Ellipsis Projection

Daniel Posmik

2025-03-13

Simulation of Points on an Elliptic Curve

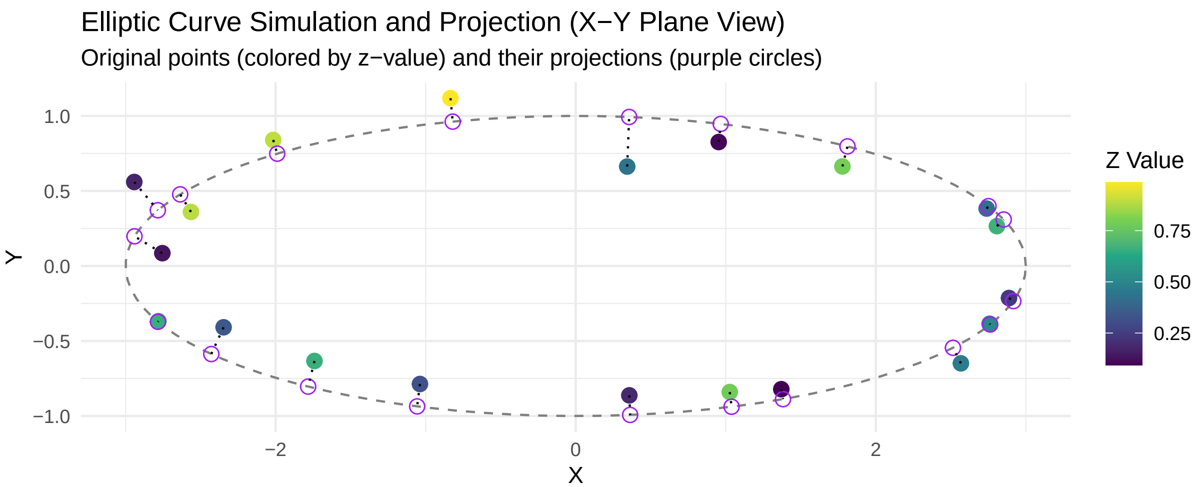
First, we set up our parameters and generate points on an elliptic curve.

Projection onto the Ellipse

Here we project the noisy points back onto the ellipse, considering all three dimensions.

```
          x          y         z x_projected y_projected z_projected
1 2.8879049 -0.2135647 0.2436195   2.9165221  -0.2342598           0
2 2.8071341  0.2654220 0.6680556   2.8524703   0.3097334           0
3 2.7387926  0.3825844 0.4176468   2.7511968   0.3987363           0
4 1.7774574  0.6632387 0.7881958   1.8113056   0.7971597           0
5 0.9529085  0.8260487 0.1028646   0.9665521   0.9466771           0
6 0.3430130  0.6626613 0.4348927   0.3561976   0.9929263           0
  projection_theta projection_phi projection_distance
1        6.0467282     0.08393118           0.2461660
2        0.3149127     0.23263845           0.6710567
3        0.4101385     0.14989382           0.4181430
4        0.9225762     0.39376209           0.8002082
5        1.2427607     0.08138677           0.1591179
6        1.4517830     0.52769806           0.5462421
```

2D Visualization



Summary of Projection Metrics

```
  Point Original_X Original_Y Original_Z Projected_X Projected_Y Projected_Z
1     1  2.8879049 -0.2135647  0.2436195   2.9165221  -0.2342598           0
2     2  2.8071341  0.2654220  0.6680556   2.8524703   0.3097334           0
3     3  2.7387926  0.3825844  0.4176468   2.7511968   0.3987363           0
4     4  1.7774574  0.6632387  0.7881958   1.8113056   0.7971597           0
5     5  0.9529085  0.8260487  0.1028646   0.9665521   0.9466771           0
6     6  0.3430130  0.6626613  0.4348927   0.3561976   0.9929263           0
  Theta_Angle  Phi_Angle Projection_Distance
1   6.0467282 0.08393118           0.2461660
2   0.3149127 0.23263845           0.6710567
3   0.4101385 0.14989382           0.4181430
4   0.9225762 0.39376209           0.8002082
5   1.2427607 0.08138677           0.1591179
6   1.4517830 0.52769806           0.5462421
```

Code Appendix

```
# Set up knit environment
knitr::opts_chunk$set(echo = F)
knitr::opts_chunk$set(error = F)
knitr::opts_chunk$set(warning = F)
```

```r
knitr::opts_chunk$set(message = F)
# Set random seed for reproducibility
set.seed(123)

# Parameters for the elliptic curve
a <- 3  # semi-major axis
b <- 1  # semi-minor axis

# Generate 5 points on an elliptic curve
n <- 20
theta <- seq(0, 2*pi, length.out = n+1)[1:n]  # equally spaced angles

# Generate points on the perfect ellipse
x_ellipse <- a * cos(theta)
y_ellipse <- b * sin(theta)

# Add Gaussian noise N(0,1) to create observed points
noise_sd <- 0.2
x_observed <- x_ellipse + rnorm(n, 0, noise_sd)
y_observed <- y_ellipse + rnorm(n, 0, noise_sd)

# Generate uniform(0,1) values for the z coordinate
z_values <- runif(n)

# Create the data frame
data <- data.frame(
  x = x_observed,
  y = y_observed,
  z = z_values
)
# Project a point onto the ellipse and return coordinates and angles
project_to_ellipse <- function(x, y, z, a, b) {
  # Starting guess for theta (x-y plane angle)
  theta_guess <- atan2(y/b, x/a)

  # Function to minimize: squared distance from (x,y,z) to a point on the ellipse
  distance_function <- function(theta) {
    ellipse_x <- a * cos(theta)
    ellipse_y <- b * sin(theta)
    ellipse_z <- 0 # Since our ellipse only lives in x,y space
    return(sqrt((x - ellipse_x)^2 + (y - ellipse_y)^2 + (z - ellipse_z)^2))
  }
```

```r
  # Optimize to find the best theta
  result <- optimize(distance_function, c(0, 2*pi))
  theta_optimal <- result$minimum

  # Calculate projected coordinates
  x_proj <- a * cos(theta_optimal)
  y_proj <- b * sin(theta_optimal)
  z_proj <- 0

  # Calculate distance between original and projected point
  dist <- sqrt((x - x_proj)^2 + (y - y_proj)^2 + (z - z_proj)^2)

  # Calculate the second angle (phi) - elevation from x-y plane
  # This is the angle between the x-y plane and the line to the original point
  phi <- atan2(z, sqrt(x^2 + y^2))

  # Return the projected point coordinates, angles, and distance
  return(list(
    x_proj = x_proj,
    y_proj = y_proj,
    z_proj = z_proj,
    theta = theta_optimal,  # Azimuthal angle in x-y plane
    phi = phi,              # Elevation angle from x-y plane
    distance = dist
  ))
}

# Apply the projection to each point
projection_results <- lapply(1:n, function(i) {
  project_to_ellipse(data$x[i], data$y[i], data$z[i], a, b)
})

# Extract results into data frame
data$x_projected <- sapply(projection_results, function(res) res$x_proj)
data$y_projected <- sapply(projection_results, function(res) res$y_proj)
data$z_projected <- sapply(projection_results, function(res) res$z_proj)
data$projection_theta <- sapply(projection_results, function(res) res$theta)
data$projection_phi <- sapply(projection_results, function(res) res$phi)
data$projection_distance <- sapply(projection_results, function(res) res$distance)

# Print the data frame with projection information
head(data)
```

```r
library(ggplot2)

# Create a dense set of points for drawing the perfect ellipse
theta_dense <- seq(0, 2*pi, length.out = 100)
ellipse_points <- data.frame(
  x = a * cos(theta_dense),
  y = b * sin(theta_dense)
)

# Create the plot
p <- ggplot() +
  # Draw the perfect ellipse
  geom_path(data = ellipse_points, aes(x = x, y = y), color = "gray50", linetype = "dashed")
  # Draw the observed points with color based on z-value
  geom_point(data = data, aes(x = x, y = y, color = z), size = 3) +
  # Add a color scale for the z-values
  scale_color_viridis_c(name = "Z Value") +
  # Draw the projected points
  geom_point(data = data, aes(x = x_projected, y = y_projected), color = "purple", shape = 1
  # Draw lines connecting observed points to their projections
  geom_segment(data = data,
               aes(x = x, y = y, xend = x_projected, yend = y_projected),
               linetype = "dotted") +
  # Add labels and theme
  labs(title = "Elliptic Curve Simulation and Projection (X-Y Plane View)",
       subtitle = "Original points (colored by z-value) and their projections (purple circle
       x = "X", y = "Y") +
  theme_minimal() +
  coord_equal()  # Equal aspect ratio for proper ellipse visualization

# Display the plot
print(p)
# Create summary table
summary_table <- data.frame(
  Point = 1:n,
  Original_X = data$x,
  Original_Y = data$y,
  Original_Z = data$z,
  Projected_X = data$x_projected,
  Projected_Y = data$y_projected,
  Projected_Z = data$z_projected,
  Theta_Angle = data$projection_theta,  # Azimuthal angle in x-y plane
```

```
  Phi_Angle = data$projection_phi,      # Elevation angle from x-y plane
  Projection_Distance = data$projection_distance
)

# Print the summary table
head(summary_table)
```