```julia
begin
    import Pkg
    Pkg.activate(mktempdir())
    Pkg.add(url="https://github.com/brendanjohnharris/NonstationaryProcesses.jl")

    Pkg.add("DifferentialEquations")
    using NonstationaryProcesses, DifferentialEquations, Plots
    plotlyjs(); fourseas!(); # Plots initialisation
end
```

Click here to view in a web browser

Question 3

The equations of motion in the Schwarzschild metric (question 2) are:

$$a^t = -\frac{2m}{r}\left(1-\frac{2m}{r}\right)^{-1} u^t u^r,$$

$$a^r = -\frac{m}{r^2}\left(1-\frac{2m}{r}\right)(u^t)^2 + \frac{m}{r^2}\left(1-\frac{2m}{r}\right)^{-1}(u^r)^2 + (r-2m)(u^\phi)^2,$$

$$a^\phi = -\frac{2}{r}u^\phi u^r,$$

where θ has been set to $\frac{\pi}{2}$ for motion in the equatorial plane.

a)

Given:

$$x_0^\alpha = (0, R, \frac{\pi}{2}, 0),$$

and that:

$$u_0^\alpha = (u_0^t, 0, 0, C),$$

the initial condition for u^t can be calculated from the normalisation of the four-velocity. .

The metric is:

$$g_{\alpha\beta} = \begin{bmatrix} -\left(1 - \frac{2m}{r}\right) & 0 & 0 & 0 \\ 0 & \left(1 - \frac{2m}{r}\right)^{-1} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2\sin^2(\theta) \end{bmatrix},$$

and the norm of the four-velocity is:

$$\mathbf{u} \cdot \mathbf{u} = g_{\alpha\beta} u^\alpha u^\beta,$$

Substituting the initial conditions:

$$\mathbf{u} \cdot \mathbf{u} = -\left(1 - \frac{2m}{R}\right)(u_0^t)^2 + R^2 (u_0^\phi)^2$$

$$\implies \mathbf{u} \cdot \mathbf{u} = -\left(1 - \frac{2m}{R}\right)(u_0^t)^2 + R^2 C^2$$

$$\implies \left(1 - \frac{2m}{R}\right)(u_0^t)^2 = R^2 C^2 - \mathbf{u} \cdot \mathbf{u}$$

$$\implies (u_0^t)^2 = \frac{R^2 C^2 - \mathbf{u} \cdot \mathbf{u}}{1 - \frac{2m}{R}}$$

$$\implies (u_0^t)^2 = \frac{(R^2 C^2 - \mathbf{u} \cdot \mathbf{u})R}{R - 2m}$$

$$\implies u_0^t = \sqrt{\frac{(R^2 C^2 - \mathbf{u} \cdot \mathbf{u})R}{R - 2m}}$$

Taking the positive solution so that the coordinate time and the proper time pass in the same direction.

c)

First set the parameters and initial conditions:

```
m, R = 1.0, 10.0;
```

```
uᵗ₀ (generic function with 1 method)
```

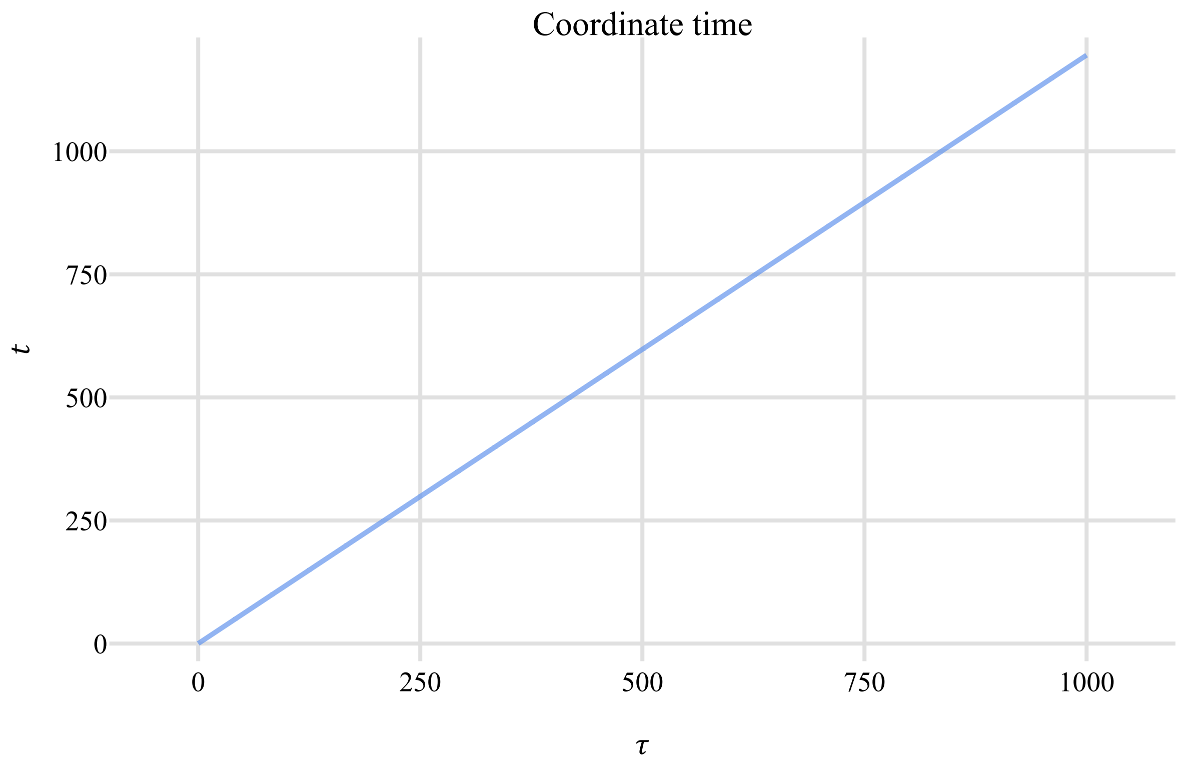
```
uᵗ₀(C, uu) = sqrt(R*(R^2*C^2 - uu)/(R - 2*m))
```

Then the equations of motion, in a function for convenience::

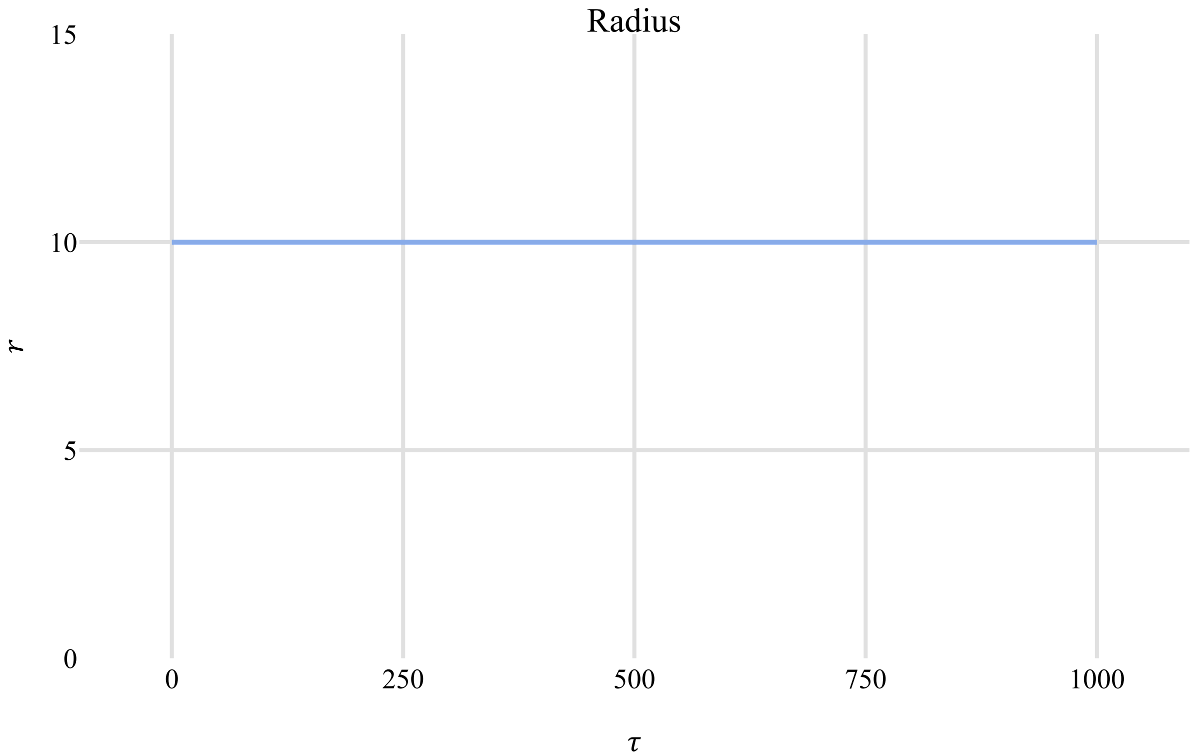
```
function f(C, uu, tmax)
    dXdτ((t, r, ϕ, uᵗ, uʳ, uᵠ), (m, C, R), τ) = [uᵗ, uʳ, uᵠ,
            (-2m/r^2)*(1-2m/r)^(-1)*uᵗ*uʳ,                                            # aᵗ
            (-m/r^2)*(1-2m/r)*uᵗ^2 + (m/r^2)*(1-2m/r)^(-1)*uʳ^2 + (r - 2*m)*uᵠ^2, # aʳ
            (-2/r)*uʳ*uᵠ ];                                                          # aᵠ

    🚀 = Process(
      process = dXdτ(P) = process2solution(P),
      parameter_profile = [m, C, R],
      varnames = [:t, :r, :ϕ, :uᵗ, :uʳ, :uᵠ],
      X0       = [0.0, R, 0.0, uᵗ₀(C, uu), 0, C],
      t0 = 0.0,
      tmax = tmax,
      alg = Vern9(),
      solver_opts = Dict(:adaptive => true, :reltol => 1e-10, :abstol => 1e-10));

    return [times(🚀), (eachcol∘timeseries)(🚀)...]
end;
```

i) $C = (10\sqrt{7})^{-1}$

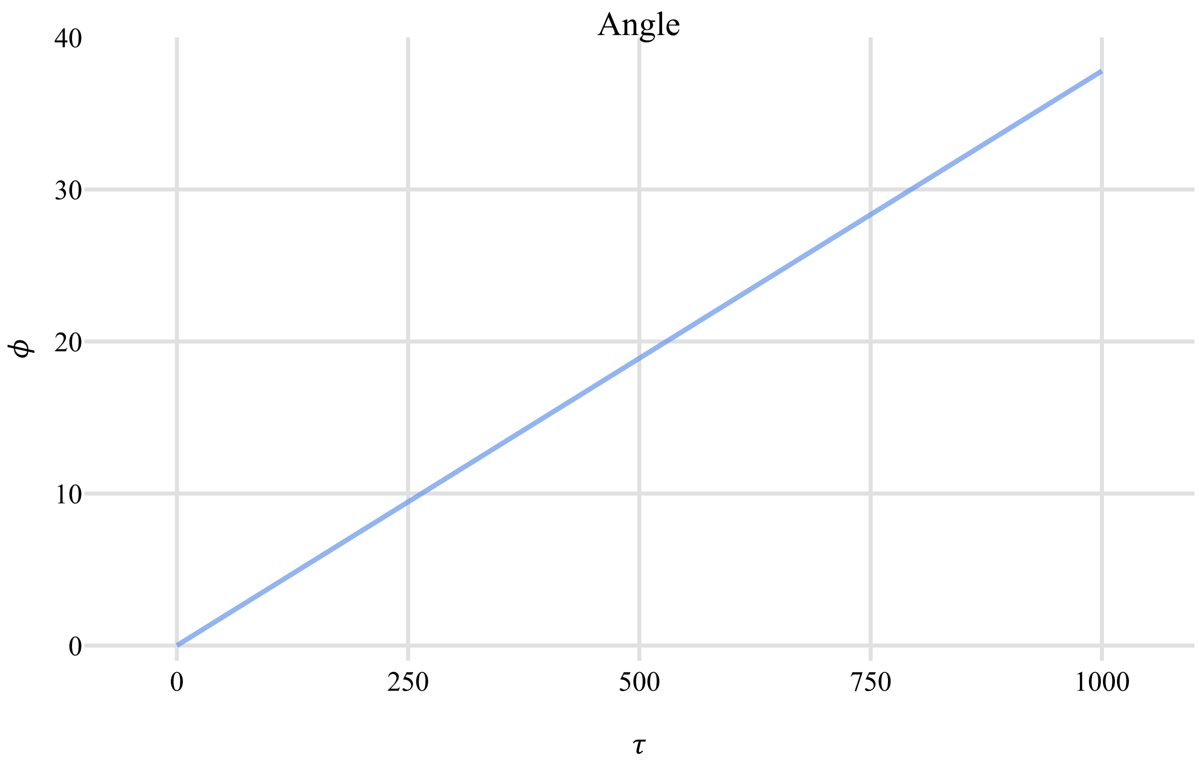
```
τ₁, t₁, r₁, ϕ₁, uᵗ₁, uʳ₁, uᵠ₁ = f((10*sqrt(7))^-1, -1, 1000);
```



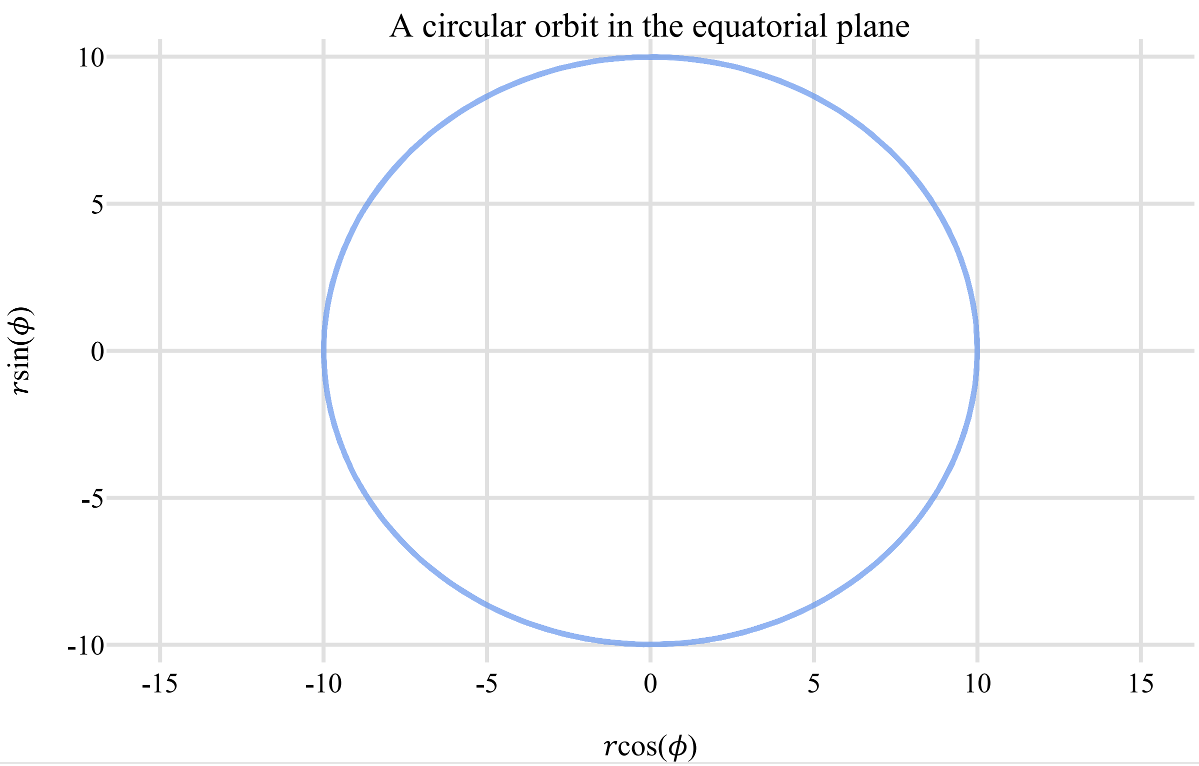

```
plot(τ₁, t₁, xguide="τ", yguide="t", xlims=(-100, 1100),
    title="Coordinate time")
```




```
plot(τ₁, r₁, xguide="τ", yguide="r", xlims=(-100, 1100), ylims=(0, 15),
    title="Radius")
```



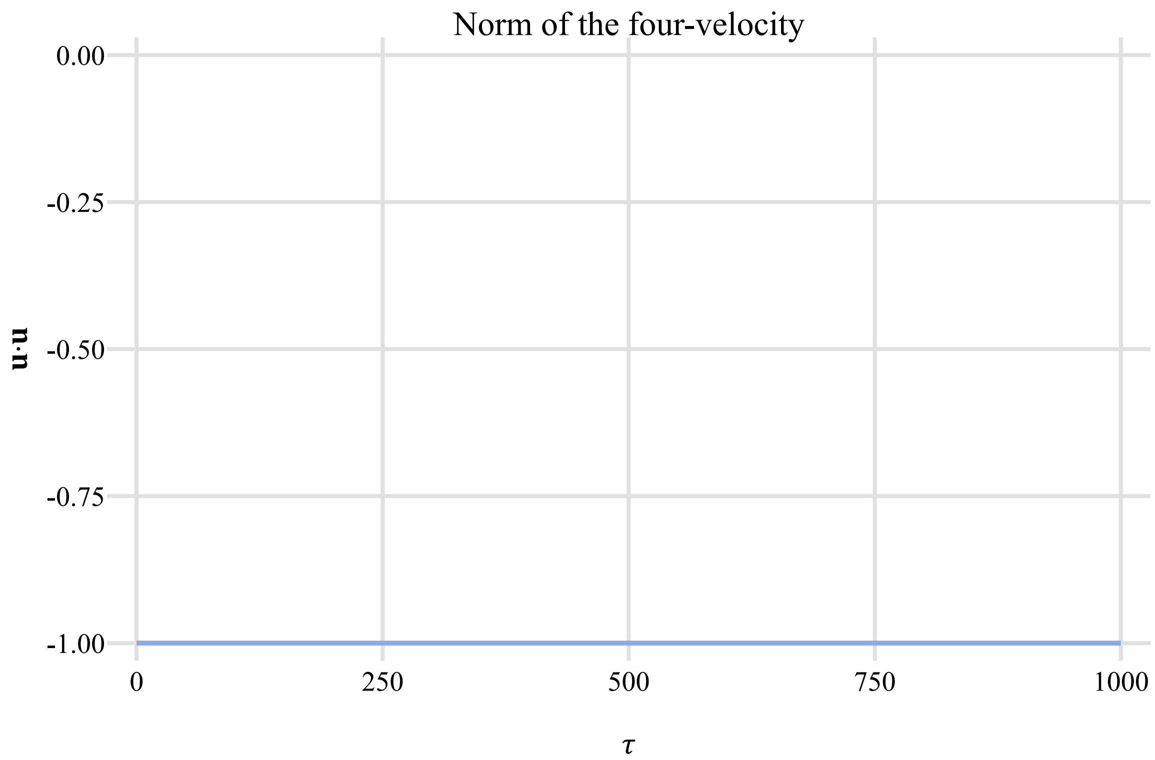

```
plot(τ₁, ϕ₁, xguide="τ", yguide="ϕ", xlims=(-100, 1100), ylims=(-1, 40),
    title="Angle")
```




```
plot(r₁.*cos.(ϕ₁), r₁.*sin.(ϕ₁), xguide="rcos(ϕ)", yguide="rsin(ϕ)",
        aspect_ratio=:equal, title="A circular orbit in the equatorial plane")
```

Using the metric and equation for the norm of the four-velocity written in part a):

```
uu(r, uᵗ, uʳ, uᵠ) = -(1-2m/r)*uᵗ^2 + (1-2m/r)^(-1)*uʳ^2 + r^2*uᵠ^2;
```

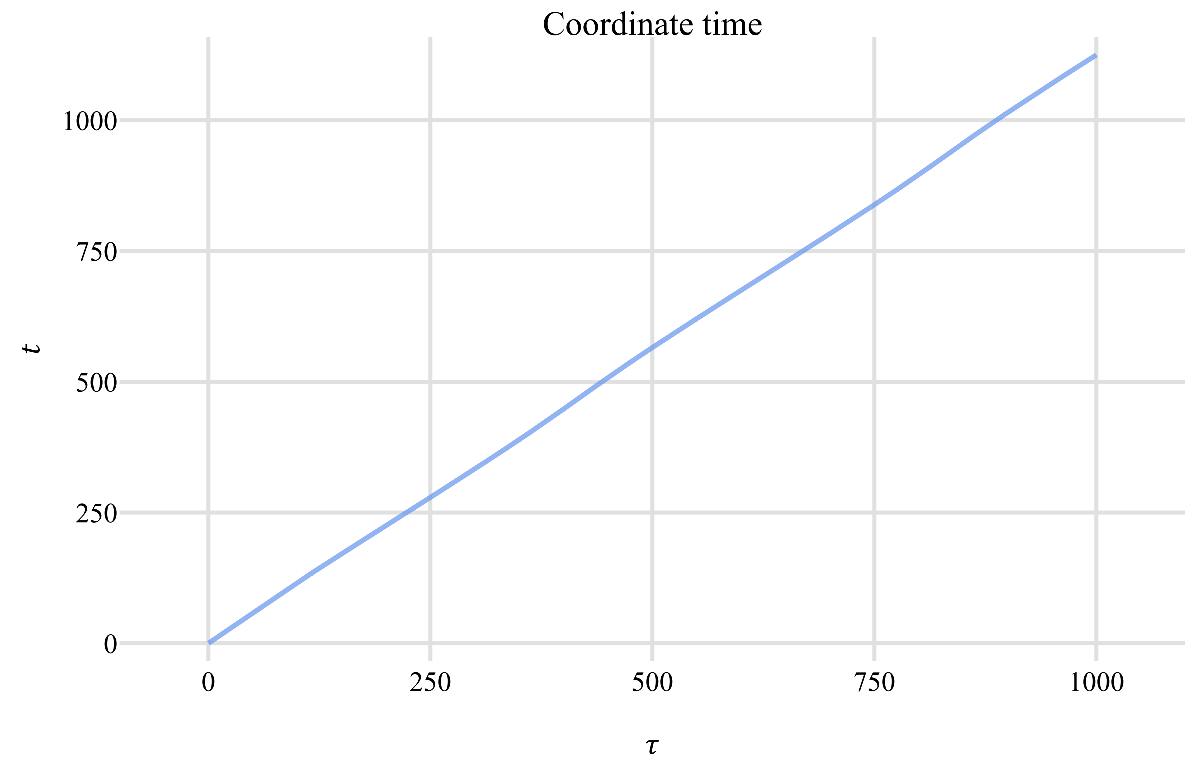



```
plot(τ₁, uu.(r₁, uᵗ₁, uʳ₁, uᵠ₁), xguide="τ", yguide="u·u", title="Norm of the four-velocity")
```

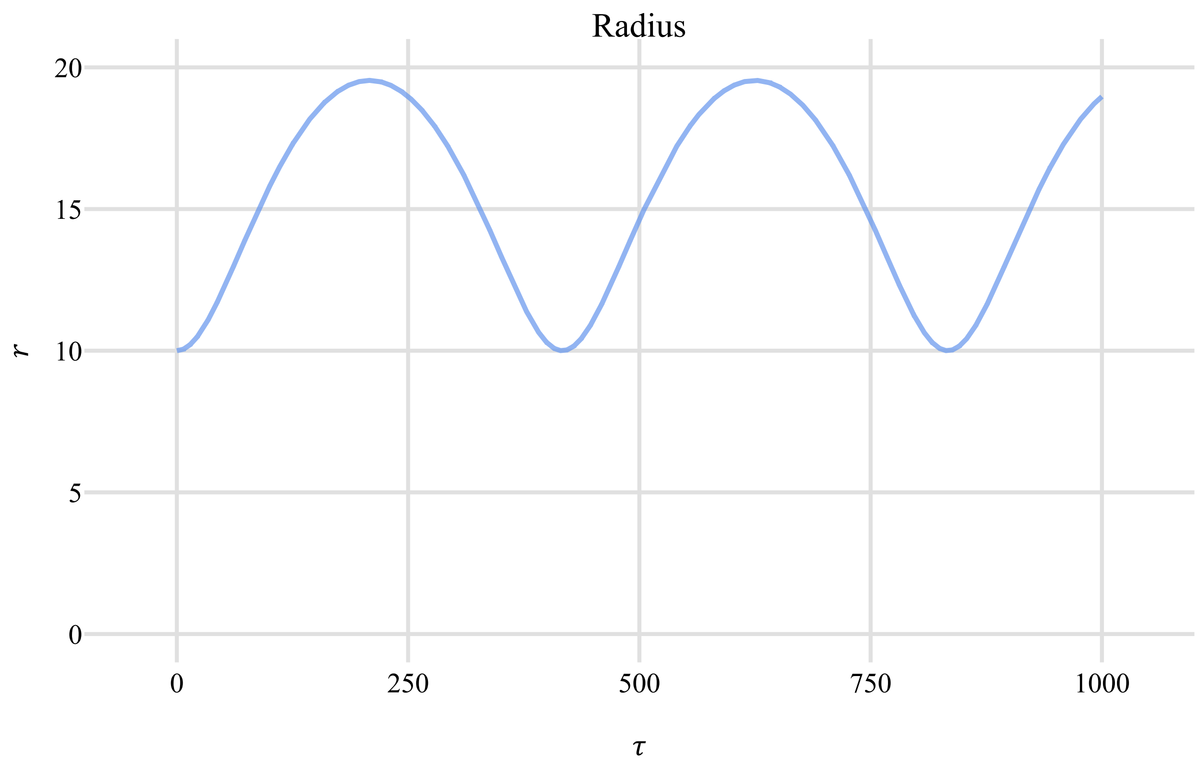
Note that with `Vern9()` the error in the normof the four-velocity is to small to be appear in float values. For a poorer integrator, such as the midpoint method, it is on the order of 10^{-16}.

ii) $C = 1.1(10\sqrt{7})^{-1}$

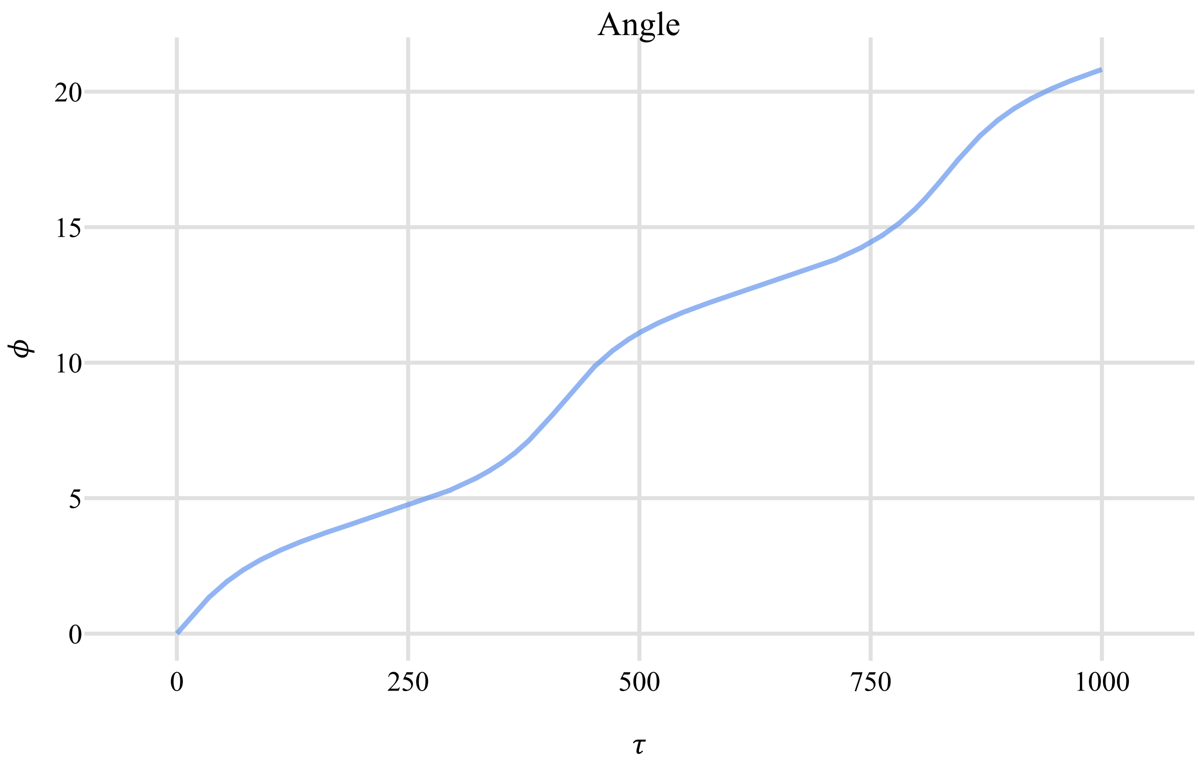
```
τ₂, t₂, r₂, ϕ₂, uᵗ₂, uʳ₂, uᵠ₂ = f(1.1*(10*sqrt(7))^-1, -1, 1000);
```



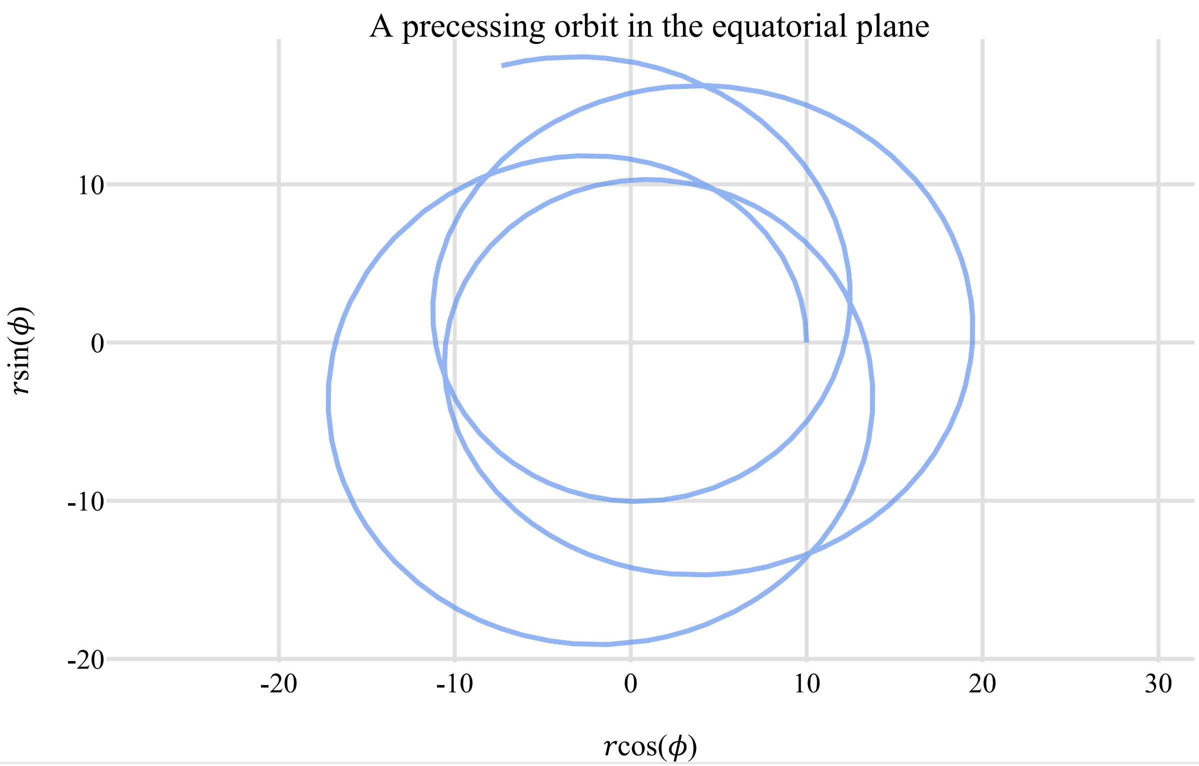

```julia
plot(τ₂, t₂, xguide="τ", yguide="t", xlims=(-100, 1100),
    title="Coordinate time")
```



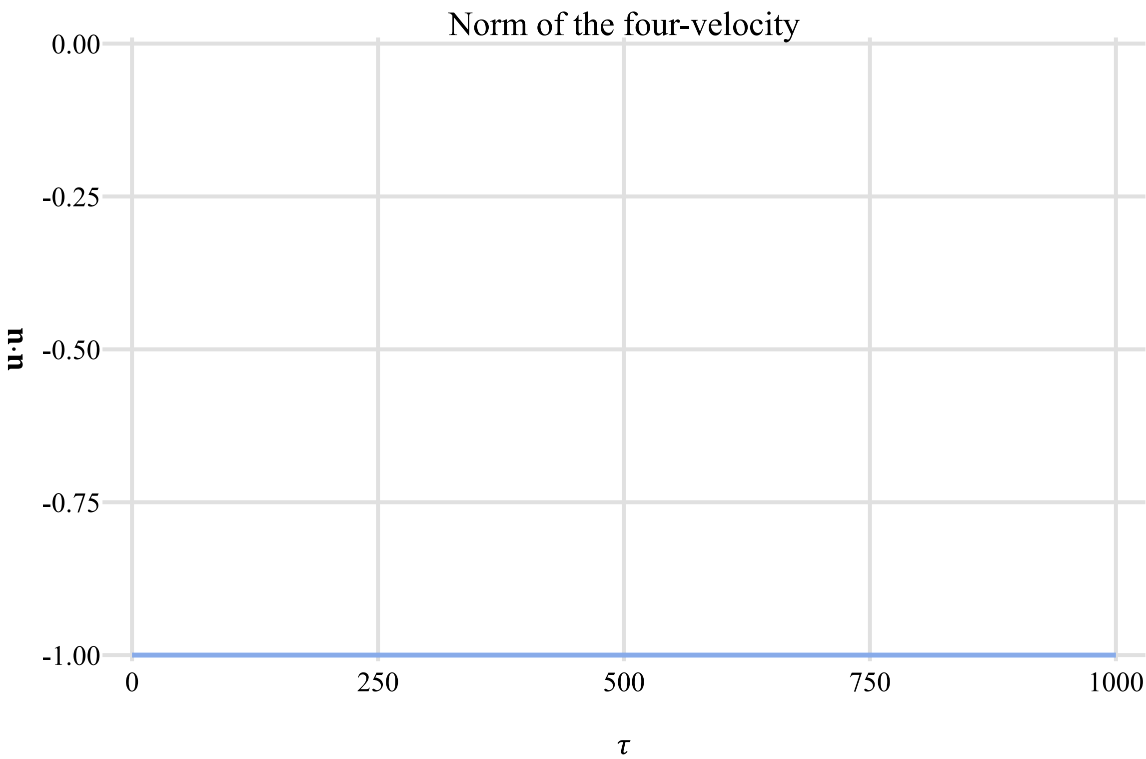

```julia
plot(τ₂, r₂, xguide="τ", yguide="r", xlims=(-100, 1100), ylims=(-1, 21),
    title="Radius")
```




```
plot(τ₂, ϕ₂, xguide="τ", yguide="ϕ", xlims=(-100, 1100), ylims=(-1, 22),
    title="Angle")
```

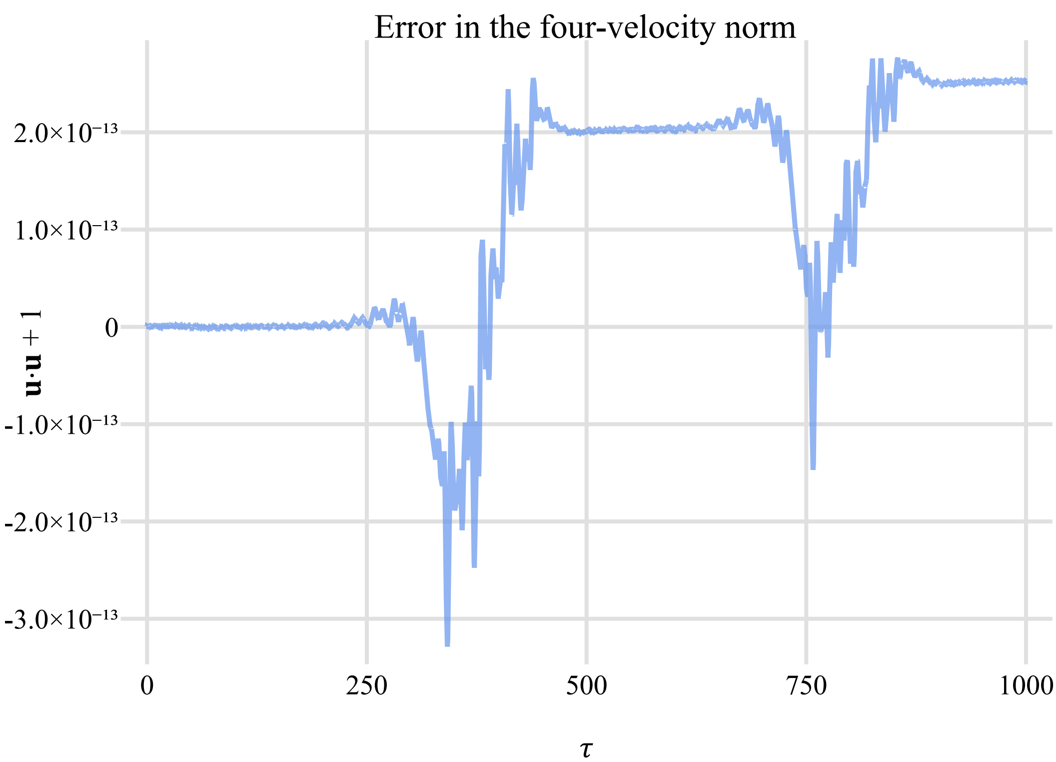



```
plot(r₂.*cos.(ϕ₂), r₂.*sin.(ϕ₂), xguide="rcos(ϕ)", yguide="rsin(ϕ)",
            aspect_ratio=:equal, title="A precessing orbit in the equatorial plane")
```




```
plot(τ₂, uu.(r₂, uᵗ₂, uʳ₂, uᵠ₂), xguide="τ", yguide="u·u", title="Norm of the four-velocity", ylims=(-1.01, 0.01))
```

This time the error is slightly greater, since the orbit is no longer a perfect circle:




```
plot(τ₂, uu.(r₂, uᵗ₂, uʳ₂, uᵠ₂) .+ 1, xguide="τ", yguide="u·u + 1", title="Error in the four-velocity norm")
```

This orbit precesses, unlike orbits under Netwonian gravity which are strictly conic sections. This is because general relativity, or the procedure to derive the path of an object from geodesic equaitons and the structure of spacetime, introduces an additional term to the effective potential of Newtonian gravity. Both descriptions of gravity have stable minima at someradius that is determined by the mass of the central body and the angular momentum of the orbiting object. Objects with apsides at this stable radius will follow a circular path, as for the first orbit shown in this question; their efective energy is minimised. Objects with apsides displaced from this stable minimum will precess; their effective energy is greather than the potential at the stable minimum. Their radius will oscillate between a point with an r smaller than the stable minimum, and one with an r greater (i.e. an elliptical orbit). This occurs in both Newtonian gravity and general relativity. However, Newtonian gravity has an effective potential with two terms: a radial force and an angular momentum. Hence elliptical orbits under Newtonian gravity are angularly stable and do not precess. General relativity introduces a third term that is cubic in the radius and also depends on angular momentum; this term breaks the (angular) stability of elliptical orbits and causes them to precess (the angle ϕ of the apsides is not constant).

(The effective potential described by general relativity also approaches negative infinity near the origin, so there are radially unstable states that do not appear in this assignment.)

d)

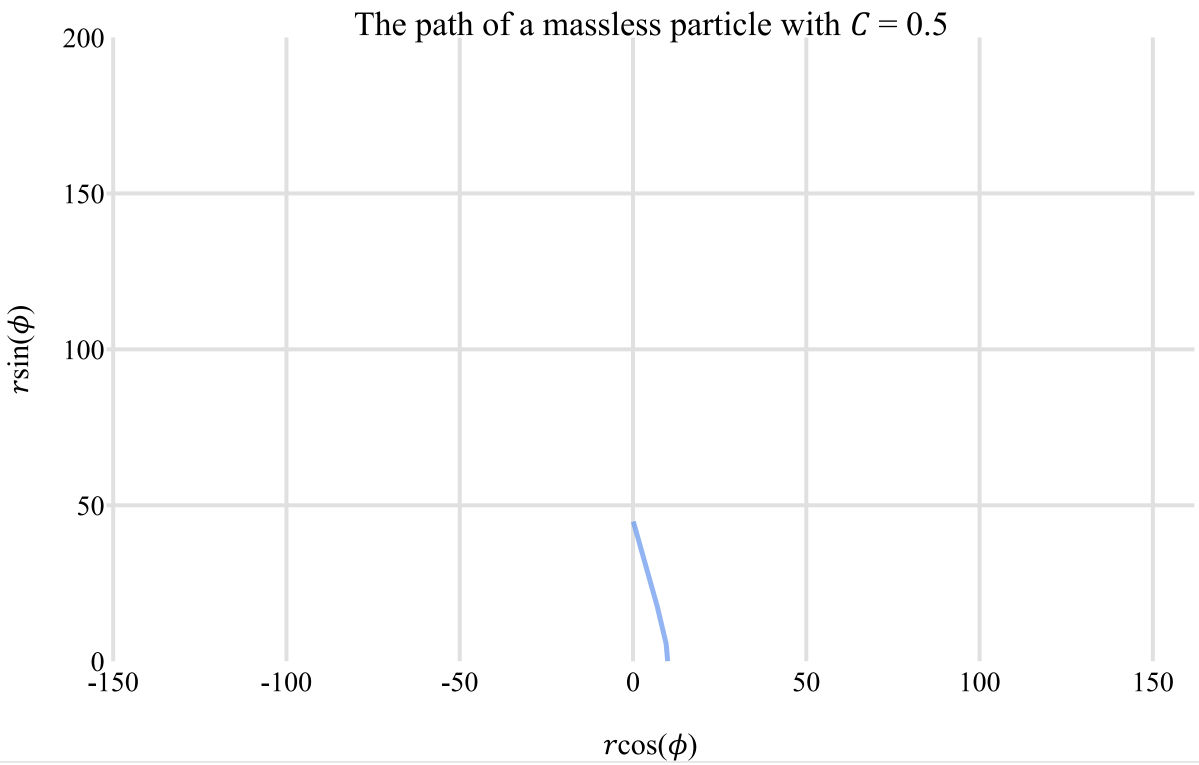
For a massless particle, $\mathbf{u} \cdot \mathbf{u}$ is 0. Additionally, the variable τ is the equations of motion no longer represents proper time but an affine parameter ($0 \leq \lambda \leq 10$).

i) $\quad C = 0.5$

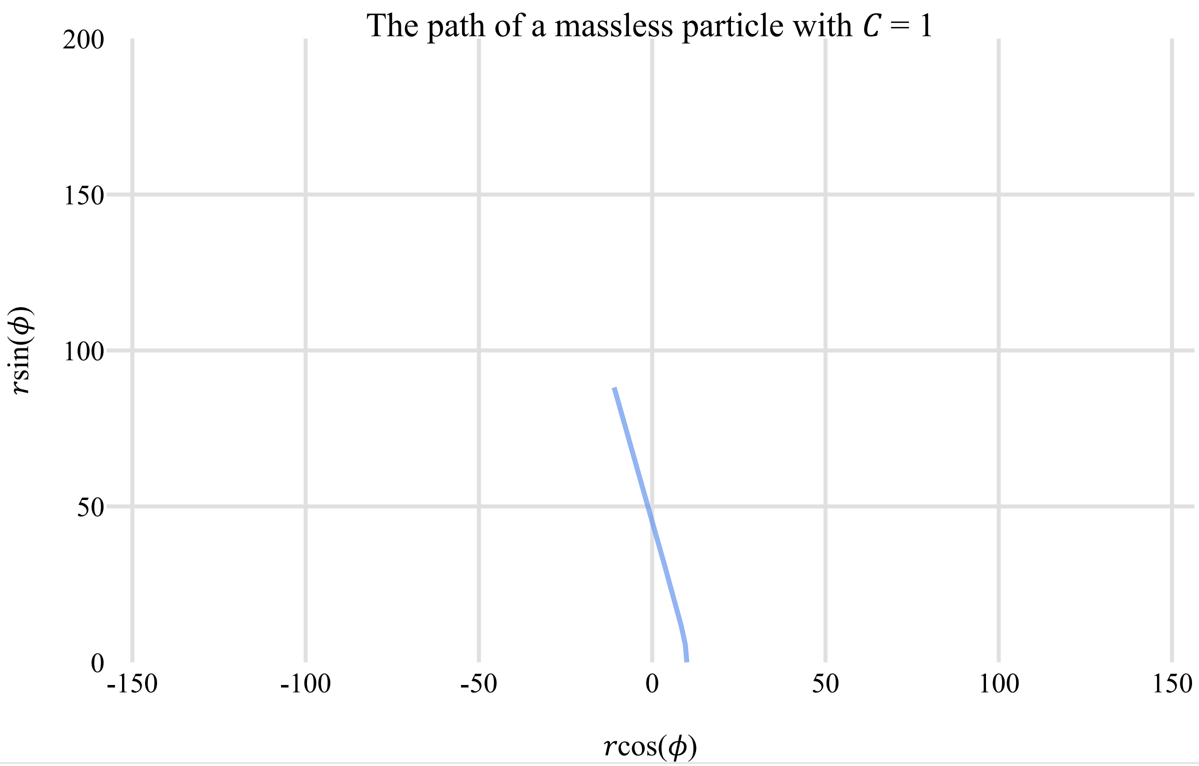
```
τ₃, t₃, r₃, ϕ₃, uᵗ₃, uʳ₃, uᵠ₃ = f(0.5, 0, 10);
```

```
τ₄, t₄, r₄, ϕ₄, uᵗ₄, uʳ₄, uᵠ₄ = f(1, 0, 10);
```

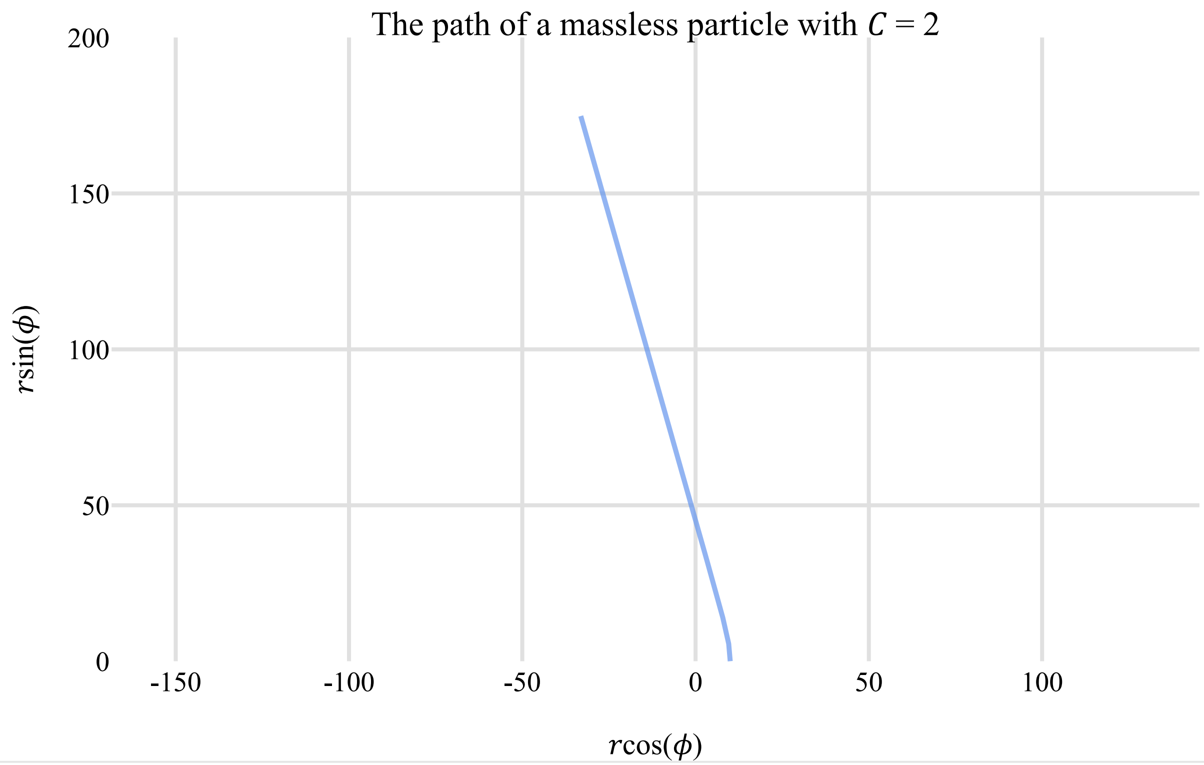
```
τ₅, t₅, r₅, ϕ₅, uᵗ₅, uʳ₅, uᵠ₅ = f(2, 0, 10);
```




```
plot(r₃.*cos.(ϕ₃), r₃.*sin.(ϕ₃), xguide="rcos(ϕ)", yguide="rsin(ϕ)", ylims=(0,
200),
         aspect_ratio=:equal, title="The path of a massless particle with C =
0.5")
```




```
plot(r₄.*cos.(ϕ₄), r₄.*sin.(ϕ₄), xguide="rcos(ϕ)", yguide="rsin(ϕ)", ylims=(0,
200),
         aspect_ratio=:equal, title="The path of a massless particle with C =
1")
```




```
plot(r₅.*cos.(ϕ₅), r₅.*sin.(ϕ₅), xguide="rcos(ϕ)", yguide="rsin(ϕ)", ylims=(0, 200),
          aspect_ratio=:equal, title="The path of a massless particle with C = 2")
```

In this question, the massless particle passes by a massive body and its trajectory (in Cartesian coordinates) is bent. This makes clear the existence of gravity as curvature in space-time, since it can influence even the paths of massless particles. Increasing the intial component of the four-velocity, C, causes the total spatial length of the massless particle's path to increase. However, the trajectory it ultimatley follows is the same; this is because the affine parameter is defined such that it can be substituted in place of (or, to generalise) τ in the equations of motion for a massive particle This gives a geodesic equation valid for a massless (0 four-velocity norm) particles. As such, the affine parameter can be linearly rescaled to give the same trajectory in spacetime for a massless photon; the affine parameter does not have an absolute correspondence to a proper time. Then C has no physical meaning as a velocity over a proper time for a massless partical, but does act to rescale the affine parameter. The result is that increasing C produces a longer path (in space, or, Cartesian coordinates) for the same interval of affine parameters, as demonstrated above.