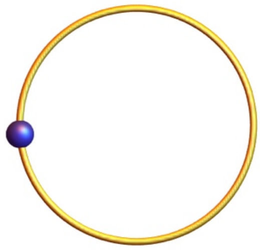
**Finding the Constraint Forces: Bead on a Wire**

What we'll see in this article is a quick overview of the "bead on the wire" problem. This is a classic example that almost every book mentions when they speak of constraints. You can see it as the "hello, world!" of constraints.

The traditional problem consists of a bead moving along a wire. The wire allows for circular movement around it's center, and the bead can move freely on the wire. We can say that the bead movement is *constrained* to the wire.



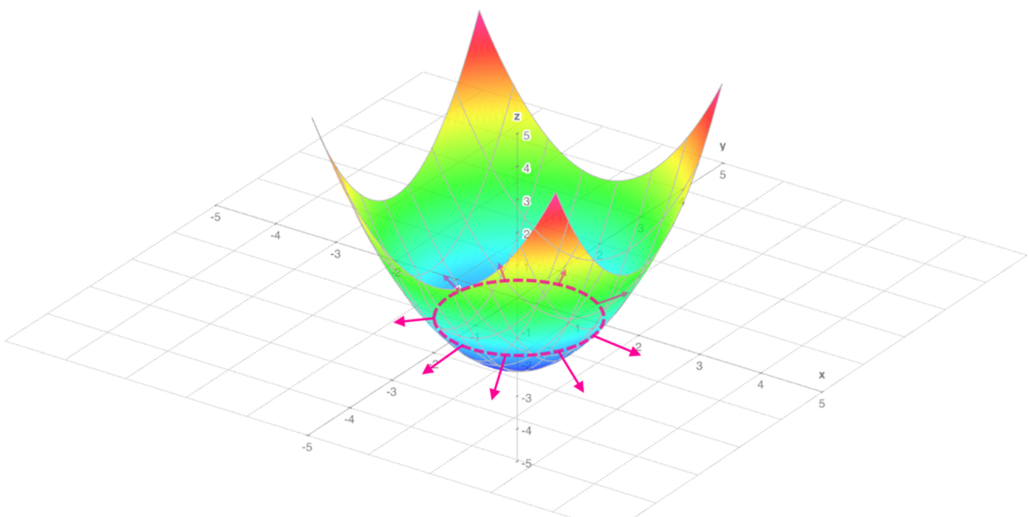
So, let's set our goals. What we are looking are the "*internal forces*" that the wire applies on the bead, regardless of what "*external forces*" are (wind, friction, weight, etc.). This means that what we want to find is the **magnitude** and the **direction** of this internal force.

Also, do you see how I am talking about "constraint force"? We'll look at this example from the point of view of *forces*, but keep in mind that in our physics engine, we ultimately want the magnitude and the direction of an ***impulse***. But for now, let's stick with forces.

We'll start out problem by first defining a position constraint function. Since this will be an equality constraint, if we consider the position of the bead at $(x, y)$, and the constant *length* from the origin as being $l$, we can state that:

$$x^2 + y^2 - l^2 = 0$$

If we set our constraint *length* as 2, for example, the graph of $C$ looks like this:



The *magenta* ring are the points where the position constraint function $C = 0$. This magenta ring represents the set of points with all the *legal* positions of our bead.

The magenta arrows represent the *gradients* of $C$, and they indicate the direction of *illegal* positions of our bead, where $C \neq 0$. If the bead moves along these arrows in any way, either positive direction (away from the origin) or negative direction (towards the origin), it will break our constraint. Therefore, we can intuitively already tell that the constraint forces we must generate to push the bead will be parallel to these arrows.

Let us rewrite our position constraint function in vector form. Our bead position at $(x, y)$ will be the vector position $\vec{r}$. We can now rewrite our constraint function in vector form:

$$C(\vec{r}) = \vec{r} \cdot \vec{r} - l^2 = 0$$

Now that we have our *position* constraint function $C$, we need to remember that we are usually not interested in changing the position directly. Our goal is to aim for a change in the velocity domain, which usually means more stability to our physics engine. This means we'll have to look at *first time derivative* of the position (you can read this as "the rate of change of the position with respect to time").

We are interested in the first derivative of the position, $\dot{C}$:

$$\dot{C} = \frac{dC}{dt} = \frac{d(\vec{r} \cdot \vec{r})}{dt} - \frac{d(l^2)}{dt} = 0$$

Let's think of what this means. We are differentiating the position function C with respect to time, which means we can differentiate both terms of the subtraction.

Now here is where all those *calculus* classes they force you to take in college come in handy. We (should) learn in our calculus classes how to analytically solve the *derivatives* of different types of functions. Since what we want is to find the first derivative (the rate of change) of the position function $C$ with respect to time, we will have to use some of that calculus knowledge here.

I do remember (at least) that the derivative of any *constant* is always zero. That means that we don't have to worry about the derivative of that length $l$ squared. All we need is to know how to take the derivative of the dot product between vectors.

$$\dot{C} = \frac{dC}{dt} = \frac{d(\vec{r} \cdot \vec{r})}{dt} - 0 = 0$$

I don't remember how to analytically solve most derivatives by heart, so I usually try to look for help and hints in my old calculus books or online articles. I did exactly that, and I found that the derivative of the dot product between any two vectors $\vec{a}$ and $\vec{b}$ is given by:

$$\frac{d(\vec{a} \cdot \vec{b})}{dt} = \frac{d\vec{a}}{dt} \cdot \vec{b} + \vec{a} \cdot \frac{d\vec{b}}{dt}$$

So, let's go ahead and use this information to compute $\dot{C}$.

$$\dot{C} = \frac{dC}{dt} = \frac{d\vec{r}}{dt} \cdot \vec{r} + \vec{r} \cdot \frac{dr}{dt} - 0 = 0$$

$$\dot{C} = \frac{dC}{dt} = 2 * \left(\vec{r} \cdot \frac{d\vec{r}}{dt}\right) = 0$$

Now, think about what the term $\frac{d\vec{r}}{dt}$ means. We are looking at the first derivative of the bead position $\vec{r}$ with respect to time. Well, the rate of change of the bead position is its **velocity**, do you agree?

$$\dot{C} = \frac{dC}{dt} = 2 * (\vec{r} \cdot \vec{v}) = 0$$

Alright, so we got our first time derivative. But since we are looking for the internal ***forces*** that we need to apply to restrict the bead motion along the wire, we should really be looking for a change in the *acceleration* domain. After all, forces and acceleration are directly connected by Newton's second law of motion. That means we'll have to also compute the *second time derivative* of our position constraint function, $\ddot{C}$.

$$\ddot{C} = \frac{d^2C}{dt} = \frac{d(2 * \vec{r} \cdot \vec{v})}{dt} = 0$$

The above formula states that we are aiming for the second time derivative of the position (our acceleration) to be equal to zero.

Once again, calculus to the rescue. We can pull the constant 2 outside, and use the same dot product derivative we used before to

$$\ddot{C} = \frac{d^2C}{dt} = 2 * \frac{d\vec{r}}{dt} \cdot \vec{v} + 2 * \vec{r} \cdot \frac{d\vec{v}}{dt} = 0$$

We know already that the rate of change $\frac{d\vec{r}}{dt}$ is the ***velocity*** of the bead at a certain position $\vec{r}$. And we also know that $\frac{d\vec{v}}{dt}$ is the rate of change of the velocity of the bead with respect to time, which is the ***acceleration*** of the bead. Therefore, we can eeplace both terms to get to our final equation for the second time derivative:

$$\ddot{C} = \frac{d^2C}{dt} = 2 * \vec{v} \cdot \vec{v} + 2 * \vec{r} \cdot \vec{a}$$

Look at that!

We have just found the second time derivative of $C$. Which, in other words, means we have just found our **acceleration** constraint.

Alright! Let's keep our eyes on the prize. Let's go back and look again at our problem from the perspective of the internal and external forces that are being applied on the bead, and how we can use the acceleration we found above to compute both the *magnitude* and the *direction* of the **constraint force** we need to apply to keep our bead always on the wire.

We know that the sum of all forces is the total force of the system. The total force must consider both internal and external forces:

$$\vec{F} = m\vec{a}$$

$$\vec{F}_{int} + \vec{F}_{ext} = m\vec{a}$$

$$\vec{a} = \frac{\vec{F}_{int} + \vec{F}_{ext}}{m}$$

Let's replace this in our formula for the second derivative,

$$\vec{v} \cdot \vec{v} + \vec{r} \cdot \left( \frac{\vec{F}_{int} + \vec{F}_{ext}}{m} \right)$$

We can use some simple algebra to rearrange our equation. We are looking for the internal forces (the constraint forces) that will keep our bead constrained to the wire.

$$\vec{r} \cdot \left( \vec{F}_{int} + \vec{F}_{ext} \right) = -m * \vec{v} \cdot \vec{v}$$

$$\vec{F}_{int} \cdot \vec{r} \ + \ \vec{F}_{ext} \cdot \vec{r} = -m * \vec{v} \cdot \vec{v}$$

$$\vec{F}_{int} \cdot \vec{r} = -m * \vec{v} \cdot \vec{v} - \ \vec{F}_{ext} \cdot \vec{r}$$

Now we'll use something called Lagrange-D'Alamber principle. This states that constraint forces do not perform *work*, and do not add energy to the system. Looking at the *energy* state of a system and deriving properties from it is a very common trait from *Lagrangian Mechanics*.

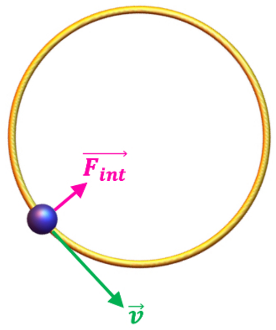
In physics, *Work* is defined as being $W = \vec{F} * \Delta d$, where $\vec{F}$ is the force applied, and $\Delta d$ is the displacement (the distance traveled).

Since the virtual work must be zero, then at any given instant we have:

$$\vec{F}_{int} \cdot \vec{v} = 0$$

Look beyond the above equation and think of what it means. The *dot product* between the constraint force and the bead velocity vector is zero. We learned that the dot product of two vectors is only zero if they are perpendicular to each other, correct?

We know that the direction of our constraint force is always *perpendicular* to the velocity of the bead, and I believe intuitively this makes sense. The constraint force that keeps the bead on the wire will always be *parallel* to the arrow pointing away or towards the origin.




Adding what we already know from our velocity constraint, which states that $2 * \vec{r} \cdot \vec{v} = 0$. We can write the internal force (our constraint force) as:

A scalar $\lambda$ that represents the **magnitude** of the force

$$\vec{F}_{int} = \lambda * \vec{r}$$

The **direction** of the force, which we intuitively know is parallel to $\vec{r}$ (the position of the bead)

What the formula above is saying is that the internal force is a multiple of the direction vector $\vec{r}$, and $\lambda$ is the scalar multiplier (sometimes called the *Lagrange multiplier*).

The scalar $\lambda$ is the **magnitude** of the constraint force that we need to apply to keep our constraint equal to zero. Plugging this back into the acceleration constraint equation, we can isolate lambda to get:

$$\lambda = -\frac{m * \vec{v} \cdot \vec{v} + \vec{F}_{ext} \cdot \vec{r}}{\vec{r} \cdot \vec{r}}$$

And there you go! We have a formula to find the magnitude $\lambda$ of our constraint force. Since we derived this in vector notation, this will work for both 2D and 3D. Just keep in mind that in 3 dimensions we probably won't have a bead on a wire, but a bead on a sphere.

This was a long journey, but we finally have the **direction** and the **magnitude** of the constraint force we need to apply to keep the bead constrained to the wire. We can apply the same process to any other type of constraint. But you'll see that we will not do that.

In the next lectures, we'll learn how to *generalize* velocity constraints. That means we'll have a framework that will work for any type of constraint, and that will definitely make our lives a lot easier when we translate all of this into code.