Hopf Bifurcations in 2D (Assignment Sheet 7)

Introduction To Chaos Applied To Systems, Processes And Products (ETSIDI, UPM)

Alfonso Allen-Perkins, Juan Carlos Bueno and Eduardo Faleiro

2025-04-10

Contents

Introduction

In nonlinear dynamics, a **Hopf bifurcation** occurs when a fixed point of a system changes stability and a **limit cycle** (periodic orbit) appears or disappears. Depending on the system, the bifurcation can be:

- **Supercritical**: A supercritical Hopf bifurcation occurs when a stable spiral changes into an unstable spiral surrounded by a small, nearly elliptical limit cycle.
- **Subcritical**: A fixed point switches from unstable to stable, and at that same critical point, an unstable limit cycle emerges.

We'll illustrate both with examples.

Supercritical Hopf bifurcation

A simple example of a supercritical Hopf bifurcation can be written in polar coordinates as:

$$\dot{r} = \mu r - r^3$$
$$\dot{\theta} = \omega$$

The system depends on two parameters:

- μ controls the stability of the fixed point at the origin.
- ω sets the frequency of small-amplitude oscillations.

When $\mu < 0$, the origin ($r = 0$) is a **stable spiral**, with the direction of rotation determined by the sign of ω.

As μ increases and crosses zero, a **Hopf bifurcation** occurs. For $\mu > 0$, the origin becomes an **unstable spiral**, and a **stable circular limit cycle** emerges at radius $r = \sqrt{\mu}$.

Simulation with two trajectories and vector field

To simulate solutions and visualize the vector field, we rewrite the system in Cartesian coordinates. Starting from the polar-to-Cartesian transformation:

$$x = r\cos(\theta) \to \dot{x} = \dot{r}\cos(\theta) - r\dot{\theta}\sin(\theta)$$
$$y = r\sin(\theta) \to \dot{y} = \dot{r}\sin(\theta) + r\dot{\theta}\cos(\theta)$$

Substituting the expressions for $\dot{r} = \mu r - r^3$ and $\dot{\theta} = \omega$, we obtain:

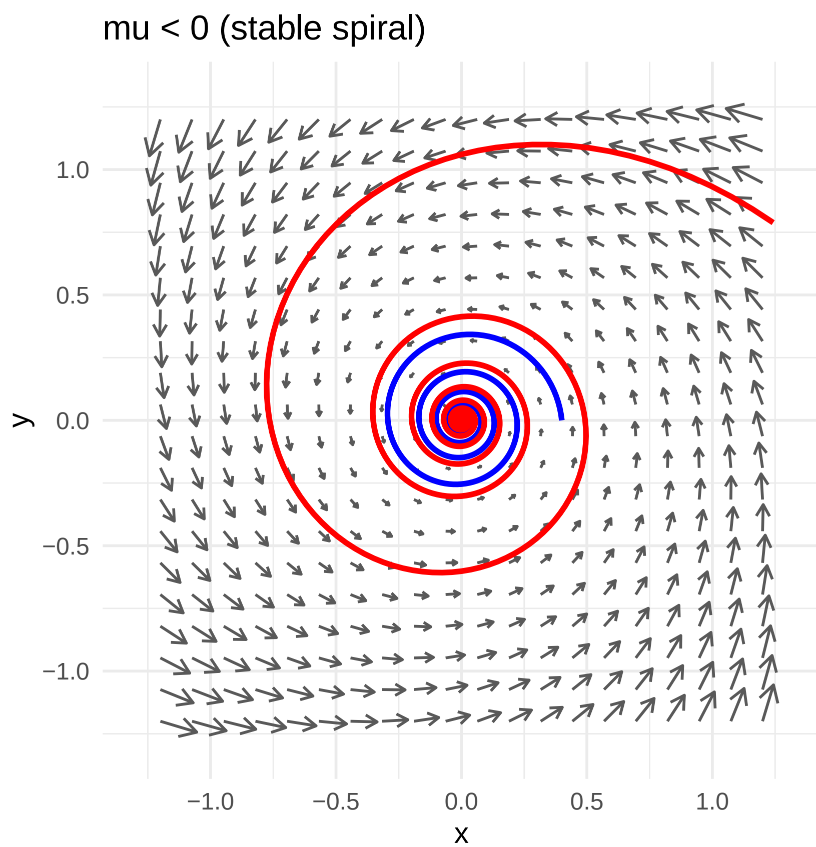
$$\dot{x} = (\mu r - r^3)\frac{x}{r} - r\omega\frac{y}{r}$$
$$\dot{y} = (\mu r - r^3)\frac{y}{r} + r\omega\frac{x}{r}$$

where $r = \sqrt{(x^2 + y^2)}$. This formulation enables us to simulate the dynamics and plot the vector field directly in Cartesian coordinates.

```
# Load the required libraries
library(deSolve)
library(ggplot2)
library(ggquiver)

# Define the polar system with the supercritical Hopf bifurcation
supercritical <- function(t, state, parameters) {
  r <- state[1]
  theta <- state[2]

  mu <- parameters["mu"]
  omega <- parameters["omega"]

  dr <- mu * r - r^3
  dtheta <- omega
  list(c(dr, dtheta))
}

# Set common initial conditions to study two trajectories (one for mu < 0 and
# the otherone for mu> 0)
state1 <- c(r = 0.4, theta = 0)
state2 <- c(r = 2, theta = 0)

# Set parameters and time for numerical integration, before the bifurcation (mu<0)
parameters_neg_mu <- c(mu = -0.5, omega = 2 * pi)
times <- seq(0, 50, by = 0.01)
```

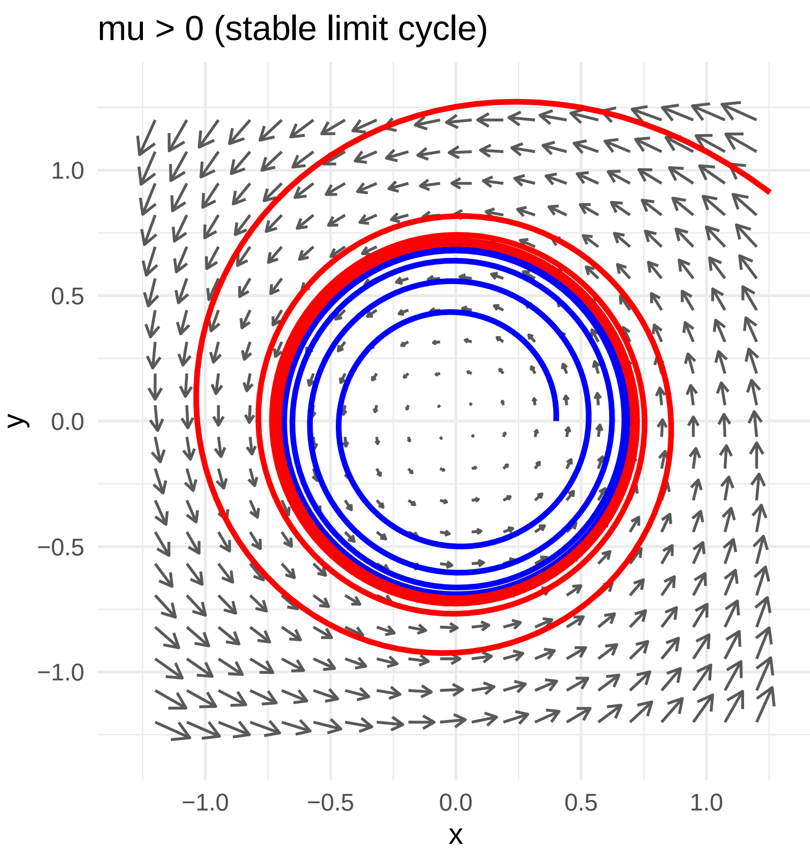
```r
# Set parameters and time for numerical integration, before the bifurcation (mu>0)
parameters_pos_mu <- c(mu = 0.5, omega = 2 * pi)
times <- seq(0, 50, by = 0.01)

# Function to simulate and convert polar to cartesian
simulate_hopf <- function(state, parameters, times) {

  # Trajectory for state
  result <- ode(y = state, times = times, func = supercritical, parms = parameters)
  result_df <- as.data.frame(result)
  result_df$x <- result_df$r * cos(result_df$theta)
  result_df$y <- result_df$r * sin(result_df$theta)

  return(result_df)

}

# Vector field in Cartesian coordinates
make_vecfield <- function(mu, omega = 2 * pi) {
  grid <- expand.grid(x = seq(-1.2, 1.2, length.out = 20),
                      y = seq(-1.2, 1.2, length.out = 20))
  grid$r <- sqrt(grid$x * grid$x + grid$y * grid$y)
  grid$costheta <- grid$x /grid$r
  grid$sintheta <- grid$y /grid$r
  grid$dr <- mu * grid$r - grid$r^3
  grid$dx <- grid$dr * grid$costheta - grid$r * omega * grid$sintheta
  grid$dy <- grid$dr * grid$sintheta + grid$r * omega * grid$costheta
  grid
}

# Simulations (mu < 0)
sol_1_neg_mu <- simulate_hopf(state1, parameters_neg_mu, times)
sol_2_neg_mu <- simulate_hopf(state2, parameters_neg_mu, times)

# Simulations (mu > 0)
sol_1_pos_mu <- simulate_hopf(state1, parameters_pos_mu, times)
sol_2_pos_mu <- simulate_hopf(state2, parameters_pos_mu, times)

# Vector fields for mu < 0 and mu > 0 respectively
vec_neg <- make_vecfield(mu = -0.5)
vec_pos <- make_vecfield(mu = 0.5)


# Plot for mu < 0
ggplot() +
  geom_quiver(data = vec_neg, aes(x, y, u = dx, v = dy), vecsize = 1.2, color = "gray35") +
  geom_path(data = sol_1_neg_mu, aes(x = x, y = y), color = "blue", linewidth = 1) +
  geom_path(data = sol_2_neg_mu, aes(x = x, y = y), color = "red", linewidth = 1) +
  ggtitle("mu < 0 (stable spiral)") +
  xlim(-1.3,1.3)+
  ylim(-1.3,1.3)+
  coord_equal() +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_path()`).
```




```r
# Plot for mu > 0
ggplot() +
  geom_quiver(data = vec_pos, aes(x, y, u = dx, v = dy), vecsize = 1.2, color = "gray35") +
  geom_path(data = sol_1_pos_mu, aes(x = x, y = y), color = "blue", linewidth = 1) +
  geom_path(data = sol_2_pos_mu, aes(x = x, y = y), color = "red", linewidth = 1) +
  xlim(-1.3,1.3)+
  ylim(-1.3,1.3)+
  ggtitle("mu > 0 (stable limit cycle)") +
  coord_equal() +
  theme_minimal()
```

```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_path()`).
```



Subcritical Hopf bifurcation

A simple example of a subcritical Hopf bifurcation can be written in polar coordinates as:

$$\dot{r} = \mu r + r^3 - r^5$$
$$\dot{\theta} = \omega$$

The system depends on two parameters:

- μ controls the stability of the fixed point at the origin.
- ω sets the frequency of small-amplitude oscillations.

When $-0.25 < \mu < 0$, the system exhibits **bistability**, with two attractors: a **stable limit cycle** and a **stable fixed point** at $r = 0$. Separating them is an **unstable limit cycle** (see an example for in Appendix 1 for $\mu = -0.1$). Trajectories that start between the unstable limit cycle and one of the attractors spiral stably toward the corresponding attractor. The direction of rotation is determined by the sign of ω.

As μ increases and crosses zero, a **subcritical Hopf bifurcation** occurs. For $\mu > 0$, the unstable limit cycle collapses into the origin, destabilizing it. The only surviving attractor is the **stable limit cycle**, which now dominates the dynamics.

Simulation with several trajectories and vector field

To simulate solutions and visualize the vector field, we rewrite the system in Cartesian coordinates. Starting from the polar-to-Cartesian transformation:

$$x = r\cos(\theta) \to \dot{x} = \dot{r}\cos(\theta) - r\dot{\theta}\sin(\theta)$$
$$y = r\sin(\theta) \to \dot{y} = \dot{r}\sin(\theta) + r\dot{\theta}\cos(\theta)$$

Substituting the expressions for $\dot{r} = \mu r - r^3$ and $\dot{\theta} = \omega$, we obtain:

$$\dot{x} = (\mu r + r^3 - r^5)\frac{x}{r} - r\omega\frac{y}{r}$$
$$\dot{y} = (\mu r + r^3 - r^5)\frac{y}{r} + r\omega\frac{x}{r}$$

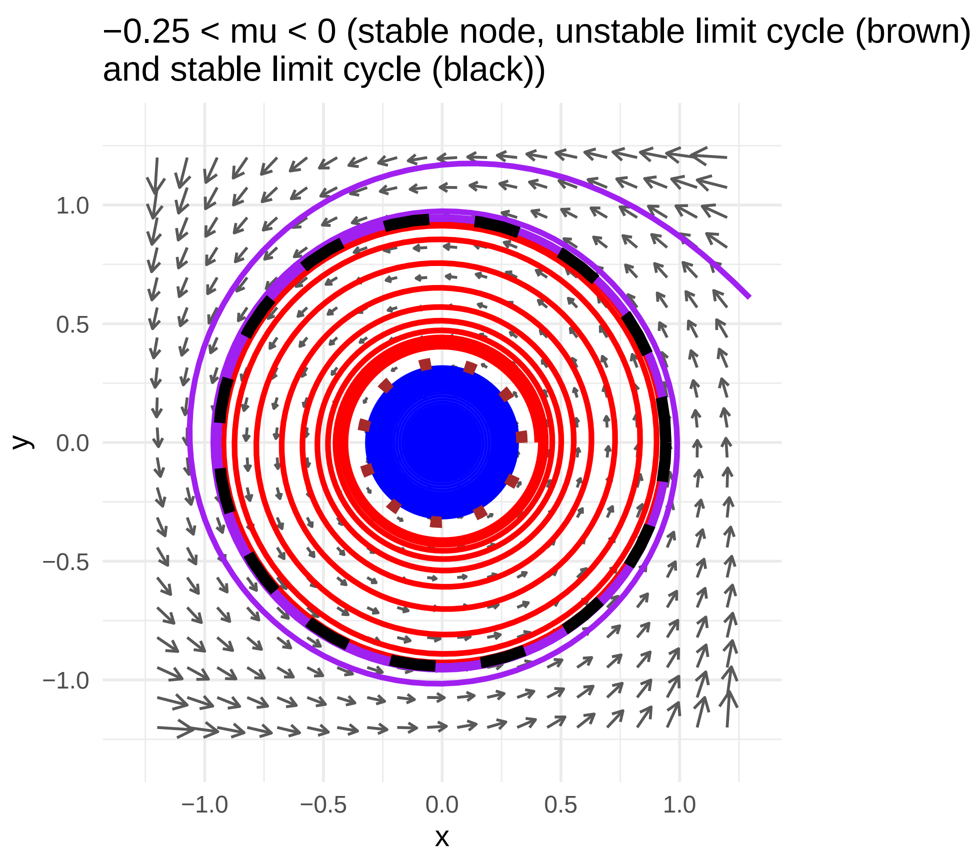
where $r = \sqrt{(x^2 + y^2)}$. This formulation enables us to simulate the dynamics and plot the vector field directly in Cartesian coordinates.

```r
# Load the required libraries
library(deSolve)
library(ggplot2)
library(ggquiver)

# Define the polar system with the subcritical Hopf bifurcation
subcritical <- function(t, state, parameters) {
  r <- state[1]
  theta <- state[2]

  mu <- parameters["mu"]
  omega <- parameters["omega"]

  dr <- mu * r + r^3 - r^5
  dtheta <- omega

  list(c(dr, dtheta))
}


# Set common initial conditions to study two trajectories (one for mu < 0 and
# the otherone for mu> 0)
state1 <- c(r = 0.32, theta = 0)
state2 <- c(r = .4, theta = 0)
state3 <- c(r = 2, theta = 0)

# Set parameters and time for numerical integration, before the bifurcation (mu<0)
parameters_neg_mu <- c(mu = -0.1, omega = 2 * pi)
times <- seq(0, 100, by = 0.01)

# Set parameters and time for numerical integration, before the bifurcation (mu>0)
parameters_pos_mu <- c(mu = 0.1, omega = 2 * pi)
times <- seq(0, 20, by = 0.01)
times_large <- seq(0, 200, by = 0.01)

# Function to simulate and convert polar to cartesian
simulate_hopf_sub <- function(state, parameters, times) {

  # Trajectory for state
  result <- ode(y = state, times = times, func = subcritical, parms = parameters)
  result_df <- as.data.frame(result)
```

```r
  result_df$x <- result_df$r * cos(result_df$theta)
  result_df$y <- result_df$r * sin(result_df$theta)

  return(result_df)

}

# Vector field in Cartesian coordinates
make_vecfield_sub <- function(mu, omega = 2 * pi) {
  grid <- expand.grid(x = seq(-1.2, 1.2, length.out = 20),
                      y = seq(-1.2, 1.2, length.out = 20))
  grid$r <- sqrt(grid$x^2 + grid$y^2)
  grid$costheta <- grid$x /grid$r
  grid$sintheta <- grid$y /grid$r
  grid$dr <- mu * grid$r + grid$r^3 - grid$r^5
  grid$dx <- grid$dr * grid$costheta - grid$r * omega * grid$sintheta
  grid$dy <- grid$dr * grid$sintheta + grid$r * omega * grid$costheta
  grid
}

# Simulations (mu < 0)
sol_1_neg_mu_sub <- simulate_hopf_sub(state1, parameters_neg_mu, times_large)
sol_2_neg_mu_sub <- simulate_hopf_sub(state2, parameters_neg_mu, times)
sol_3_neg_mu_sub <- simulate_hopf_sub(state3, parameters_neg_mu, times)

# Simulations (mu > 0)
sol_1_pos_mu_sub <- simulate_hopf_sub(state1, parameters_pos_mu, times)
sol_2_pos_mu_sub <- simulate_hopf_sub(state2, parameters_pos_mu, times)
sol_3_pos_mu_sub <- simulate_hopf_sub(state3, parameters_pos_mu, times)

# Vector fields for mu < 0 and mu > 0 respectively
vec_neg_sub <- make_vecfield_sub(mu = -.1)
vec_pos_sub <- make_vecfield_sub(mu = .1)

# limit cycles ( mu = -0.1, appendix 1)
seq_theta <- c(seq(0, 2 * pi, by = 0.1), 0)
limit_cycle_stable_neg <- data.frame(x = 0.942*cos(seq_theta),
                                     y = 0.942*sin(seq_theta)
)

limit_cycle_unstable_neg <- data.frame(x = 0.336*cos(seq_theta),
                                       y = 0.336*sin(seq_theta)
)

# limit cycles ( mu = 0.1, not included)
limit_cycle_stable_pos <- data.frame(x = 1.0448*cos(seq_theta),
                                     y = 1.0448*sin(seq_theta)
)



# Plot for mu < 0
ggplot() +
```

```
  geom_quiver(data = vec_neg_sub, aes(x, y, u = dx, v = dy),
              vecsize = 1.2, color = "gray35") +
  geom_path(data = sol_1_neg_mu_sub, aes(x = x, y = y),
            color = "blue", linewidth = .5) +
  geom_path(data = sol_2_neg_mu_sub, aes(x = x, y = y),
            color = "red", linewidth = 1) +
  geom_path(data = sol_3_neg_mu_sub, aes(x = x, y = y),
            color = "purple", linewidth = 1) +
  geom_path(data = limit_cycle_stable_neg, aes(x = x, y = y),
            color = "black", linetype = "dashed", linewidth = 2) +
  geom_path(data = limit_cycle_unstable_neg, aes(x = x, y = y),
            color = "brown", linetype = "dotted", linewidth = 2) +
  ggtitle("-0.25 < mu < 0 (stable node, unstable limit cycle (brown)\nand stable limit cycle (black))")
  xlim(-1.3,1.3)+
  ylim(-1.3,1.3)+
  coord_equal() +
  theme_minimal()
```

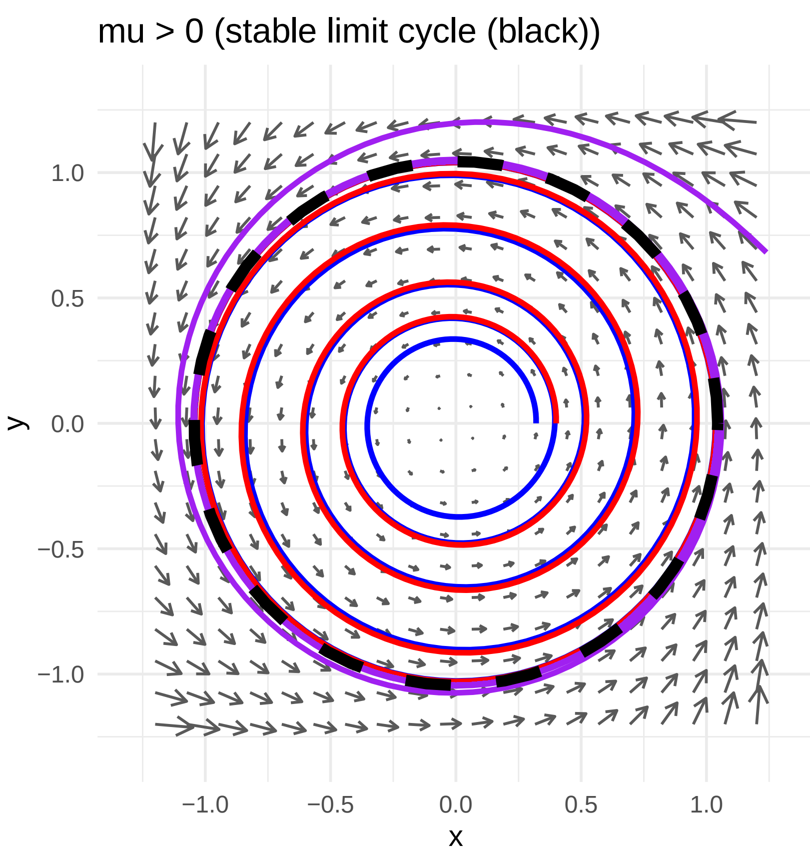
```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_path()`).
```




```
# Plot for mu > 0
ggplot() +
  geom_quiver(data = vec_pos_sub, aes(x, y, u = dx, v = dy),
```

```r
               vecsize = 1.2, color = "gray35") +
  geom_path(data = sol_1_pos_mu_sub, aes(x = x, y = y),
            color = "blue", linewidth = 1) +
  geom_path(data = sol_2_pos_mu_sub, aes(x = x, y = y),
            color = "red", linewidth = 1) +
  geom_path(data = sol_3_pos_mu_sub, aes(x = x, y = y),
            color = "purple", linewidth = 1) +
  geom_path(data = limit_cycle_stable_pos, aes(x = x, y = y),
            color = "black", linetype = "dashed", linewidth = 2) +
  xlim(-1.3,1.3)+
  ylim(-1.3,1.3)+
  ggtitle("mu > 0 (stable limit cycle (black))") +
  coord_equal() +
  theme_minimal()
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_path()`).
```



Appendix 1: Finding the limit cycles for the subcritical Hopf bifurcation when $\mu = -0.1$

Consider the radial equation $\dot{r} = \mu r + r^3 - r^5$, with $\mu = -0.1$. We look for limit cycles, which correspond to positive roots of $\mu r + r^3 - r^5 = 0 \rightarrow \mu + r^2 - r^4 = 0$. Aside from the trivial solution $r = 0$, any positive root of $\mu + r^2 - r^4 = 0 \rightarrow r^4 - r^2 - \mu = 0$ gives the radius of a limit cycle. Below, we solve this numerically and determine which cycles are stable vs. unstable by checking the sign of $f'(r) = \frac{d}{dr}\left(\mu r + r^3 - r^5\right) = \mu + 3r^2 - 5r^4$. If $f'(r_0) < 0$, the cycle at $r = r_0$ is stable; if $f'(r_0) > 0$, it is unstable.

```r
# Parameter
mu <- -0.1

# We want to solve mu*r + r^3 - r^5 = 0
# polyroot() expects polynomial coefficients in ascending order:
#    p(x) = a0 + a1*x + a2*x^2 + a3*x^3 + a4*x^4 + a5*x^5
# Here, that corresponds to 0 + (mu)*x + 0*x^2 + 1*x^3 + 0*x^4 + (-1)*x^5
roots_r <- polyroot(c(0, mu, 0, 1, 0, -1))

# Define the derivative f'(r) = mu + 3*r^2 - 5*r^4
fprime <- function(r) {
  mu + 3*r^2 - 5*r^4
}

# Evaluate f'(r) for each root
fprime_roots_r <- fprime(roots_r)

# Determine stability: if f'(r) < 0 => "Stable", else "Unstable"
# Use ifelse() to handle multiple roots at once.
stability_roots_r <- ifelse(Re(fprime_roots_r) < 0, "Stable", "Unstable")

# Print out the stability classification
data.frame(root = roots_r,
           real_part = Re(roots_r),
           imaginary_part = round(Im(roots_r), 10),
           stability = stability_roots_r)
```

```
##                        root  real_part imaginary_part stability
## 1  0.0000000+0.000000e+00i  0.0000000              0    Stable
## 2  0.3357107+0.000000e+00i  0.3357107              0  Unstable
## 3 -0.3357107-4.038968e-28i -0.3357107              0  Unstable
## 4 -0.9419651-5.169879e-26i -0.9419651              0    Stable
## 5  0.9419651+5.210269e-26i  0.9419651              0    Stable
```

Interpretation

We obtain two positive solutions $r_1 \approx 0.336$ and $r_2 \approx 0.942$.

Checking $f'(r)$ shows that:

- r_1 is **unstable** (because $f'(r_1) > 0$)
- r_2 is **stable** (because $f'(r_2) < 0$)

Hence, at $\mu = -0.1$, there is one unstable limit cycle at the smaller radius and one stable limit cycle at the larger radius. The trivial solution $r = 0$ also exists, but its stability typically depends on μ and the interplay with the other cycles (in a subcritical Hopf scenario, the origin can be stable for certain ranges of μ, leading to bistability).
