theta}$ for $\frac{d\theta}{dt}$. We will use these notations interchangeably.

2 What do solutions look like?

Now that we have some equations in hand, our first goal is to understand what the solutions look like. Note that the equations are nonlinear, and we cannot hope to find explicit formulae for the solutions— they don't exist except in very special cases.

Let's deal with the easiest case first: pretend that drag doesn't exist and set $R = 0$. Does this help?

The equations become

$$\frac{dv}{dt} = -\sin\theta \qquad \frac{d\theta}{dt} = \frac{v^2 - \cos\theta}{v}$$

Now, let's see what we can say about the solutions to this equation. First, notice that there must be many solutions: Any choice of initial conditions $\theta(0) = \theta_0, v(0) = v_0$ gives rise to a different solution. (Informally, if the glider starts with a different initial angle and/or velocity, it will fly along a different path. If we say it this way, it seems so obvious we shouldn't need to say it at all.) Can we find *any* solutions at all?

There is one very easy solution to find. Notice that if we can find a choice of θ and v so that $\frac{dv}{dt} = 0 = \frac{d\theta}{dt}$, then since their derivatives are zero, $v(t)$ and $\theta(t)$ must be constant functions. But solving this is simple:

$$0 = -\sin\theta \qquad\qquad 0 = \frac{v^2 - \cos\theta}{v}$$

is easily seen to be true when $\theta = 0$ and $v = 1$. In fact, θ can be any multiple of 2π. So we have found the special solution

$$v(t) \equiv 1 \qquad\qquad \theta(t) \equiv 0.$$

This is a glider which always flies level, with a constant velocity of 1.

What about other solutions? Now we must work harder. There is no hope to find a general analytic expression for $v(t)$ and $\theta(t)$, but that does not mean all is lost. We can get an idea of how other solutions behave by thinking about what it means to have a solution $\varphi(t) = [\theta(t), v(t)]$.

First, notice that such a solution φ can be thought of as a parametric curve in the (θ, v)-plane which passes through the point $\varphi(0) = [\theta_0, v_0]$. At any point along this parametric curve, we can calculate its derivative:

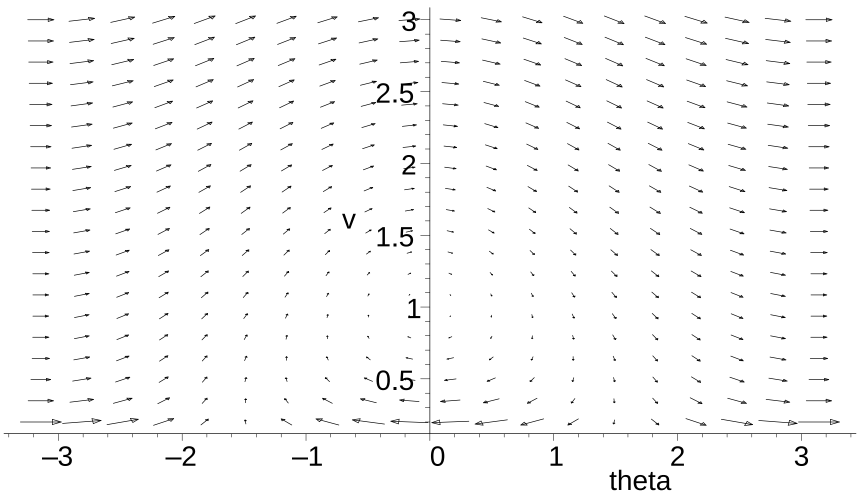
$$\varphi'(t) = \left[\frac{d}{dt}\theta(t), \frac{d}{dt}v(t)\right] = \left[\dot{\theta}(t), \dot{v}(t)\right].$$

But we started out with expressions for $\dot{\theta}$ and $\dot{v}$: our original differential equation. Because our differential equations only involve t implicitly via θ and v (the system is *autonomous*), we need only know the values of θ and v to know the tangent vector to any solution curve passing through that point.

We can get an idea of how the solutions behave by making a picture of the vector field corresponding to this: At each value for the pair (θ, v), we draw the vector $\langle\dot{\theta}, \dot{v}\rangle$, which only depends on (θ, v).

To get maple to do this, we can use the `fieldplot` command. Because the arrows in the `fieldplot` show not only direction but also relative magnitude, we shouldn't get too close to the $v = 0$ axis or they will overwhelm the smaller arrows for which occur for larger v.

```
> plots[fieldplot]( [ (v^2 - cos(theta))/v, -sin(theta) ],
                     theta=-Pi..Pi, v=0.2..3, arrows=slim);
```



We can certainly see that near the constant solution $\{\theta = 0, v = 1\}$, nearby solutions rotate around it in a clockwise manner. This "rotation" in the $\theta - v$ coordinates corresponds to a flight path which wobbles: the nose angle alternately increases and decreases, as does the velocity. For v large, it seems that the solutions move from left to right— θ constantly increases, which means that the glider is always increasing its nose angle. This means it is doing loops.

We can also use the `DEplot` command from Maple's `DEtools` package to produce a direction field. The direction field is obtained from the vector field by rescaling all nonzero vectors to be of unit length. Sometimes discarding the extra information makes it easier to see what is happening. In addition, `DEplot` has other very useful options, and will be one of our primary tools throughout this chapter.

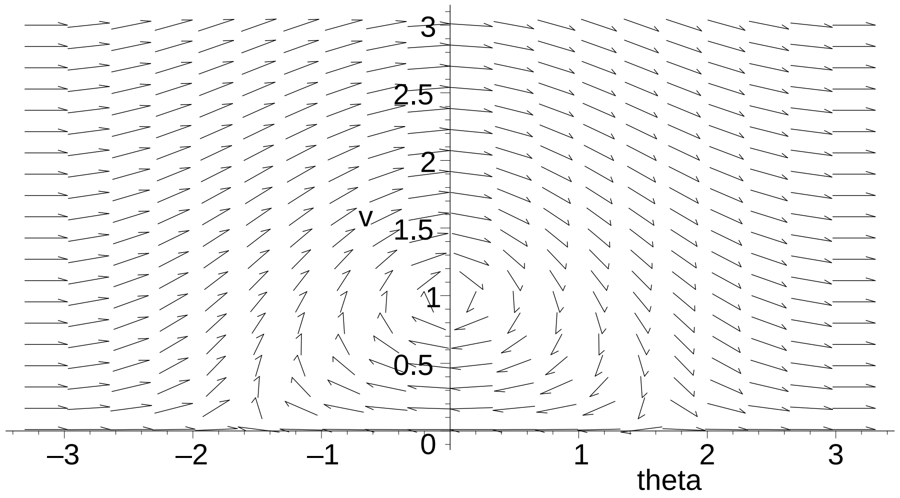
For convenience, we will name our system of differential equations as `phug`, so that we needn't keep retyping them.

```
> phug := [ diff(theta(t),t) =  (v(t)^2 - cos(theta(t)))/v(t),
            diff(v(t),t)     =  -sin(theta(t)) - R*v(t)^2 ];
  R:=0;
```

$$phug := [\frac{\partial}{\partial t}\,\theta(t) = \frac{\mathrm{v}(t)^2 - \cos(\theta(t))}{\mathrm{v}(t)},\ \frac{\partial}{\partial t}\,\mathrm{v}(t) = -\sin(\theta(t))]$$

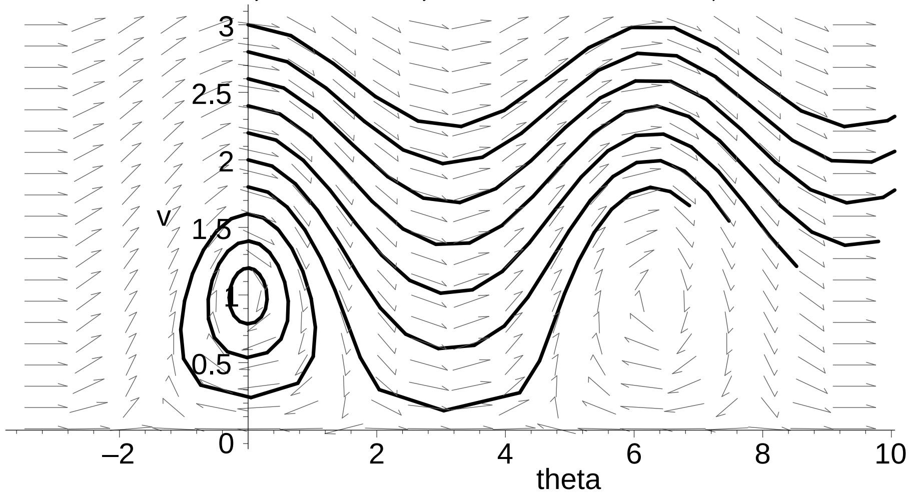
$$R := 0$$

```
> with(DEtools):
  DEplot( phug, [theta(t), v(t)], t=0..1, theta=-Pi..Pi, v=0.01..3);
```



We can also ask maple to plot a number of solutions corresponding to several initial conditions. Rather than typing in 11 initial conditions, we can use `seq` to generate them. We tell maple that the solution curves should be black, because otherwise it uses a sickly yellow. If we had wanted each solution curve to be a different color, we could use something like `linecolor=[seq(COLOR(HUE,i/10),i=0..10)]`.

```
> DEplot( phug, [theta(t), v(t)], t=0..5,
          [seq([theta(0)=0, v(0)=1+0.2*i], i=0..10)],
          theta=-Pi..3*Pi, v=0.01..3, linecolor=black );
```

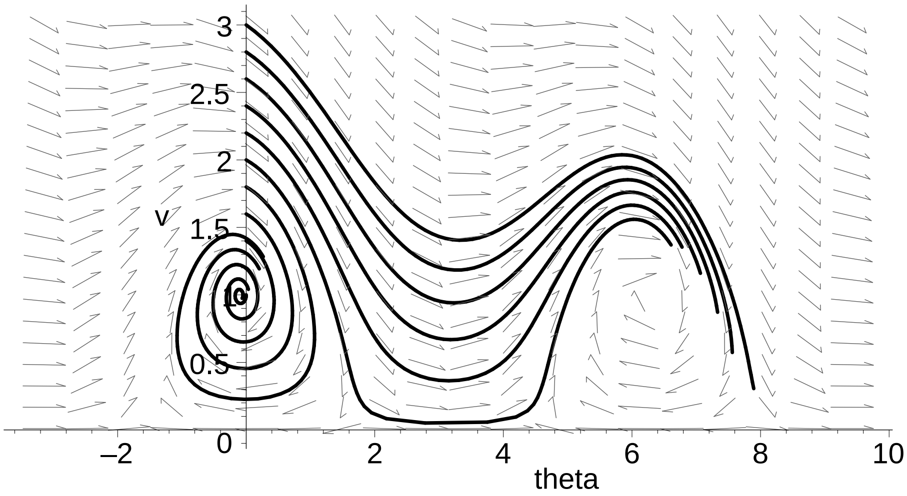


The above figure confirms our earlier impressions about the “wobbling” and “looping” solutions. We'll see how to get Maple to demonstrate this more explicitly in section 5, where we will plot the path of the glider through space (i.e., in x-y coordinates).

Just to see how things differ if we make R nonzero, let's change R and get Maple to show us what happens. We need to adjust the stepsize to get a reasonable accuracy; what precisely this does is described in the next section.

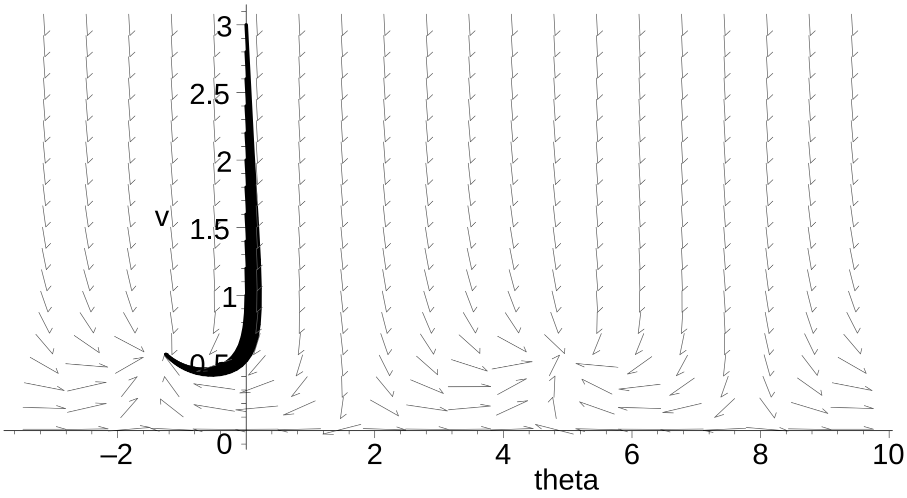
```
> R:=0.1;
  DEplot( phug, [theta(t), v(t)], t=0..5,
          [seq([theta(0)=0, v(0)=1+0.2*i], i=0..10)],
          theta=-Pi..3*Pi, v=0.01..3, linecolor=black, stepsize=0.1 );
```

$$R := .1$$




```
> R:=3;
  DEplot( phug, [theta(t), v(t)], t=0..5,
          [seq([theta(0)=0, v(0)=1+0.2*i], i=0..10)],
          theta=-Pi..3*Pi, v=0.01..3, linecolor=black,stepsize=0.1 );
```

$$R := 3;$$



The behavior of the glider appears to be dramatically affected by the drag parameter R. One of our main goals in this chapter will be to classify the types of solutions that are possible as a function of the drag parameter R.

3 Existence of Solutions

We've been acting as though just by specifying an initial condition, there must be a solution, and it must be unique (that is, the only one corresponding to that initial condition). And, in fact, this is typically true for any "nice" differential equation. But which differential equations are "nice" enough?

We won't prove the theorem here, but we will state it. The reader is encouraged to look up the proof in any differential equations text, such as [BDH] or [HW].

Theorem 3.1 *Consider the differential equation*

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}, t).$$

If $\vec{F}$ is continuous on an open set in containing $(\vec{x}_0, t_0)$, then there is a solution $\vec{x}(t)$ which is defined on some interval $(t_0 - \epsilon, t_0 + \epsilon)$ which satisfies the initial value problem

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}, t) \qquad \vec{x}(t_0) = \vec{x}_0.$$

Furthermore, if $\vec{F}$ has continuous partial derivatives, this solution is unique.

In fact, the hypothesis can be weakened a little bit and still preserve the uniqueness. As long as $\vec{F}$ is at least Lipshitz in $\vec{x}$, the solution will be unique. A function $\vec{F}$ if called Lipschitz if there is some K so that

$$|\vec{F}(\vec{x}_1, t) - \vec{F}(\vec{x}_2, t)| < K|\vec{x}_1 - \vec{x}_2|$$

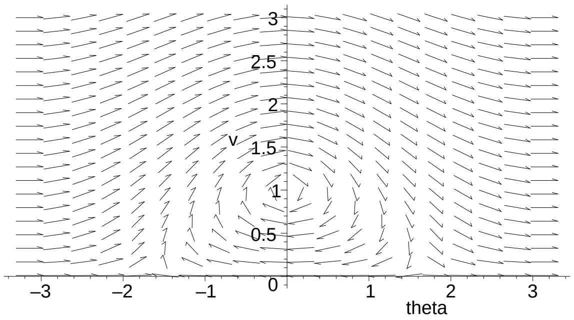
for all t, $\vec{x}_1$, and $\vec{x}_2$. This condition says, roughly, that $\vec{F}$ doesn't spread out too quickly. All functions with continuous partials are automatically Lipshitz.

This means that for us, as long as we stay away from the place where our differential equation isn't defined (i.e. ensure that $v > 0$), we can be assured that there is a solution through every point, and that solution is unique. Notice that since our equations are autonomous (the right-hand side has no explict dependence on t), uniqueness of solutions means that solutions cannot cross in the (θ, v)-plane.

4 Numerical Methods

What does Maple do when you ask it to display a solution of a differential equation with `DEplot`? Obviously, it cannot solve the equations analytically, since solutions don't always exist in closed form. Instead, it approximates them numerically.

4.1 Euler's method



To get an idea of how this can be done, take a look again at the direction field for the glider. By looking at how the arrows point, you can almost guess how the solution must turn. From a given initial condition, move in the direction of the arrow to the next position. At that new point, step in the direction of the vector at that point, and so on. This is the idea behind the simplest numerical integration scheme, called Euler's method.

More precisely, given a differential equation and an initial condition

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}, t) \qquad\qquad \varphi(t_0) = \vec{x}_0,$$

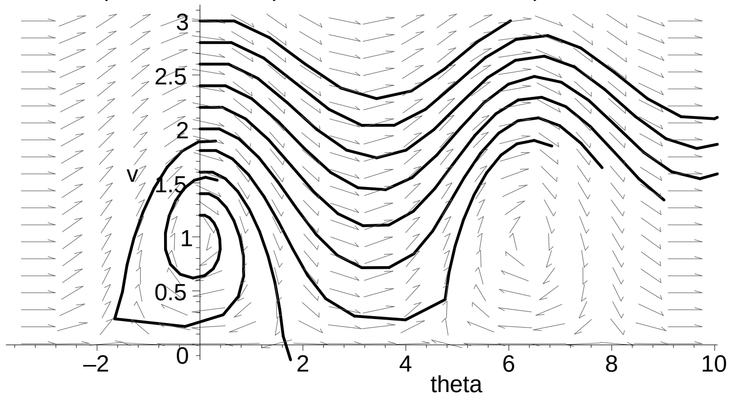
the goal is to construct a sequence of points $\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ which approximate the solution φ at times $t_0, t_1, t_2, \ldots, t_n$. We take our initial point to be the initial condition, and choose some small number $h > 0$ called the *stepsize*. Then we set

$$\vec{x}_{i+1} = \vec{x}_i + h\vec{F}(\vec{x}_i, t_i) \qquad\qquad t_{i+1} = t_i + h$$

for $0 \le i \le n-1$. This very simple scheme works best for small values of h; the error in the approximation for a fixed time interval is proportional to h. This means that to improve the accuracy of a numerical solution by a factor of 10, we need to do about 10 times more work.

Maple doesn't typically use Euler's method because the errors accumulate too quickly. To see this, we reproduce one of the earlier figures from §2 using Euler's method.

```
> DEplot( phug, [theta(t), v(t)], t=0..5,
      [seq([theta(0)=0, v(0)=1+0.2*i], i=0..10)],
      theta=-Pi..3*Pi, v=0.01..3, linecolor=black, method=classical[foreuler]);
```



While the errors in the above figures are considerable, using a much smaller stepsize (say, `stepsize=0.01`) will essentially reproduce the earlier figure, with a lot more computation.

4.2 Numerical Solutions, Numerical Integration, and Runge-Kutta

To get some ideas about improving on Euler's method, let's first notice that Euler's method corresponds exactly to the left-hand Riemann sum when computing a numerical approximation to an integral. You may recall from calculus that the left-hand sum converges to the value of the integral, but you must take a large number of subdivisions to get any accuracy.

A more efficient method is the trapezoid rule, which is the average of the left-hand and right-hand sum. There is a numerical scheme for integrating ODEs which corresponds to this, known variously as the *improved Euler method* or the *Heun formula.* In the trapezoid rule, we compute the function at the left and right endpoints of the function and average the result; the analog of this would require us to average the value of the vector field at our current point and at the next point. Perhaps you see the circularity implicit in that reasoning: to know the next point, we would need to solve the differential equation. So, instead we fudge a bit: we get our approximation of the next point by using Euler's method. More specifically, using the notation of §4.1, given an approximation $\vec{x}_i$ at time t_i, we find the next one using the scheme below.

$$
\begin{array}{rcll}
t_{i+1} & = & t_i + h & \\
\vec{k}_\ell & = & \vec{F}(\vec{x}_i, t_i) & \text{slope at the left side of the interval} \\
\vec{k}_r & = & \vec{F}(\vec{x}_i + h\vec{k}_\ell, t_{i+1}) & \text{slope at the right side of the interval} \\
\vec{x}_{i+1} & = & x_i + h(\vec{k}_\ell + \vec{k}_r)/2 &
\end{array}
$$

To do the improved Euler method, we need to evaluate the vector field twice as often as the regular Euler method, but there is a big gain in accuracy. The error of the approximation for a fixed time interval is proportional to h^2. Thus, if we decrease h by a factor of 10, we should expect to reduce the error 100-fold. You can get Maple to use the improved Euler method by specifying `method=classical[heunform]` as an argument to `DEplot` and related commands.

You might remember from your calculus course that Simpson's rule can get very accurate numerical approximations for integrals with a very few number of points. This is because Simpson's rule is a fourth-order method: the error in the approximation is proportional to h^4. For Simpson's rule, we evaluate the function at the right endpoint, the left endpoint, and the middle, and then take a weighted average of those values.

The analog of Simpson's rule for differential equations is called the Runge-Kutta method[2], developed by the German mathematicians C. Runge and W. Kutta at the end of the nineteenth

[2]In fact, there is a whole family of such methods, of various orders. But generally, if someone says "Runge-Kutta" without specifying the order, they mean this fourth-order Runge-Kutta.

century. It takes a linear combination of four slopes, as below:

$$
\begin{array}{rcll}
t_{i+1} & = & t_i + h & \\
\vec{k}_\ell & = & \vec{F}(\vec{x}_i, t_i) & \text{slope at the left side of the interval} \\
\vec{k}_m & = & \vec{F}(\vec{x}_i + \frac{h}{2}\vec{k}_\ell, t_i + \frac{h}{2}) & \text{slope at the middle of the interval} \\
\vec{k}_c & = & \vec{F}(\vec{x}_i + \frac{h}{2}\vec{k}_m, t_i + \frac{h}{2}) & \text{corrected slope at the middle of the interval} \\
\vec{k}_r & = & \vec{F}(\vec{x}_i + h\vec{k}_c, t_{i+1}) & \text{slope at the right side of the interval} \\
\vec{x}_{i+1} & = & x_i + h(\vec{k}_\ell + 2\vec{k}_m + 2\vec{k}_c + \vec{k}_r)/6 &
\end{array}
$$

When the slope depends only on t and not on $\vec{x}$, then $\vec{k}_m$ and $\vec{k}_c$ are the same, and the formula reduces exactly to that of Simpson's rule.

Runge-Kutta is probably the most commonly used method of numerical integration, because of its generally high accuracy. Like Simpson's rule, it is a fourth-order method, so the error is proportional to h^4. Maple has several numerical methods for ODEs built in to it; see the help page on `dsolve[numeric]` for more information about them; the ones we have described are "classical" methods, and are described (along with others) on Maple's help page for `dsolve[classical]`. Unless asked to do otherwise, Maple's `DEplot` command uses the fourth-order Runge-Kutta method described above (Maple's name for this method is `classical[rk4]`), although the `numeric` option of `dsolve` defaults to a variation called *fourth-fifth order Runge-Kutta-Fehlberg*, which Maple refers to as `rkf45`.

5 Seeing the flight path

In §2, we exhibited a number of solutions to the phugoid model with various initial conditions and values for the drag, as plots of the parametric curves $[\theta(t), v(t)]$. From these, we could deduce an approximate path of the glider through space via reasoning like "$\theta(t)$ is monotonically increasing, so the glider must be doing loop-the-loops". We will now discuss how to solve for the position $[x(t), y(t)]$ directly.

First, notice that if we know the value of v and θ at some time t, we also have $\frac{dx}{dt}$ and $\frac{dy}{dt}$ at that t, since

$$\frac{dx}{dt} = v\cos\theta \qquad \text{and} \qquad \frac{dy}{dt} = v\sin\theta.$$

So, one way to find $x(t)$ and $y(t)$ would be to solve the original ODE (numerically) for $\theta(t)$ and $v(t)$, obtain a new differential equation, and then solve that numerically for $x(t)$ and $y(t)$.

Aside from the fact that there is an easier way, there is a serious practical problem with this approach. When we get a numerical solution $[\theta(t), v(t)]$, this is a collection of points $(\theta_0, v_0, t_0), (\theta_1, v_1, t_1), \ldots, (\theta_n, v_n, t_n)$ along a curve. This means we only know the vector field for (x, y) at those points. It is highly unlikely that these will be the points we need to use Runge-Kutta. We could use these to approximate our (x, y)-solution using Euler's method, although even then we would have to program it ourselves rather than use Maple's built-in method.

If you think for just a moment, you should realize that there is no reason we must find $\theta(t)$ and $v(t)$ before attempting to find $x(t)$ and $y(t)$. Instead, we can augment the original system, adding in the new equations for $\dot{x}$ and $\dot{y}$. Then, we can use numerical methods to solve for $[\theta(t), v(t), x(t), y(t)]$ all in one step. The methods we discussed in §4.2 apply equally well to vector fields in any number of variables.

Consequently, we rewrite our phugoid system as

$$\frac{d\theta}{dt} = \frac{v^2 - \cos\theta}{v} \qquad \frac{dv}{dt} = -\sin\theta - Rv^2 \qquad \frac{dx}{dt} = v\cos\theta \qquad \frac{dy}{dt} = v\sin\theta.$$

Using this system, we need to specify our initial conditions as $(\theta_0, v_0, x_0, y_0)$, and each solution is a curve in $\mathbb{R}^4$ parameterized by time. We can take a projection of this curve to get our solutions in θ-v or x-y coordinates.

Asking Maple to solve this new system is almost the same as before, but we must specify the additional variables and use the `scene` option to `DEplot` to tell Maple which projection we want to see. Note that we reset R to ensure that it appears in the equations as an arbitrary constant, rather than with any previously assigned value.

```
> R:='R':
  xphug :=[ diff(theta(t),t) =  (v(t)^2 - cos(theta(t)))/v(t),
            diff(v(t),t)     =  -sin(theta(t)) - R*v(t)^2 ,
            diff(x(t),t)     =  v(t)*cos(theta(t)),
            diff(y(t),t)     =  v(t)*sin(theta(t))];
```

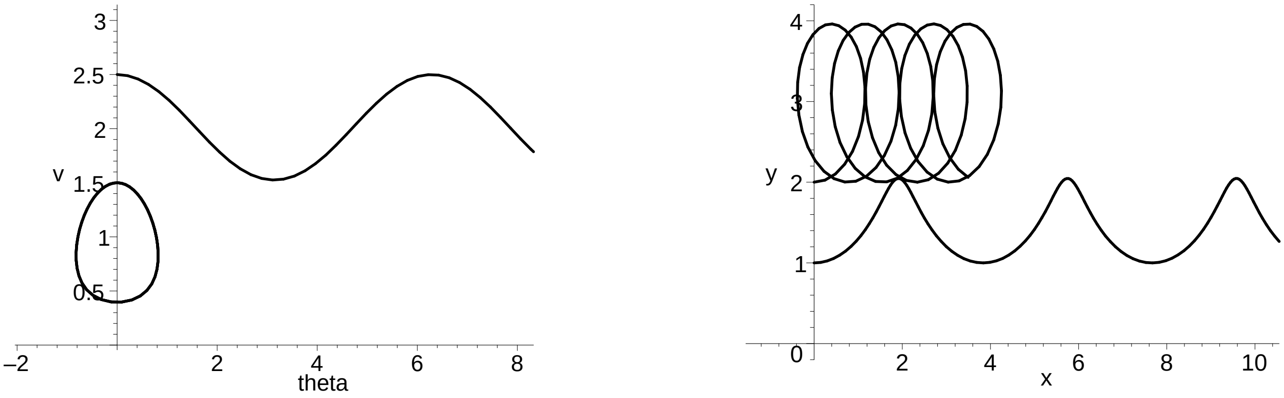
$$xphug := [\frac{\partial}{\partial t}\theta(t) = \frac{\mathrm{v}(t)^2 - \cos(\theta(t))}{\mathrm{v}(t)}, \frac{\partial}{\partial t}\mathrm{v}(t) = -\sin(\theta(t)) - R\,\mathrm{v}(t)^2, \frac{\partial}{\partial t}\mathrm{x}(t) = \mathrm{v}(t)\cos(\theta(t)), \frac{\partial}{\partial t}\mathrm{y}(t) = \mathrm{v}(t)\sin(\theta(t))]$$

Now we ask Maple to plot two solution curves in both θ, v coordinates and x, y. One is started with an initial angle of $\theta = 0$ and velocity $v = 1.5$ at a height $y(0) = 1$; the other starts with $\theta = 0$, $v = 2.5$, and initial height $y = 2$. We also use `display` with an `array` to show the two scenes side by side.

```
> R:=0;
  plots[display](array([
   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[theta,v]),

   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[x,y]),
   ]));
```

$R := 0;$

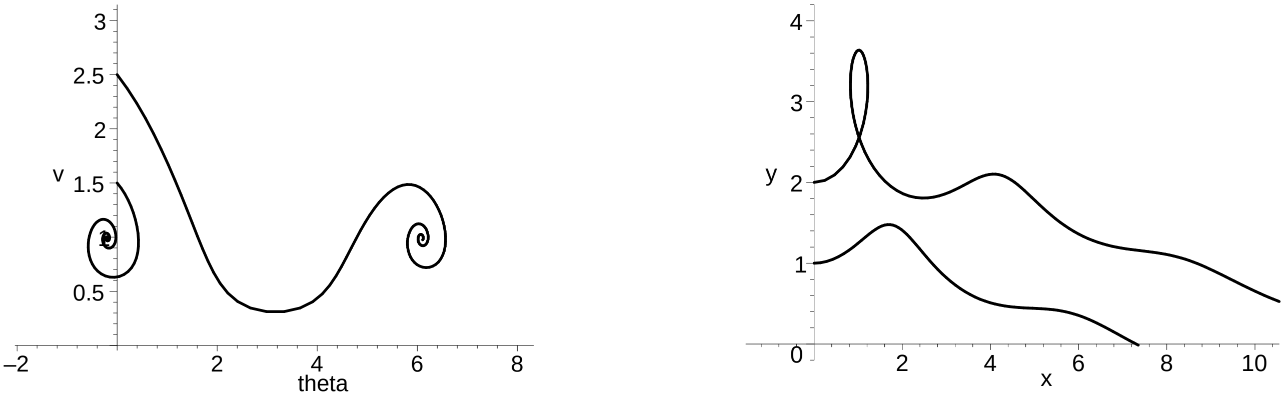


If we change the value of R to 0.2 and repeat the command, we can see how things change when there is friction. Notice how spiraling towards the fixed point in the θ-v coordinates corresponds to a plane which is diving with an oscillatory motion.

```
> R:=0.2;
  plots[display](array([
   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[theta,v]),

   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[x,y]),
   ]));
```

$R := 0.2;$

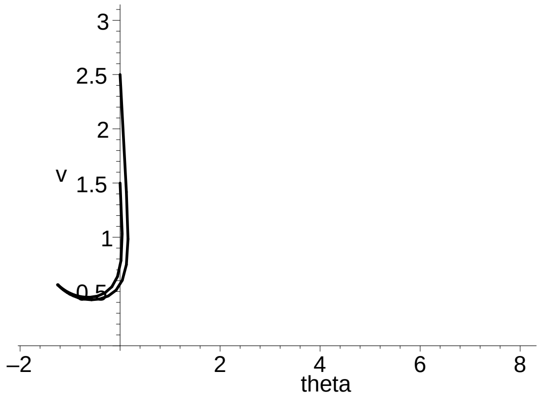


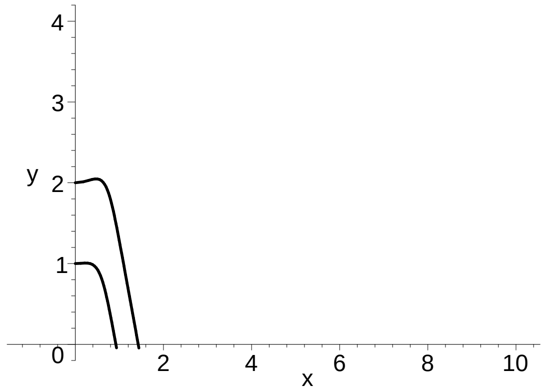
Finally, we redo the pictures with a large amount of friction ($R = 3$). Note how the glider acts more like a rock than a glider, and there is no oscillation at all.

```
> R:=3;
  plots[display](array([
   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[theta,v]),

   DEplot(xphug, [theta,v,x,y], t=0..15,
    [[theta(0)=0,v(0)=1.5,x(0)=0,y(0)=1],[theta(0)=0,v(0)=2.5,x(0)=0,y(0)=2]],
    theta=-Pi..5*Pi/2, v=0.1..3, x=-1..10, y=0..4,
    linecolor=black, stepsize=0.1, scene=[x,y]),
   ]));
```

$$R := 3;$$





6 Fixed Point Analysis

So far, the cases we've looked at seem to have a number of common features. Each has has a solution with $\theta(t)$ and $v(t)$ constant; there are solutions which immediately tend toward the constant solution (or oscillate around it), and solutions which do some number of loops first (for $R = 3$, we didn't demonstrate this– if the inital angle is 0, a solution needs an initial velocity larger than 86.3 before it will do a loop). How do the possible solutions depend on the value of R? For example, the behavior of solutions for $R = 0.1$ and $R = 0.2$ are quite similar, but are dramatically different from $R = 0$ and $R = 3$.

One way we can get a better handle on exactly how the behavior of the solutions depends on R is to examine what happens to solutions near the constant solution. If we look in the θ-v plane, this solution corresponds to a single point, which is often referred to as a "fixed point". We explicitly found the fixed point $\{\theta(t) = 0, v(t) = 1\}$ solution for $R = 0$ in §2 by noticing that whenever $\dot{\theta} = 0$ and $\dot{v} = 0$, we must have a constant solution (and vice-versa). Using `solve`, we can get maple to tell us the where the fixed point is for arbitrary values of R. We use `convert` to insist that maple represent the result as a radical, instead of using the `RootOf` notation.

```
> R:='R':
  FixPoint:=convert(
            solve({(v^2-cos(theta))/v=0,-sin(theta)-R*v^2=0}, {theta,v}),
            radical);
```

$$FixPoint := \{v = (\frac{1}{R^2+1})^{(1/4)},\ \theta = \arctan(-R\sqrt{\frac{1}{R^2+1}},\ \sqrt{\frac{1}{R^2+1}})\}$$

So, we see that there is a fixed point for all values of R (since $1 + R^2$ is always positive, the radical always takes real values). For $R > 0$, $\theta(t)$ at this fixed solution is negative, so this corresponds to a diving solution. As R increases, the angle of the dive becomes steeper and steeper.

What can we say about behaviour of solutions *near* this fixed solution?

In order to answer this, notice that we can apply Taylor's theorem to the right-hand side of our system of differential equations. That is, think of our differential equation as being in the vector form

$$\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}),$$

where (in our case) $\vec{X}$ is $[\theta(t), v(t)]$, and $\vec{F}(X) = [(v^2 - \cos(\theta))/v,\ -\sin(\theta) - Rv^2]$. Taylor's theorem says that for $\vec{X}$ near $\vec{X}_0$,

$$\vec{F}(\vec{X}) = \vec{F}(\vec{X}_0) + D\vec{F}(\vec{X}_0)(\vec{X} - \vec{X}_0) + \text{higher order terms},$$

where $D\vec{F}(\vec{X}_0)$ is the Jacobian of $\vec{F}$ evaluated at $\vec{X}_0$. Since $\vec{X}_0$ is a fixed point, $\vec{F}(\vec{X}_0)$ is the zero vector. Hence, we can get a good idea of what is happening close to the fixed point by studying the *linear* system of differential equations given by the derivative at the fixed point.

First, we give a brief refresher about linear 2×2 systems of equations. The reader is refered to the texts in the references for more comprehensive treatment of this topic.

6.1 Linear Systems of ODEs

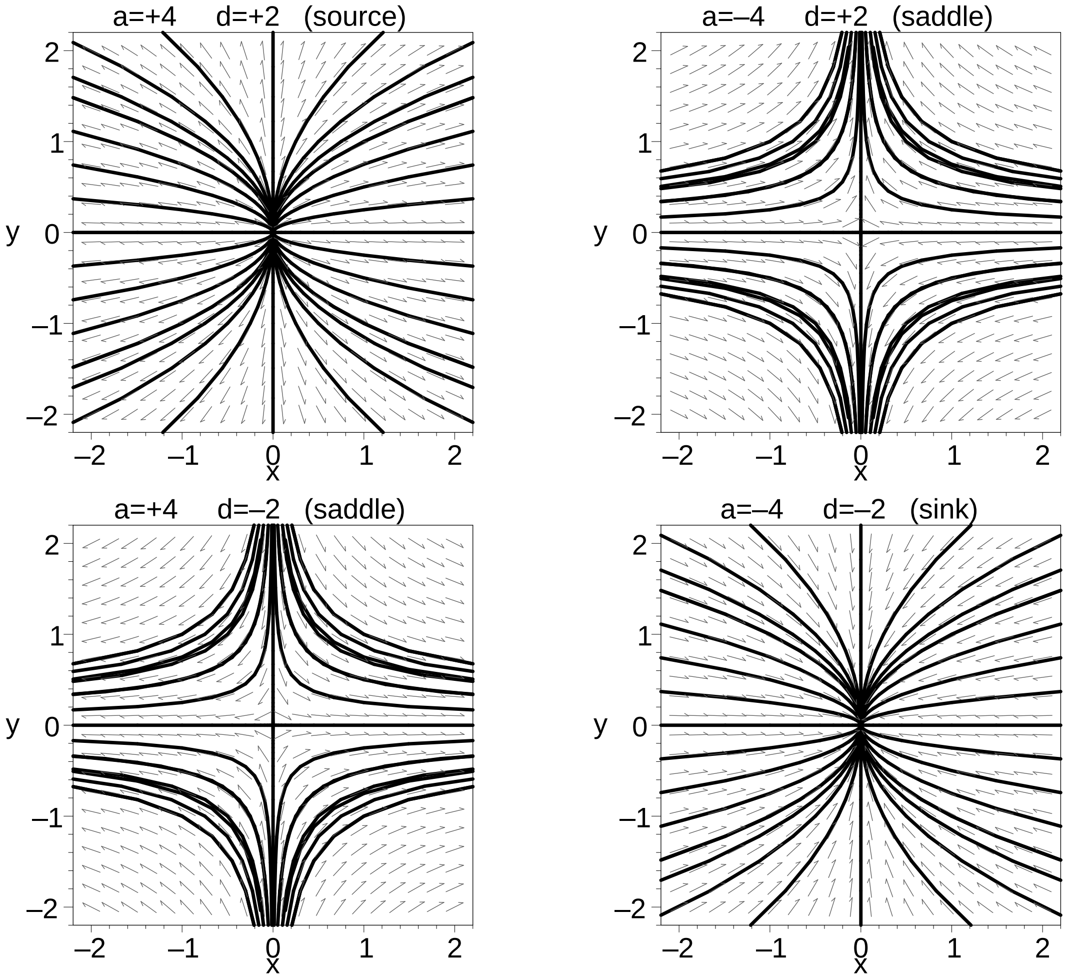
Linear differential equations (with constant coefficients) are the simplest ones to solve and understand. Such equations have the form $\frac{dX}{dt} = AX$, where A is an $n \times n$ matrix and X is an n-vector. We'll be content with $n = 2$.

First, notice that $X = 0$ is always a fixed point for this system, and it is the only one. Secondly, if $X_1(t)$ and $X_2(t)$ are solutions to the system, then so is $X_1(t) + X_2(t)$. This property is called *superposition* and will be very useful.

Let's look at the simplest case for the 2×2 system: $A = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}$, that is,

$$\begin{aligned} \dot{x} &= ax \\ \dot{y} &= \quad dy \end{aligned}$$

If we have an initial condition with $y(0) = 0$, then $y(t) = 0$. This means that the problem reduces to the one-dimensional case, and $x(t) = x(0)e^{at}$. If $a > 0$, the solution moves away from the origin along the x-axis as t increases; if $a < 0$, the solution moves toward the origin. If the initial condition is on the y-axis, the sign of d controls whether the solution moves away ($d > 0$) or towards ($d < 0$) the origin. Because of the superposition property mentioned before, an initial condition with $x(0) \neq 0$ and $y(0) \neq 0$ gives rise to a solution which is a combination of these two behaviours.



A large number of linear systems behave very similarly to the cases above, except the straight-line solutions may not lie exactly along the coordinate axes.

Suppose there is a solution of $\dot{X} = AX$ which lies along a straight line, that is, along some vector $\vec{v}$. Because the system is linear, this means the tangent vectors to this solution must be of the form λX for some number λ. Such a number λ is called an *eigenvalue*, and the corresponding vector $\vec{v}$ is the corresponding *eigenvector*. Note that in the case above, we had eigenvalues a and d with eigenvectors $[1,0]$ and $[0,1]$.

To find the eigenvalues, we need to solve

$$AX = \lambda X, \quad \text{or equivalently,} \quad (A - \lambda I)X = 0.$$

This can only happen if X is the zero vector, or the determinant of $A - \lambda I$ is zero. In the latter case, we must have

$$(a - \lambda)(d - \lambda) - bc = 0,$$

That is,

$$\lambda^2 - (a + d)\lambda + (ad - bc) = 0.$$

Using the quadratic formula, we see that the eigenvalues must be

$$\lambda = \frac{(a + d) \pm \sqrt{(a + d)^2 - 4(ad - bc)}}{2}.$$

The quantity $a + d$ is called the *trace* of A (more generally, the trace of a matrix is the sum of diagonal entries), and $ad - bc$ is the determinant of A. The eigenvalues of a 2×2 matrix can be expressed in terms of the trace and the determinant[3] as

$$\frac{TrA \pm \sqrt{(TrA)^2 - 4\,detA}}{2}.$$

We'll use this form again in a little while.

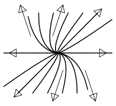
For linear systems, the eigenvalues determine the ultimate fate of solutions to the ODE. From the above, it should be clear that there are always two eigenvalues for a 2×2 matrix (although sometimes there might be a double eigenvalue, when $(TrA)^2 - 4\,detA = 0$). Let's call them λ_1 and λ_2.

We've already seen prototypical examples where the eigenvalues are real, nonzero, and distinct. But what if the eigenvalues are complex conjugate (which happens when $(TrA)^2 < 4\,detA$)? In this case, there is no straight line solution in the reals. Instead, solutions turn around the origin. If the real part of the eigenvalues is positive, solutions spiral away from the origin, and if it is negative, they spiral towards it. If the eigenvalues are purely imaginary, the solutions neither move in nor out; rather, they circle around the origin.

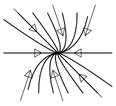
We can summarize this in the table below. We are skipping over dealing with the degenerate cases, when $\lambda_1 = \lambda_2$ and when one of the eigenvalues is zero.

[3] It is true for all matrices that the trace is the sum of the eigenvalues, and the determinant is their product.

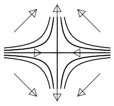
$\lambda_1 > \lambda_2 > 0$
source



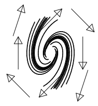
$\lambda_1 < \lambda_2 < 0$
sink



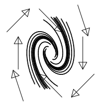
$\lambda_1 < 0 < \lambda_2$
saddle



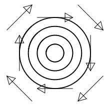
λ_i complex, $Re\lambda > 0$
spiral source



λ_i complex, $Re\lambda < 0$
spiral sink



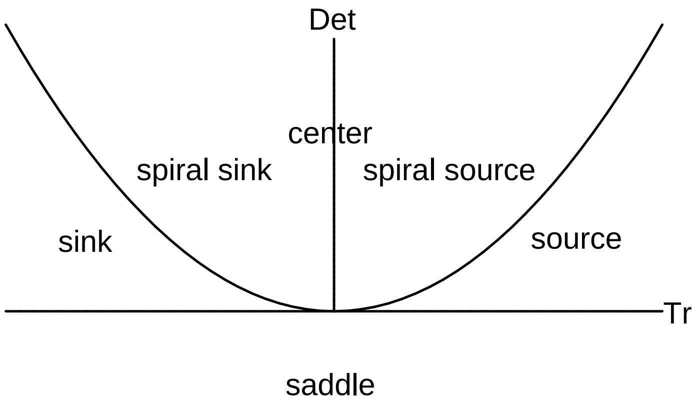
λ_i complex, $Re\lambda = 0$
center



Finally, we remark that there is a convenient way to organize this information into a single diagram. Since the eigenvalues of A can be written as

$$\frac{TrA \pm \sqrt{(TrA)^2 - 4\,detA}}{2},$$

we can consider the matrix A as being a point in the Trace-Determinant plane. Then the curve where $(TrA)^2 = 4\,detA$ is a parabola. If A is above this parabola, it has complex eigenvalues. Furthermore, if the determinant is negative, we must have a saddle because[4] $\sqrt{(TrA)^2 - 4\,detA} > TrA$, which means there is one positive and one negative eigenvalue. Finally, if the determinant is positive, the eigenvalues (or their real part, if they are complex) is the same sign as the trace of A. We summarize this in the diagram below.



[4]Since the determinant is the product of the eigenvalues, the only way it can be negative is if they are of opposite signs.

6.2 Fixed Points for the Glider

We now return to our nonlinear system, and look at the linearization near the fixed point. As we saw earlier in this section, for every value of R the Phugoid model

$$\frac{d\theta}{dt} = \frac{v^2 - \cos\theta}{v} \qquad \frac{dv}{dt} = -\sin\theta - Rv^2$$

has a fixed point at

$$\theta = -\arctan\left(R\sqrt{\frac{1}{1+R^2}}\right) \qquad v = \sqrt[4]{\frac{1}{1+R^2}}.$$

We now calculate the Jacobian matrix, using Maple.

```
> with(linalg):
  R:='R':
  J:=jacobian( [(v^2-cos(theta))/v, -sin(theta)-R*v^2], [theta, v]);
```

$$J := \begin{bmatrix} \dfrac{\sin(\theta)}{v} & 2 - \dfrac{v^2 - \cos(\theta)}{v^2} \\ -\cos(\theta) & -2\,R\,v \end{bmatrix}$$

Once we know the trace and determinant of the Jacobian matrix at the fixed point, we can use that information to determine how its type (sink, source, saddle, etc.) depends on R. Rather than solve for the eigenvalues directly, we will compute the trace and determinant, which have a simpler form in this case.

First, let's find the fixed point again, issuing the same command we did earlier. Then, we'll substitute that into the result of `trace` and `det` to calculate the trace and determinant at the fixed point.

```
> FixPoint:=convert(
            solve({(v^2-cos(theta))/v=0,-sin(theta)-R*v^2=0}, {theta,v}),
            radical);
```

$$\mathit{FixPoint} := \{\theta = \arctan(-R\sqrt{\frac{1}{R^2+1}}, \sqrt{\frac{1}{R^2+1}}),\ v = (\frac{1}{R^2+1})^{(1/4)}\}$$

```
> trfix := simplify(subs(FixPoint, trace(J)));
  detfix:= simplify(subs(FixPoint, det(J)));
```

$$\mathit{trfix} := \frac{-3R}{(R^2+1)^{1/4}}$$

$$\mathit{detfix} := 2\sqrt{R^2+1}$$

So, we see that for all real values of R, the determinant is positive (meaning saddles are impossible). Furthermore, for $R > 0$, the trace is negative, so the fixed point is always a sink. (When $R = 0$, the trace is zero, which means it is possible for this fixed point to be a center.[5]) From the plots we made earlier, we expect that for small values of R, we will have a spiral sink, but it will probably not spiral for sufficiently large values of R. We can check this by solving for the point where $Tr(J)^2 = 4\det(J)$, which happens when $R = 2\sqrt{2}$.

```
> solve( (trfix)^2 = 4*detfix, R);
```

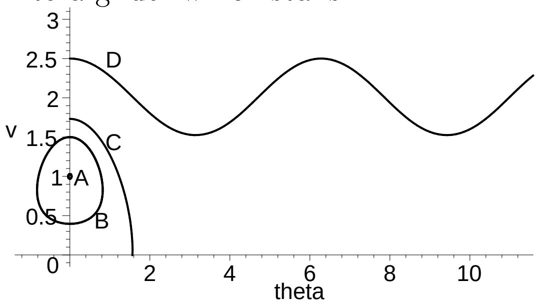
$$2\sqrt{2},\ -2\sqrt{2}$$

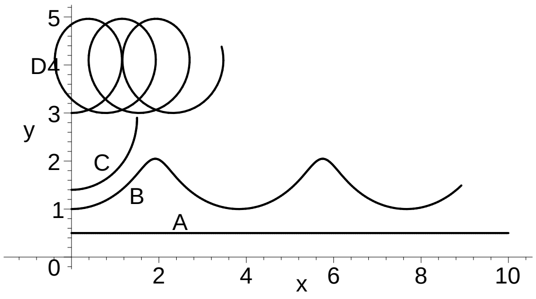
7 Qualitative Classification of Solutions

This analysis of the previous section allows us to completely classify what happens in the Phugoid model for all $R \in [0, \infty)$. In all cases, the fixed point in (θ, v) coordinates corresponds to the solution

$$\theta(t) = -\arctan\left(R\sqrt{\frac{1}{1+R^2}}\right) \quad v(t) = \sqrt[4]{\frac{1}{1+R^2}} \quad x(t) = \frac{t}{(1+R^2)^{\frac{3}{4}}} \quad y(t) = \frac{-R\,t}{(1+R^2)^{\frac{3}{4}}}$$

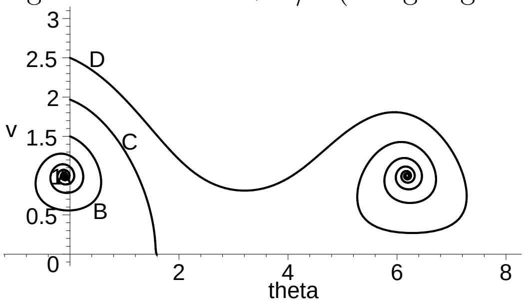
$R = 0$: In this case, there is no drag. The fixed point at $\theta = 0, v = 1$ corresponds to a glider flying level with a constant speed (labeled `A` in the figures below). The fixed point is a center in θ, v coordinates: nearby solutions are closed curves and correspond to a glider with an oscillatory path, alternately diving and climbing (labeled `B`). For initial conditions further away from the fixed point, $v(t)$ oscillates, but $\theta(t)$ constantly increases (see `D`). Such solutions correspond to a glider endlessly looping, and a pilot with a severe case of nausea. Between these two is a solution which cannot be continued beyond a certain time (`C`), because $v(t)$ becomes zero and our equations are no longer defined. This corresponds to a glider which stalls.

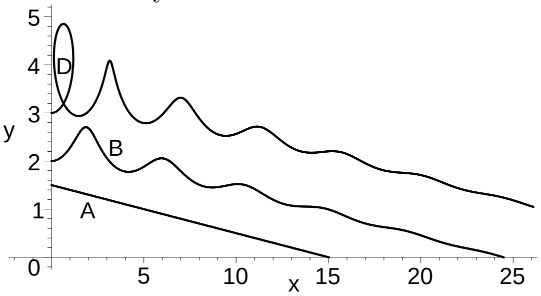




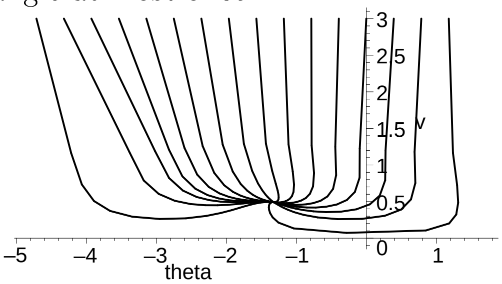
[5] We didn't address this point. If the trace is nonzero, it is impossible to have a center. But if the trace is zero, then the higher-order terms in the Taylor series play a role, and anything is possible. In fact, it can be shown that the fixed point for $R = 0$ is indeed a center. One way to do this is to show that the quantity $E(\theta, v) = v^3 - 3v\cos\theta$ must be constant for any solution when $R = 0$. Thus, the solutions are the level sets of E, which has a local minimum at $\theta = 0, v = 1$. Around a local minimum, the level sets are simple closed curves, so the fixed point is indeed a center.

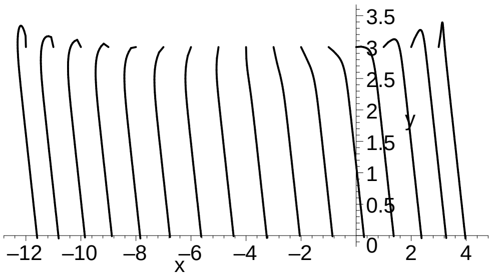
$0 < R < 2\sqrt{2}$**:** Here the fixed point (`A`) corresponds to a glider which dives with a constant velocity and not too steep an angle (the steepest angle is slightly less than $\pi/4$ below horizontal). The fixed point is a spiral sink: nearby initial conditions spiral into it (`B`). Such a spirals corresponds to a glider for which the angle and velocity constantly oscillate, but these oscillations get smaller and smaller as time passes, limiting on the same angle and velocity as the fixed point. Solutions with initial conditions further away (`D`) from the fixed point do some number of loops before settling into a pattern of oscillations which limit on the diving solution. Between the solutions which do k loops and those which do $k+1$ loops lies a solution which stalls out after k loops (`C`) — at some time the solution gets to $\theta = 2k\pi + \pi/2$ (i.e. going straight up) and velocity $v = 0$.





$R \geq 2\sqrt{2}$**:** The fixed point is a sink corresponding to a steeply diving solution.[6] Causing the glider to loop becomes increasingly difficult as R increases. For example when $R = 3$, if the initial angle θ is zero, an initial velocity of larger than 86.3 is required to get the glider to do one loop. If the initial angle differs from that of the fixed point, the glider fairly quickly turns towards that angle. No oscillation occurs; the angle can pass the limiting angle at most once.

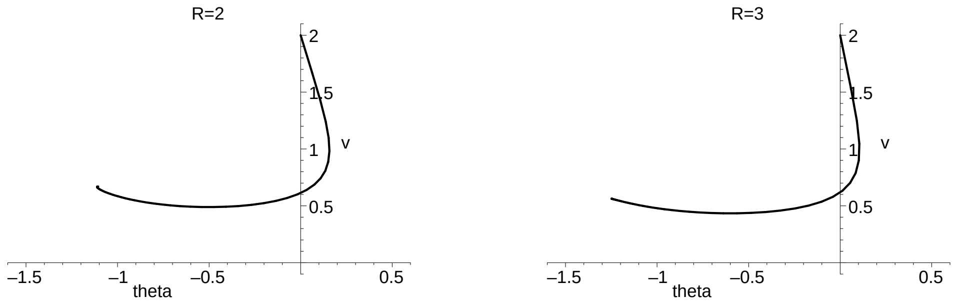




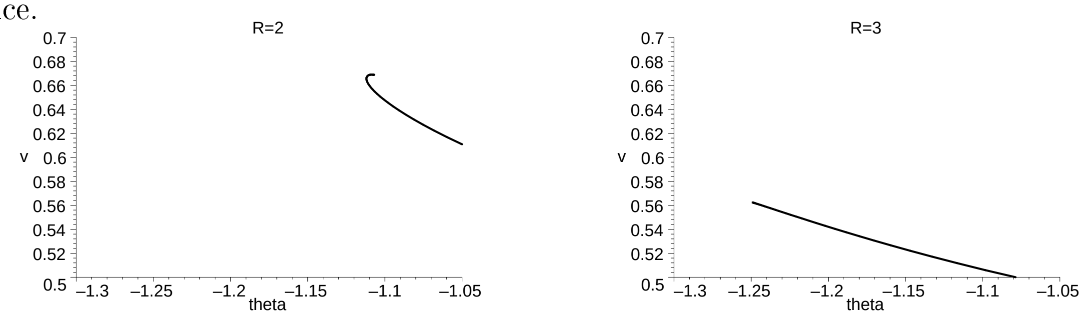
We have shown that for $R < 2\sqrt{2}$, the fixed point is a spiral sink: nearby solutions oscillate toward it. We should, however, point out that while the oscillations are always present mathematically, they can be very hard to discern. To emphasize this, we will compare a solution in the θ, v-plane for $R = 2$ (a spiral sink) and $R = 3$ (a sink with two real eigenvalues).

---

[6]For $R = 2\sqrt{2}$, the linearization corresponds to a degenerate case with a double eigenvalue, rather than a regular sink. However, what we say here still applies.



The two plots look rather similar. However, if we look closely at the solutions near the fixed point (we can use `zoom` to do this without recomputing the picture), we see a significant difference.



For $R = 2$, there is a “hook” at the end, but for $R = 3$, the solution goes straight in to the fixed point. Further magnifications show the same pattern, as you may want to verify for yourself. The spiral for $R = 2$ is always present, but very tight and hard to detect. Similarly, the oscillations in the x, y-plane when R is just slightly less than $2\sqrt{2}$ are nearly indiscernable, but present.

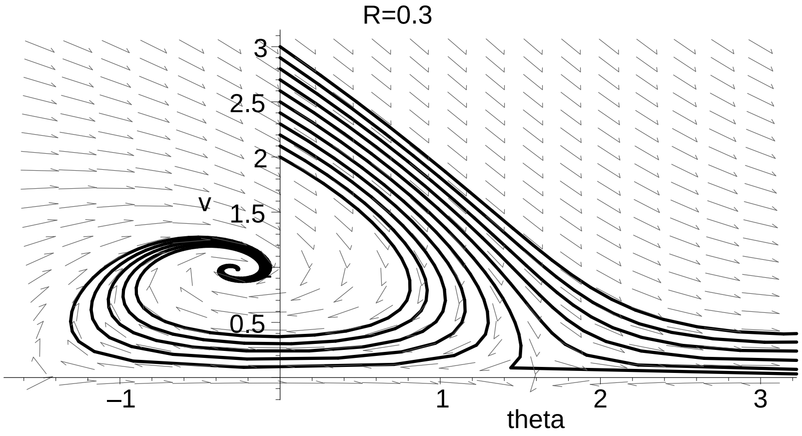
8 Dealing with the Singularity

The phugoid equations

$$\dot{\theta} = \frac{v^2 - \cos\theta}{v} \qquad \dot{v} = -\sin\theta - Rv^2$$

run into trouble when $v = 0$, because $\dot{\theta}$ becomes infinite as $v \to 0$. Since there are very few solutions for which v is ever zero, you might be tempted to ignore this issue, as we have done so far. But in fact, since we are solving the system numerically (rather than exactly), we run into problems even when v is merely small. So far, we have taken care to avoid such situations. But now let’s throw caution to the wind, and see what happens.

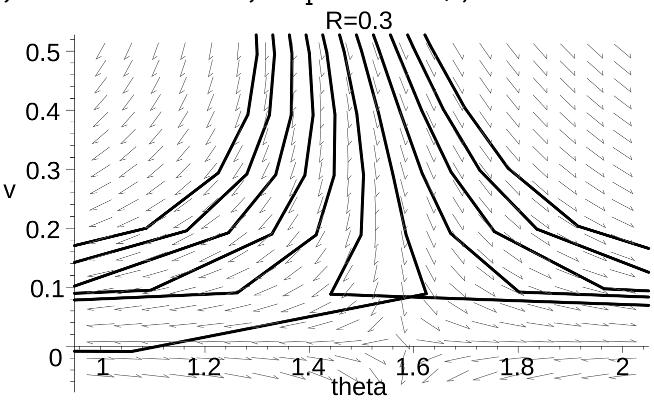
```
> R:=0.3:
  DEplot([diff(theta(t),t)=(v(t)^2 - cos(theta(t)))/v(t),
          diff(v(t),t)    = -sin(theta(t)) - R*v(t)^2],
    [theta,v], t=0..10,
    [seq([theta(0)=0,v(0)=2+i/10],i=0..10)],
    theta=-1.5..3, v=-0.05..3, title="R=0.3",linecolor=black,stepsize=0.1);
```



Something looks very wrong with the solution which starts at $\theta = 0, v = 2.5$. Notice how the solution makes a sharp turn, heading in a completely different direction from the vector field. Why is this happening?

Just to verify that it isn't a fluke, let's examine several solutions with initial conditions $\theta(0) = 0$ and $2.4 < v(0) < 2.6$, looking in the problem area. We'll use `obsrange=false` to tell Maple we want it to continue computing the solutions, even when they go outside of the viewport.

```
>  DEplot( [diff(theta(t),t) =  (v(t)^2 - cos(theta(t)))/v(t),
           diff(v(t),t)     =  -sin(theta(t)) - R*v(t)^2 ],
     [theta,v], t=0..10,
     [seq([theta(0)=0,v(0)=2.4+.02*i],i=0..10)],
     theta=1..2, v=-0.05..0.5, obsrange=false,
     title="R=0.3",linecolor=black,stepsize=0.1);
```



Even worse! Not only does the solution for $v(0) = 2.5$ go the wrong way, the one with $v(0) = 2.52$ also turns the wrong way, crossing over another solution. Since our equations are autonomous, this is *never* supposed to happen.

We can remedy this by decreasing the stepsize significantly. Decreasing the stepsize by a factor of 10 (to 0.01) in the above example gives solutions which do the right thing. But a significant price in computation must be paid, and this doesn't really solve the problem. There are other initial conditions nearby which lead to the same trouble.

In order to decide how to fix the problem, we must first understand what is going wrong. The first problem we noticed happened for a solution with $\theta(t_i) \approx 1.44$, $v(t_i) \approx 0.09$, using a stepsize $h = 0.1$. Since Maple is using a Runge-Kutta method as described in §4.2, let's calculate what is happening.

Recall that in Runge-Kutta, we evaluate the vector field at four points, and average them together to get the next point. We denoted these vectors as $\vec{k}_\ell$, $\vec{k}_c$, $\vec{k}_m$, and $\vec{k}_r$. Since we have to calculate the vector field at four points, we'll write a little function to do this for us, and use the fact that Maple lets us do aritmetic on lists as though they are actual vectors; that is, we can add them and multiply by scalars.[7]

```
> h:=0.1:
  VF:=p->[(p[2]^2 - cos(p[1]))/p[2], -sin(p[1])-R*p[2]^2]:

  p:=[1.44, 0.9];
  k[l]:=VF(p);
  k[c]:=VF(p+h/2*k[l]);
  k[m]:=VF(p+h/2*k[c]);
  k[r]:=VF(p+h*k[m]);
  step:=h/6*(k[l] + 2*k[c] + 2*k[m] + k[r]);
```

$$\begin{aligned} p &:= [1.44, 0.09] \\ k_l &:= [-1.359152319, -.9938883482] \\ k_c &:= [-4.858468860, -.9808007323] \\ k_m &:= [-8.872157510, -.9314790880] \\ k_r &:= [270.3559960, -.5250618133] \\ step &:= [4.025593182, -.08905849670] \end{aligned}$$

The problem here is the vector $\vec{k}_r$, which has a huge $\dot{\theta}$ component, and points in a different direction from the others. This happens because k_r is the value of the vector field calculated at $\theta \approx 1.35$, $v \approx -0.00315$. Not only is the value of v very small here (giving a huge $\dot{\theta}$ component), it is outside the range of allowed values, and the vector field points the wrong way. Similar problems will occur for any solution which comes too close to the $v = 0$ axis with $v(t)$ decreasing.

Now the question is how to avoid this? One not very satisfying answer is just to use a very small step size. But the problem with this idea is that we then must do a huge amount of computation, even when the vector field is behaving nicely.

A better solution is to use an *adaptive stepsize*: when the vector field is "nice", we can use a large step, and when the vector field is being troublesome, we use a small step. Maple has such methods built in to it, such as the Runge-Kutta-Fehlberg method, which we can use by appending `method=rkf45` to the list of options to `DEplot`.

Sometimes, it is not always possible to change the numerical method. In the case of the phugoid, we can accomplish the same goal (in fact, gain something extra) by adjusting the

[7] This ability is not present in releases prior to Maple 6, making this computation a little more cumbersome.

length of the vectors that make up the vector field.

If we think of our solution curves as being parametric curves with tangent vectors which coincide with the vector field, then changing the lengths of the vectors (but not their direction) will not change the solution *curves*, only the parameterization. That is, we can rescale time so that the vectors no longer blow up as solutions approach the $v = 0$ line.

Multiplying each vector by $v(t)$ cancels out the problems in $\dot{\theta}$, and gives a system which has the same solution curves as long as $v > 0$. Thus, we get the new system

$$\dot{\theta} = v^2 - \cos\theta \qquad \dot{v} = -v\sin\theta - Rv^3 \qquad \dot{x} = v^2\cos\theta \qquad \dot{y} = v^2\sin\theta$$

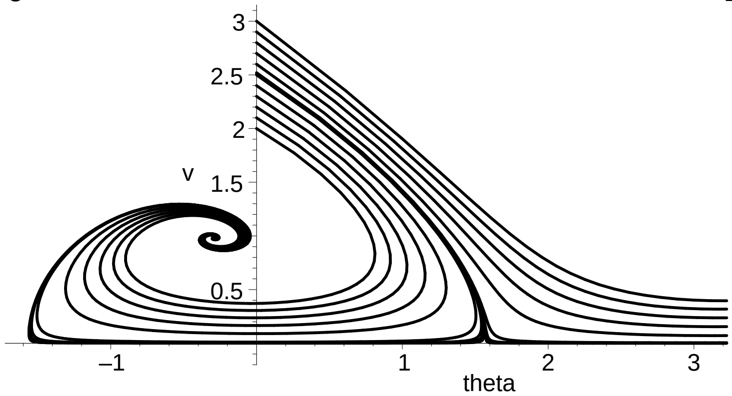
This new system of equations is defined and continuous for all values of v and θ.

Furthermore, if we want to keep track of the original time variable t, we can add a new variable T. In the original, non-rescaled equations, the time could be described as $T(t) = t$, or equivalently, $\dot{T} = 1$. After rescaling the vector field by the factor of v, this gives $\dot{T} = v$.

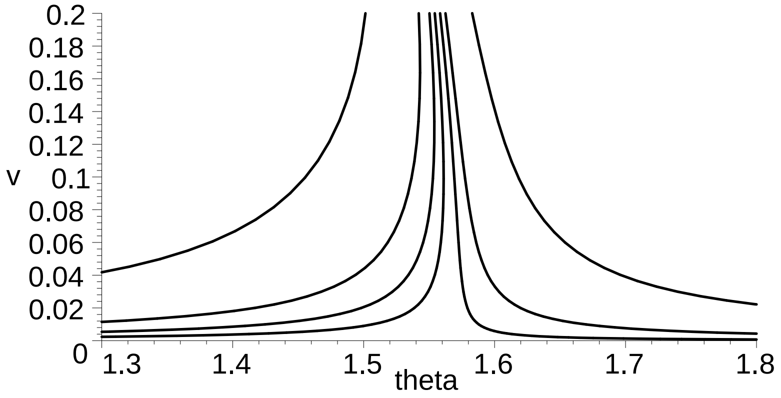
```
> R:='R';
  vphug:= [ diff(theta(t),t) =  v(t)^2 - cos(theta(t)),
            diff(v(t), t)    = -v(t)*sin(theta(t)) - R*v(t)^3,
            diff(x(t), t)    =  v(t)^2 *cos(theta(t)),
            diff(y(t), t)    =  v(t)^2 *sin(theta(t)),
            diff(T(t), t)    =  v(t) ]:

  R:=0.3:

  DEplot(vphug, [theta, v, x, y, T], t=0..20,
         [seq([theta(0)=0,  v(0)=2+i/10, x(0)=0, y(0)=5, T(0)=0],i=0..10),
              [theta(0)=0,  v(0)=2.51, x(0)=0, y(0)=5, T(0)=0],
              [theta(0)=0,  v(0)=2.52, x(0)=0, y(0)=5, T(0)=0],
           seq([theta(0)=0,  v(0)=2.51+i/1000, x(0)=0, y(0)=5, T(0)=0],i=2..5) ],
        theta=-1.5..3, v=-0.05..3, x=0..10, y=0..5,  T=0..20,
        obsrange=false, linecolor=black, scene=[theta,v], stepsize=0.1);
```




```
> zoom(%, 1.3..1.8, 0..0.2);
```



As you can see, using the desingularized equations solves the problems for small values of v.

In addition, we can see additional structure not apparent in the original equations. There are new fixed points at $v = 0$, $\theta = \pm\pi/2$. These fixed points are saddles, as we can see by looking at the Jacobian. In the desingularized equations, the Jacobian is

$$\begin{pmatrix} \sin\theta & 2v \\ -v\cos\theta & -\sin\theta - 3Rv^2 \end{pmatrix}$$

so the linearization at $(\frac{\pi}{2}, 0)$ is $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and at $(-\frac{\pi}{2}, 0)$ we have $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$. These fixed points correspond to the solutions which stall out, and as we surmised earlier, they divide up the qualitative behaviors. The eigenvectors for both of them are parallel to the θ and v axes, which says that solutions tending to a stall (or leaving one) do so by flying nearly straight up (resp. down) with an angle of $\pi/2$ (resp. $-\pi/2$). Solutions with very small velocity and an angle slightly less than $\pi/2$ flip over rapidly to an angle slightly more than $-\pi/2$ and then begin to increase their velocity.

While it is possible to deduce this behavior from the original equations, it is much more obvious one the singularity is filled in. Similar techniques are used in other branches of mathematics (blowing up a singularity in algebraic geometry, construction of a collision manifold in celestial mechanics) with great success.

References

[BDH] P. Blanchard, R. Devaney, G. R. Hall, *Differential Equations.* Brooks/Cole, 1998.

[EP] C. H. Edwards, D. E. Penney, *Differential Equations & Linear Algebra.* Prentice-Hall, 2001.

[HW] J. H. Hubbard, B. H. West, *Differential Equations: A Dynamical Systems Approach. Part 1: Ordinary Differential Equations*, Texts in Applied Mathematics **5**. Springer-Verlag, New York. 1991. *Part 2: Higher Dimensional Systems*, Texts in Applied Mathematics **18**. Springer-Verlag, New York. 1995.

[Lan] F. W. Lanchester, *Aerial Flight Vol II: Aerodonetics*. Colchester & Co., London. 1908.