

Assignment_1

22CE01002_Shivam_Kumar_Rai

Github: <https://github.com/RaiShivam-ui>

Question 1

```
#A1_Q1_(a)
#Hill Scalar Field
using Plots
using CalculusWithJulia

# Scalar field
h(x, y) = 200 - x^2 - 2*y^2

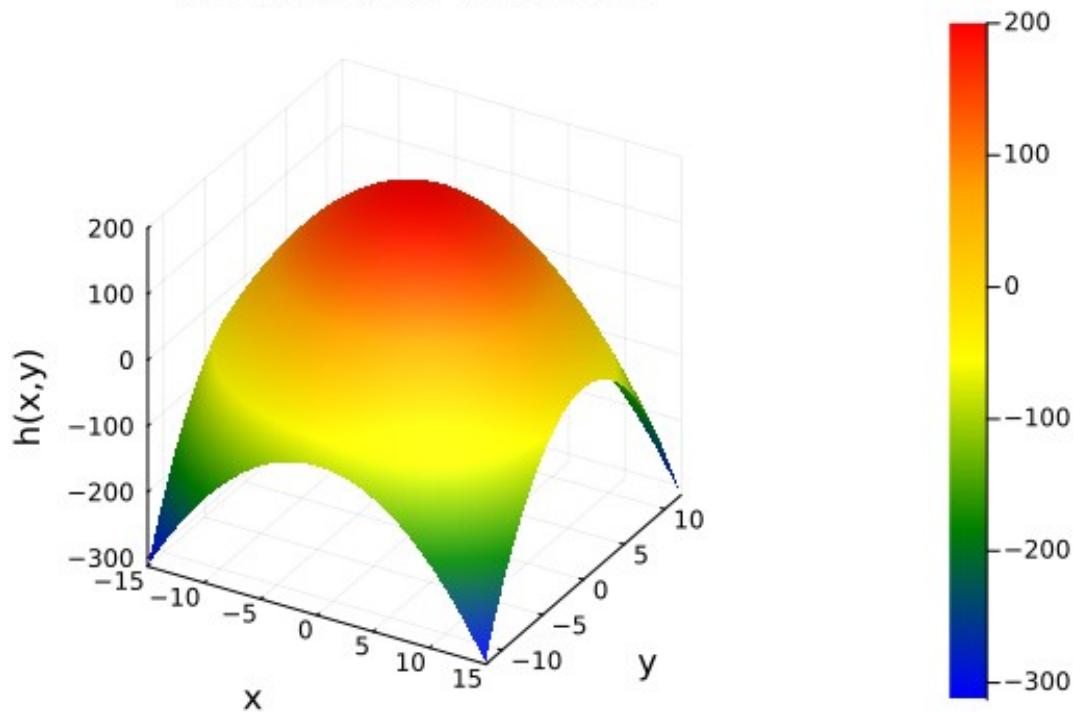
# Grid definition
x = -15:0.1:15
y = -12:0.1:12

# Color gradient
color_grad = cgrad([ :blue, :green, :yellow, :orange, :red])
```

```
# 3D Surface Plot

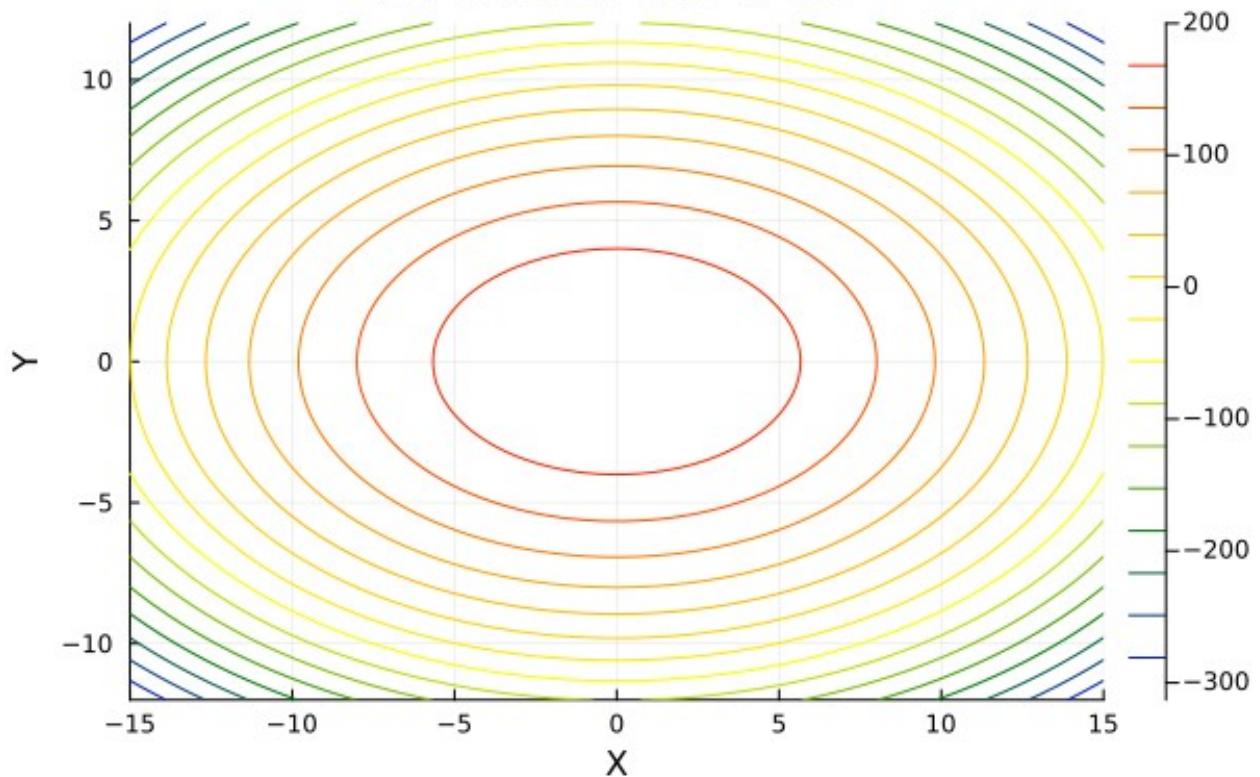
surface(x, y, h,
        c = color_grad,
        xlabel = "x", ylabel = "y", zlabel = "h(x,y)",
        title = "3D Surface Plot of Hill"
)
```

3D Surface Plot of Hill



```
# 2D Contour Plot
contour(x, y, h,
        c = color_grad,
        xlabel = "X", ylabel = "Y",
        title = "2D Contour Plot of Hill"
    )
```

2D Contour Plot of Hill



```

# Test: Evaluating h at a point
println("Height at (1,1) = ", h(1,1))

# Gradient at a point (1,1)
grad_h = gradient(u -> h(u[1], u[2]), [1,1])
println("Gradient at (1,1) = ", grad_h)

Height at (1,1) = 197
Gradient at (1,1) = [-2, -4]

#Q1_b&c
# Question 1: Gradient Vector Field

h(x, y) = 200 - x^2 - 2*y^2

x2 = -10:0.8:10
y2 = -10:0.8:10

U = [gradient(u -> h(u[1], u[2]), [x, y])[1] for x in x2, y in y2]
V = [gradient(u -> h(u[1], u[2]), [x, y])[2] for x in x2, y in y2]

X = [x for x in x2, y in y2]
Y = [y for x in x2, y in y2]

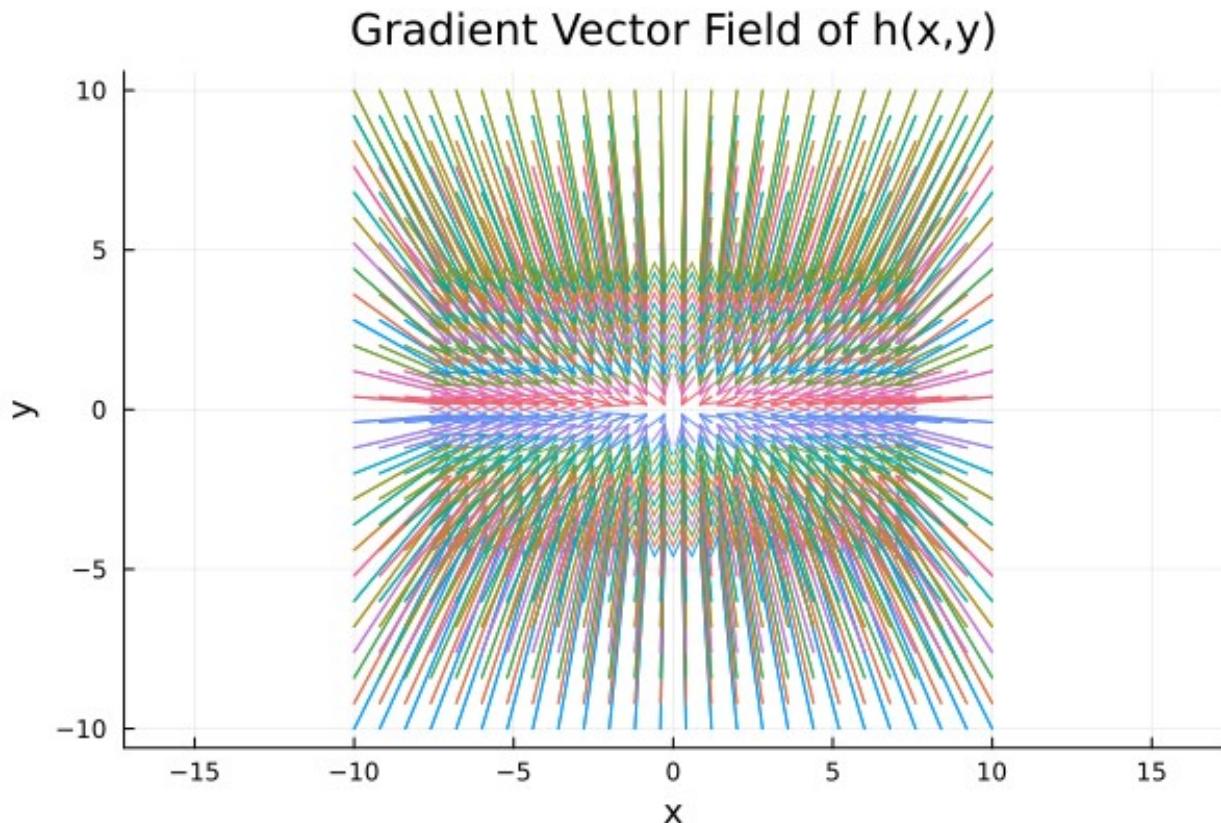
```

```

scale = 0.15
U .*= scale
V .*= scale

quiver(
    X, Y, quiver=(U, V),
    linealpha=0.9,
    arrowsize=0.2,
    aspect_ratio=:equal,
    title="Gradient Vector Field of h(x,y)",
    xlabel="x", ylabel="y"
)

```



Question 2

```

# Question 2(a): Cyclone Velocity Vector Field
function velocity_vector(x, y)
    return [x, -y^2]
end

x3 = -10:0.8:10
y3 = -10:0.8:10

U = [velocity_vector(x, y)[1] for x in x3, y in y3]
V = [velocity_vector(x, y)[2] for x in x3, y in y3]

```

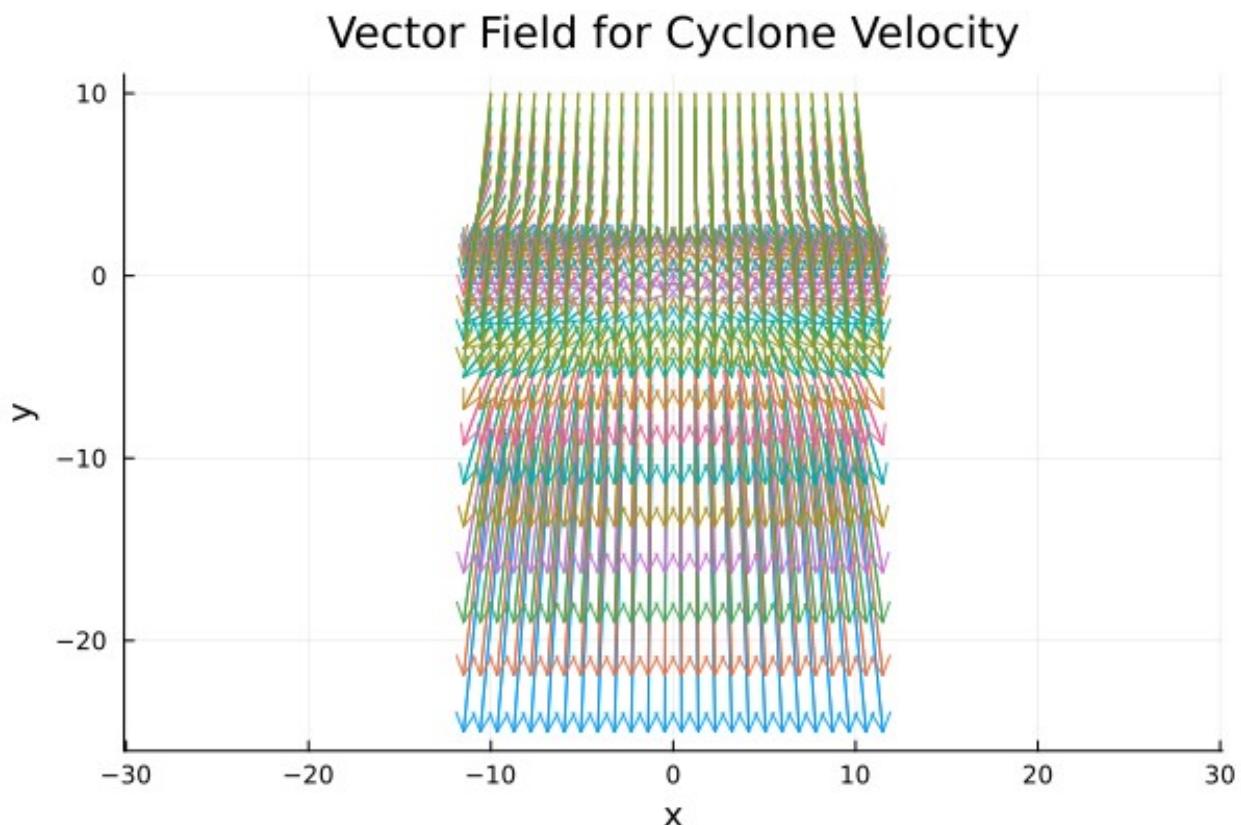
```

X = [x for x in x3, y in y3]
Y = [y for x in x3, y in y3]

scale = 0.15
U .*= scale
V .*= scale

quiver(
    X, Y, quiver=(U, V),
    title="Vector Field for Cyclone Velocity",
    xlabel="x", ylabel="y",
    linealpha=0.9, arrowsize=0.2,
    aspect_ratio=:equal
)

```



```

using Plots
using CalculusWithJulia
function velocity_vector(x, y)
    return [x, -y^2]
end

div_manual(x, y) = (gradient(u -> velocity_vector(u[1], u[2]))[1], [x,
y])[1]) +

```

```

        (gradient(u -> velocity_vector(u[1], u[2]))[2], [x,
y])[2])

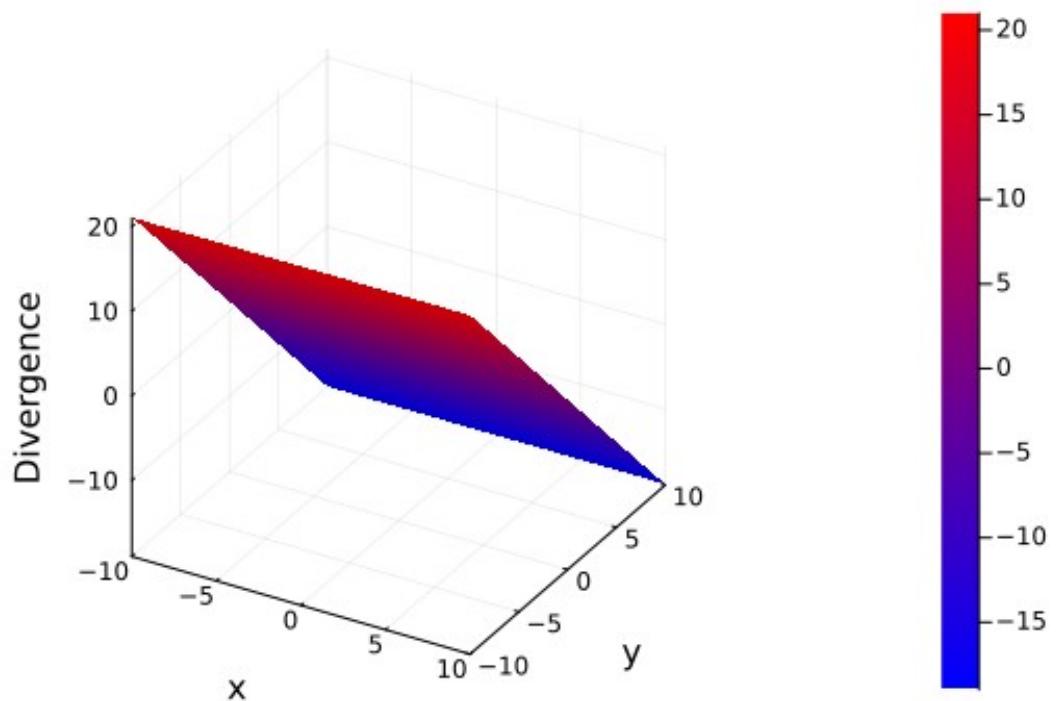
x4 = -10:0.5:10
y4 = -10:0.5:10

color_grad = cgrad([:blue, :red])

surface(
    x4, y4, div_manual,
    c = color_grad,
    title = "Divergence Plot (Manual Calculation)",
    xlabel = "x", ylabel = "y", zlabel = "Divergence"
)

```

Divergence Plot (Manual Calculation)



```

# Question 2(b): Divergence of Cyclone Velocity Field (Automatic)
function velocity_vector(x, y)
    return [x, -y^2]
end

# Automatic divergence calculation
div_auto(x, y) = divergence(u -> velocity_vector(u[1], u[2])), [x, y])

x4 = -10:0.8:10

```

```

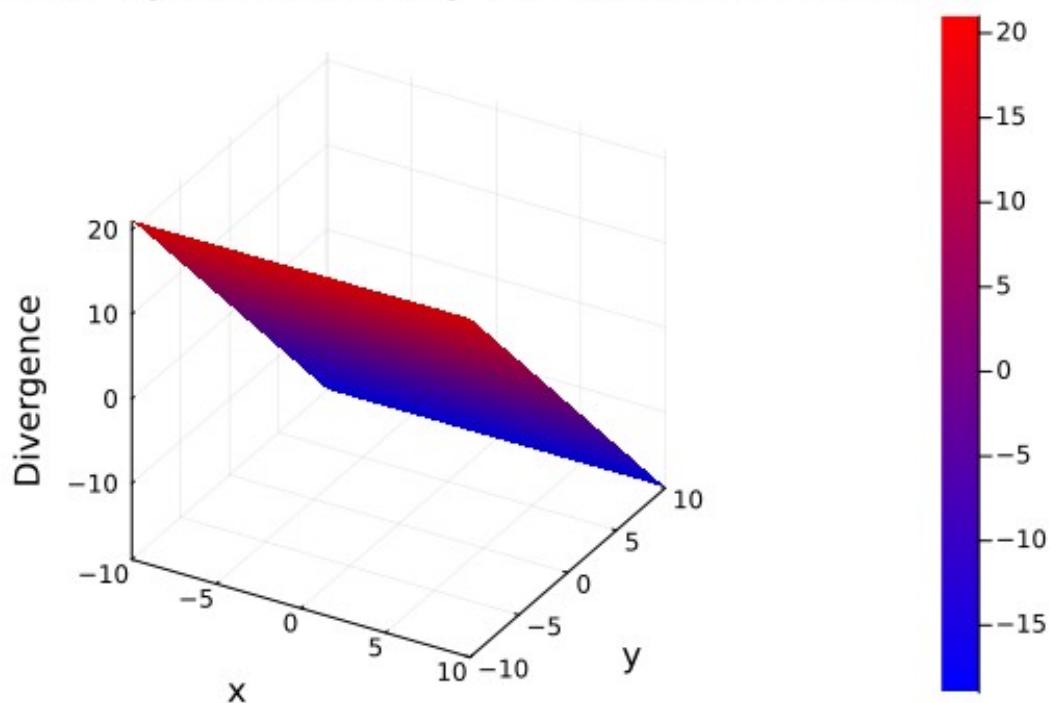
y4 = -10:0.8:10

color_grad = cgrad(:blue, :red)

# Surface plot of automatic divergence
surface(
    x4, y4, div_auto,
    c = color_grad,
    title = "Divergence of Cyclone Velocity (Automatic Calculation)",
    xlabel = "x", ylabel = "y", zlabel = "Divergence"
)

```

Divergence of Cyclone Velocity (Automatic Calculation)



```
# Question 2(c): Curl of Cyclone Velocity Field
```

```

function velocity_vector(x, y)
    return [x, -y^2]
end

curl_auto(x, y) = curl(u -> velocity_vector(u[1], u[2])), [x, y])

x3 = -10:0.8:10
y3 = -10:0.8:10

```

```

U = [velocity_vector(x, y)[1] for x in x3, y in y3]
V = [velocity_vector(x, y)[2] for x in x3, y in y3]

U .*= 0.1
V .*= 0.1

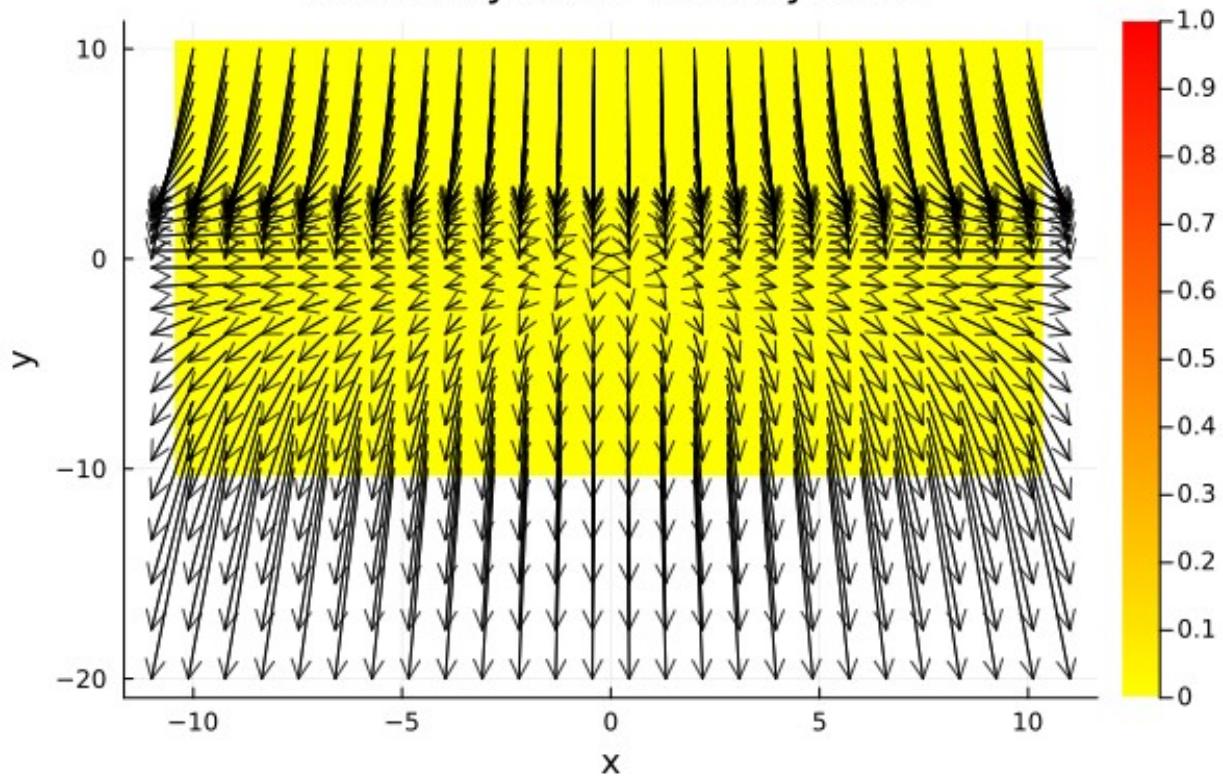
X = [x for x in x3, y in y3]
Y = [y for x in x3, y in y3]

color_grad = cgrad([:yellow, :red])
# Heatmap of Curl
heatmap(
    x3, y3, curl_auto,
    c = color_grad,
    colour=:balance,
    xlabel="x", ylabel="y",
    title="Curl of Cyclone Velocity Field"
)

# Overlay vector field
quiver!(
    X, Y, quiver=(U, V),
    colour=:black,
    arrowsize=0.2,
    linealpha=0.8
)

```

Curl of Cyclone Velocity Field



```
# Question 2(c): Manual Curl of Cyclone Velocity Field
function velocity_vector(x, y)
    return [x, -y^2]
end

curl_manual(x, y) = (gradient(u -> velocity_vector(u[1], u[2])[2], [x,
y])[1]) -
                    (gradient(u -> velocity_vector(u[1], u[2])[1],
[x, y])[2])

x3 = -10:0.8:10
y3 = -10:0.8:10

U = [velocity_vector(x, y)[1] for x in x3, y in y3]
V = [velocity_vector(x, y)[2] for x in x3, y in y3]

U .*= 0.1
V .*= 0.1

X = [x for x in x3, y in y3]
Y = [y for x in x3, y in y3]

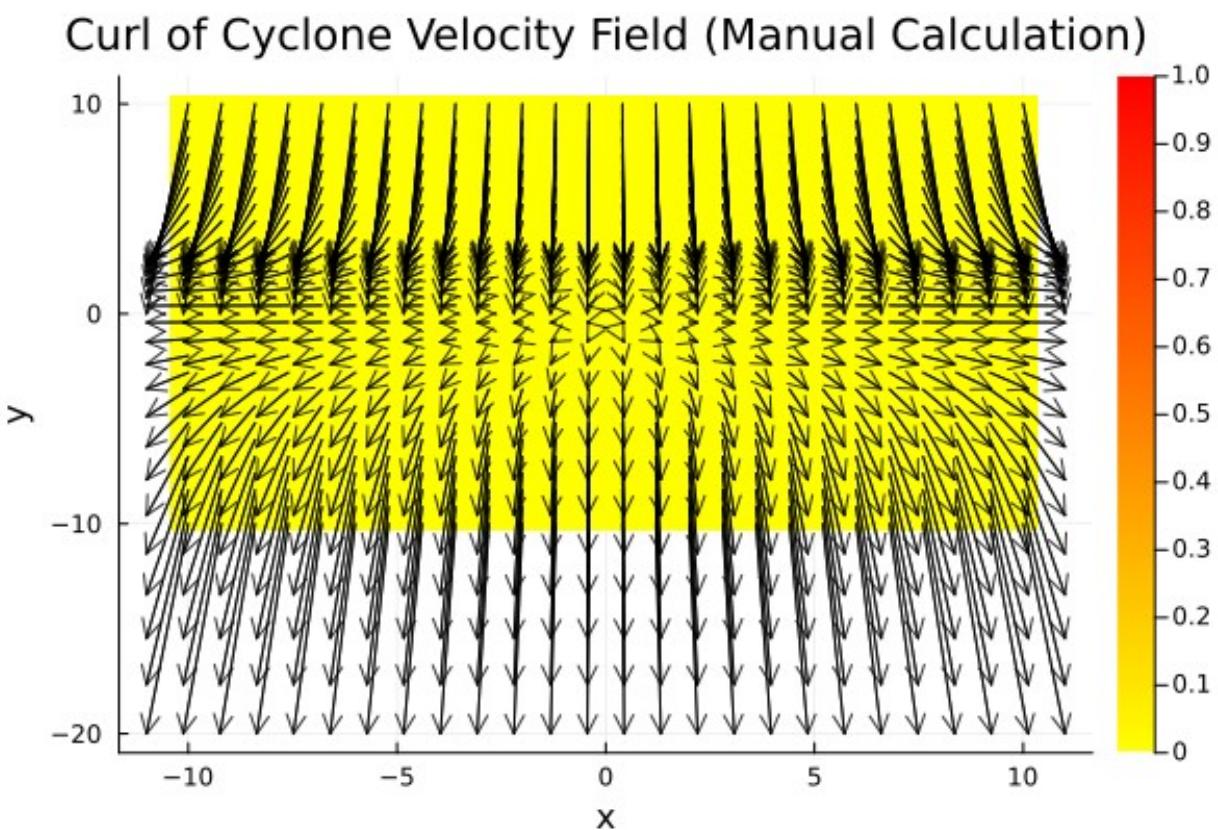
color_grad = cgrad([:yellow, :red])
# Heatmap of Manual Curl
heatmap(
```

```

x3, y3, curl_manual,
c = color_grad,
colour=:balance,
xlabel="x", ylabel="y",
title="Curl of Cyclone Velocity Field (Manual Calculation)"
)

# Overlay vector field
quiver!(
    X, Y, quiver=(U, V),
    colour=:black,
    arrowsize=0.2,
    linealpha=0.8
)

```



Question 3

```

# Question 3(a): River Velocity Vector Field
function velocity_vector(x,y)
    return [e^x*y^2,x + 2*y]
end

x3 = -2:0.2:2
y3 = -2:0.2:2

```

```

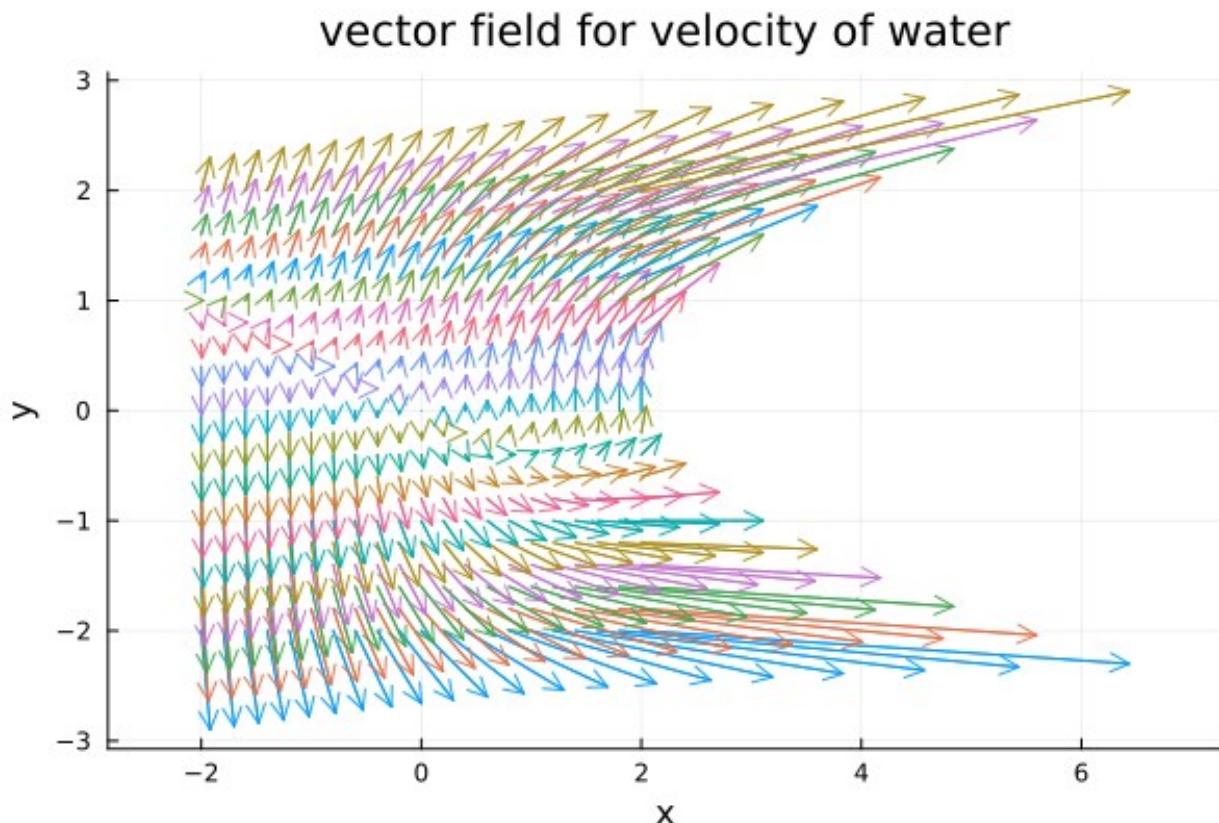
U = [velocity_vector(x,y)[1] for x in x3, y in y3]
V = [velocity_vector(x,y)[2] for x in x3, y in y3]

X = [x for x in x3, y in y3]
Y = [y for x in x3, y in y3]

scale = 0.15
U .*= scale
V .*= scale

quiver(X,Y, quiver = (U,V),
        title = "vector field for velocity of water",
        xlabel = "x", ylabel = "y",
        linealpha=0.9,
        arrowsize=0.2,
        aspect_ratio = :equal)

```



```

# Question 3(b): Divergence of River Velocity Field (Automatic)
function velocity_vector(x, y)
    return [exp(x*y^2), x + 2*y]
end

div(x,y) = divergence(u -> velocity_vector(u[1],u[2]),[x,y])

```

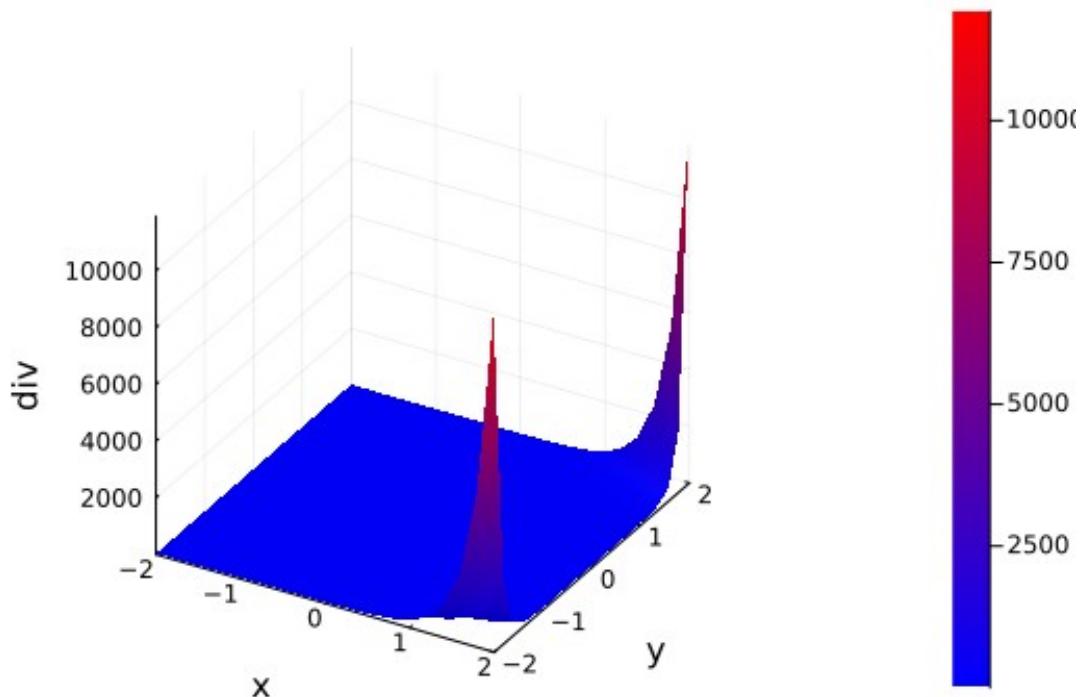
```

x4 = -2:0.2:2
y4 = -2:0.2:2

color_grad = cgrad(:blue, :red)
surface(x4, y4, div,
        c = color_grad, title = "Divergence of River Velocity(Automatic
Calculation)",
        xlabel = "x",
        ylabel = "y",
        zlabel = "div")

```

Divergence of River Velocity(Automatic Calculation)



```

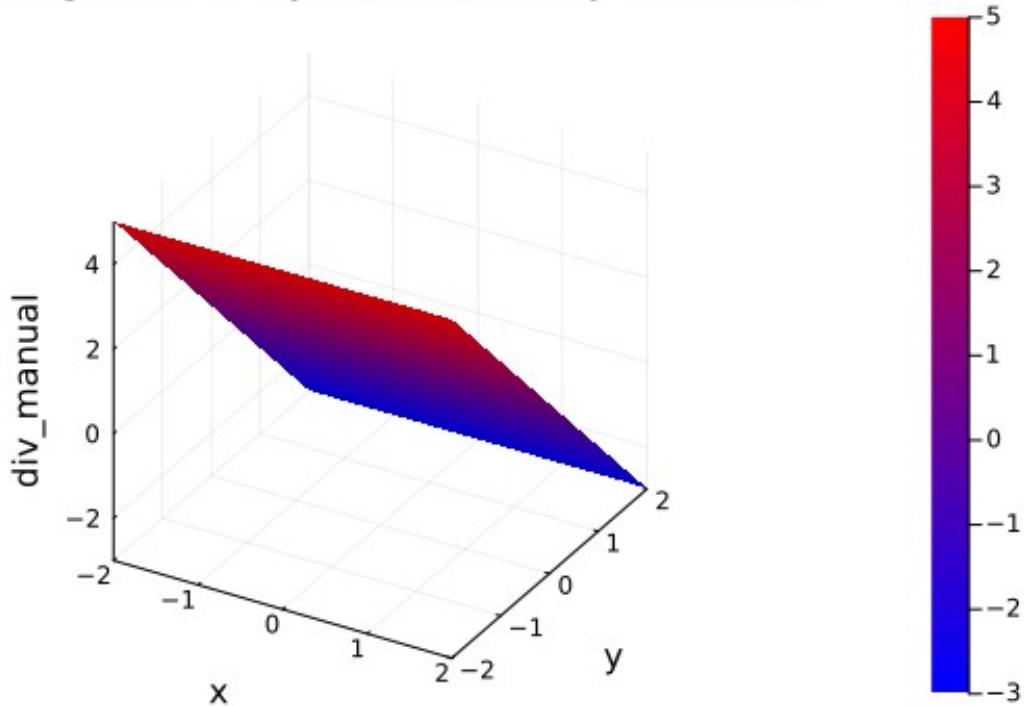
# Question 2(b): Divergence of Cyclone Velocity Field (Manual)
function velocity_vector(x, y)
    return [x, -y^2]
end

div_manual(x,y) = (gradient(u -> velocity_vector(u[1],u[2])[1],[x,y])
[1]) + (gradient(u -> velocity_vector(u[1],u[2])[2],[x,y])[2])

color_grad = cgrad(:blue, :red)
surface(x4, y4, div_manual, c = color_grad,
        title = "Divergence of Cyclone Velocity (Manual)",
        xlabel = "x", ylabel = "y", zlabel = "div_manual")

```

Divergence of Cyclone Velocity (Manual)



```
# Question 3(c): Curl of River Velocity Field (Automatic)
function velocity_vector(x, y)
    return [exp(x*y^2), x + 2*y]
end

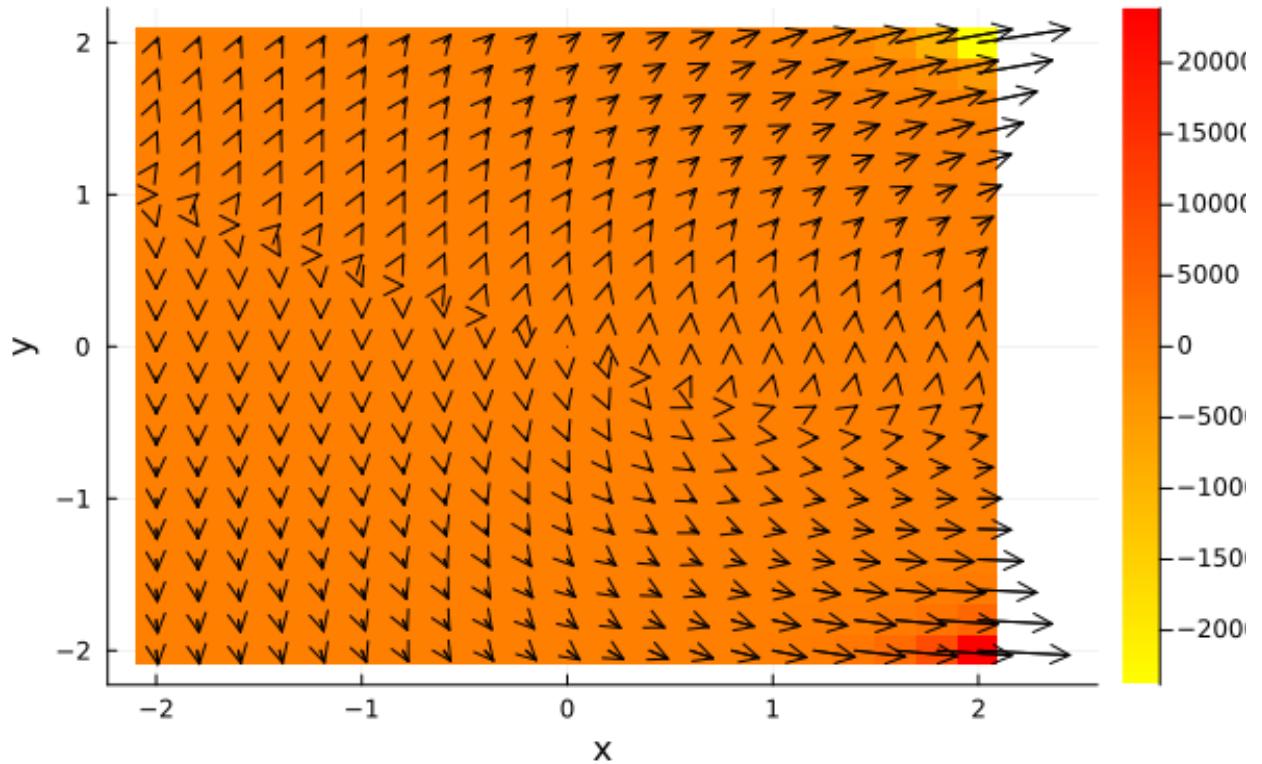
curl_auto(x,y) = curl(u -> velocity_vector(u[1],u[2]),[x,y])

color_grad = cgrad(:yellow, :red)
heatmap(x3,y3,curl_auto, colour=:balance, c = color_grad)

U .*= 0.1
V .*= 0.1

quiver!(X,Y, quiver = (U,V), colour=:black
    , title = "Curl of vector field",
    xlabel = "x", ylabel = "y")
```

Curl of vector field



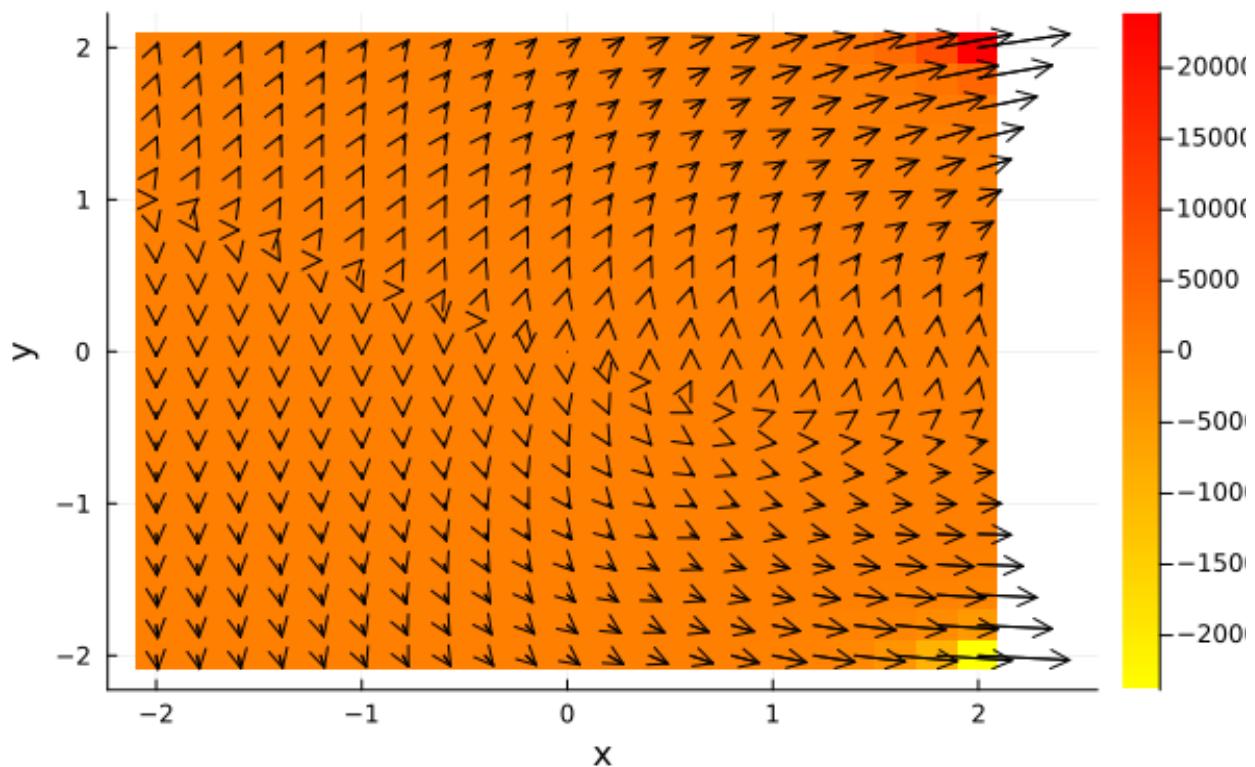
```
# Question 3
# (c) (using manual curl calculation)

curl_manual(x,y) = (gradient(u -> velocity_vector(u[1],u[2])[1],[x,y])
[2]) - (gradient(u -> velocity_vector(u[1],u[2])[2],[x,y])[1])

color_grad = cgrad([:yellow, :red])
heatmap(x3,y3,curl_manual, colour=:balance, c = color_grad)

quiver!(X,Y, quiver = (U,V), colour=:black, title = "Curl of vector
field (manual)", xlabel = "x", ylabel = "y")
```

Curl of vector field (manual)



Question 4

Plot BMD and SFD

```
using Plots

# Length of the main span in m
l = 10.0
# Uniformly distributed load in KN/m
q = 2.0
total_load = q * (1.25 * l)

R_B= (25 * q * l / 32)

R_A = total_load - R_B

println("Reaction at A (R_A): $R_A kN")
println("Reaction at B (R_B): $R_B kN")

function shear_force(x)
    if 0 <= x <= l
        return (15* q * l / 32 - q * x)
    else
# For the overhang section l < x <= 1.25*l
        return 40 * q * l / 32 - q * x
    end
end
```

```

    end
end

function bending_moment(x)
    if 0 <= x <= l
        return 15 * q * l * x / 32 - (q * x^2) / 2
    else
        return 15 * q * l / 32 * x + (25 * q * l / 32) * (x - l) - (q
    * x^2) / 2
    end
end

total_length = 1.25 * l

x_values = 0:0.01:total_length

V_values = shear_force.(x_values)
M_values = bending_moment.(x_values)

# Shear Force Diagram (SFD)
sfd_plot = plot(x_values, V_values,
    label="Shear Force",
    title="Shear Force Diagram (SFD)",
    xlabel="Position along beam X (m)",
    ylabel="Shear Force V (KN)",
    lw=2,
    color=:red,
    legend=:topright
)

hline!([0], color=:black, linestyle=:dash, label="")

# Bending Moment Diagram (BMD)

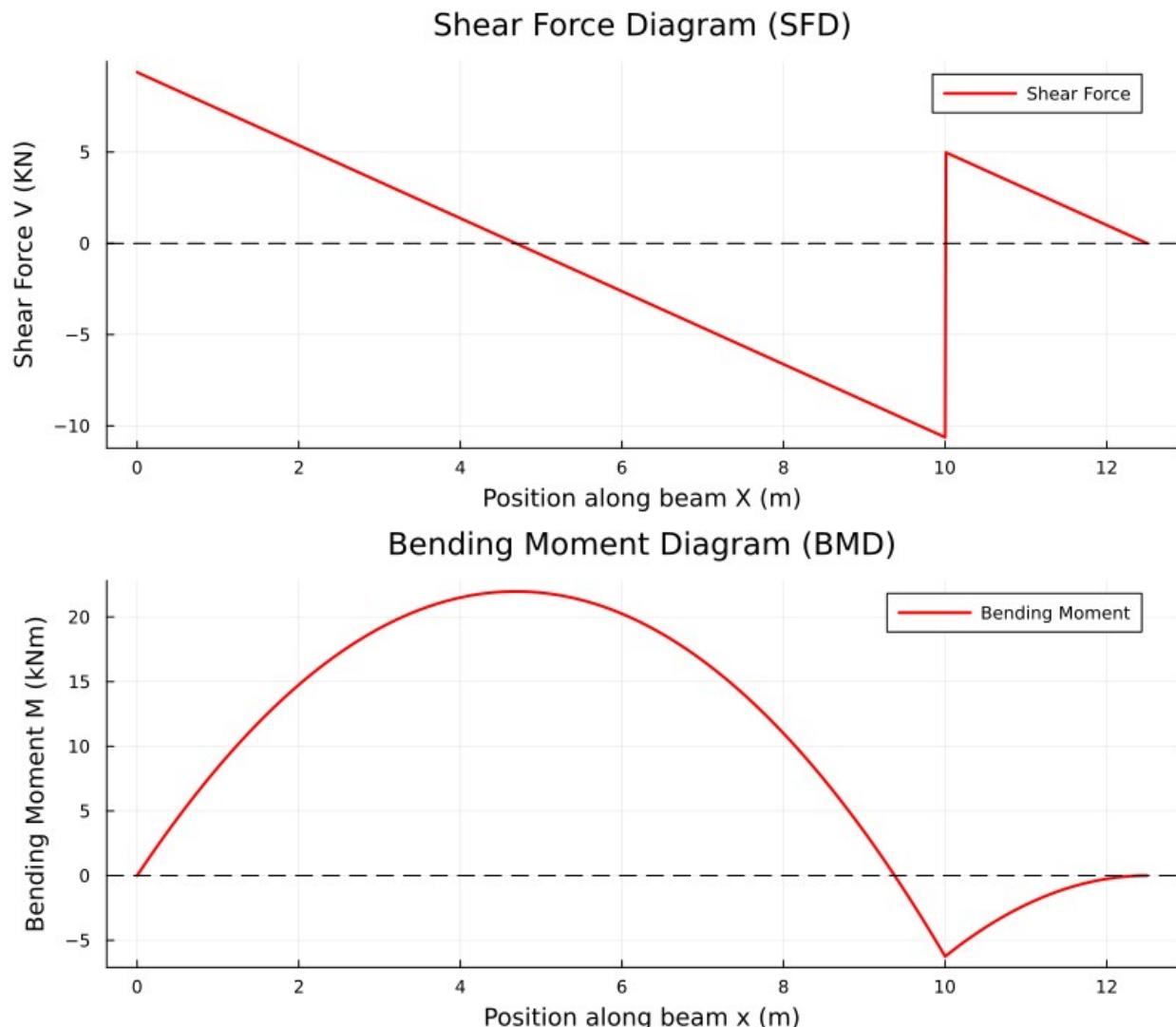
bmd_plot = plot(x_values, M_values,
    label="Bending Moment",
    title="Bending Moment Diagram (BMD)",
    xlabel="Position along beam x (m)",
    ylabel="Bending Moment M (kNm)",
    lw=2,
    color=:red,
    legend=:topright
)

hline!([0], color=:black, linestyle=:dash, label="")

```

```
plot(sfd_plot, bmd_plot, layout=(2, 1), size=(800, 700))
```

Reaction at A (R_A): 9.375 kN
Reaction at B (R_B): 15.625 kN



Calculations of Support Reactions

For a simply supported **overhang beam** of span (l) and overhung part of ($0.25l$), subjected to a uniformly distributed load (q):

Equilibrium Equations

$$\sum F_y = 0 \Rightarrow R_A + R_B - \frac{5}{4}ql = 0$$

$$R_A + R_B = \frac{5}{4}ql$$

Moment About A

$$\sum M_A = 0 \Rightarrow R_B \cdot l - ql \cdot \frac{(1.25l)}{2} = 0$$

$$R_B = \frac{3}{25}ql$$

Reactions

$$R_A = \frac{5}{4}ql - \frac{3}{25}ql = \frac{32}{15}ql$$

$$R_B = \frac{3}{25}ql$$

Shear Force and Bending Moment Equations

Let (x) be the distance measured from the left support (A).

For ($0 \leq x \leq l$) (within the main beam):

$$V(x) = R_A - qx$$

$$M(x) = R_Ax - \frac{qx^2}{2}$$

For ($x > l$) (overhanging part):

$$V(x) = R_A + R_B - qx$$

$$M(x) = R_A x + R_B(x - l) - \frac{qx^2}{2}$$

Question 5

Plot BMD and SFD

```
using Plots

# Length (in m)
l = 10.0
# Uniformly distributed load (kN/m)
q = 5.0

R_A = (0.8 * q * l * (0.4 * l)) / (0.8 * l)
R_C = (R_A*l - (0.8*q*l)*(0.6*l) + (q*l)*(0.5*l)) / l
R_B = (0.8 * q * l) + (q * l) - R_A - R_C

println("Reaction at A (R_A): $R_A kN")
println("Reaction at B (R_B): $R_B kN")
println("Reaction at C (R_C): $R_C kN")
println("Net reactions: $(R_A+R_B+R_C) kN")
println("Total load: $((0.8*q*l) + (q*l)) kN")

function shear_force(x)
    if 0 <= x < 0.4*l
        # Segment A-E
        return R_A
    elseif 0.4*l <= x < l
        # Segments E-D and D-B
        return R_A - (0.8 * q * l)
    elseif l <= x <= 2*l
        # Segment B-C
        return R_A - (0.8 * q * l) + R_B - q * (x - l)
    else
        return 0.0
    end
end

function bending_moment(x)
    if 0 <= x < 0.4*l
        # Segment A-E
        return R_A * x
    elseif 0.4*l <= x < l
        # Segments E-D and D-B
        return R_A * x - (0.8 * q * l) * (x - 0.4 * l)
    elseif l <= x <= 2*l
        # Segment B-C
        return R_A * x - (0.8 * q * l) * (x - l) + R_B * (x - l)
    else
        return 0.0
    end
end
```

```

        return R_A * x - (0.8 * q * l) * (x - 0.4 * l) + R_B * (x - l)
- q * (x - l)^2 / 2
else
    return 0.0
end
end

total_length = 2 * l
x_values = 0:total_length/1000:total_length

V_values = shear_force.(x_values)
M_values = bending_moment.(x_values)

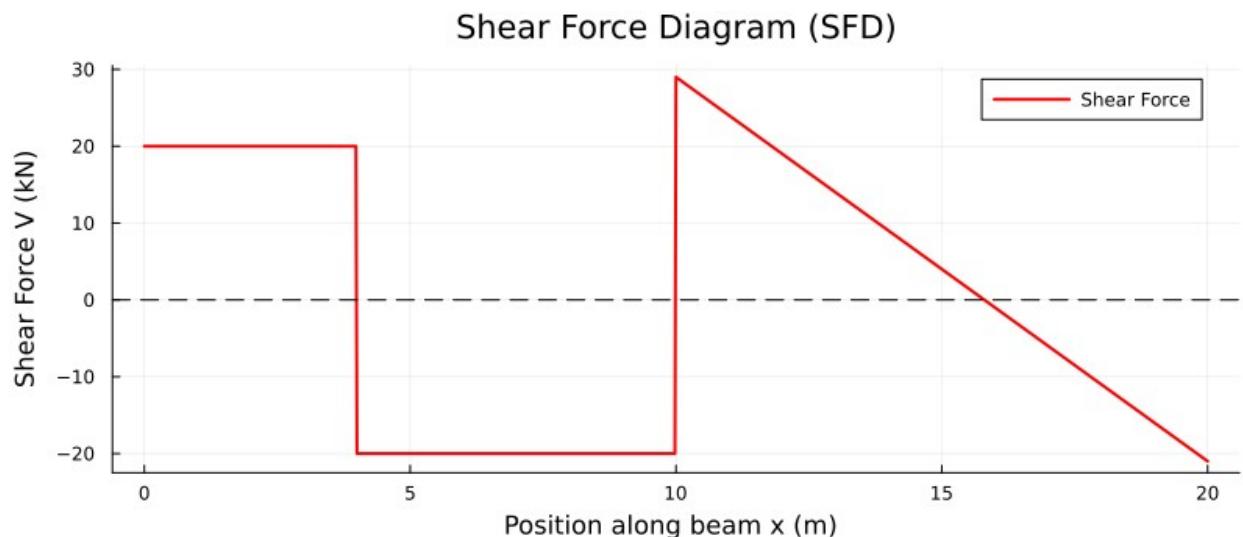
# Shear Force Diagram (SFD)
sfd_plot = plot(x_values, V_values,
    label="Shear Force",
    title="Shear Force Diagram (SFD)",
    xlabel="Position along beam x (m)",
    ylabel="Shear Force V (kN)",
    lw=2,
    color=:red,
    legend=:topright,
)
hline!([0], color=:black, linestyle=:dash, label="")

# Bending Moment Diagram (BMD)
bmd_plot = plot(x_values , M_values,
    label="Bending Moment",
    title="Bending Moment Diagram (BMD)",
    xlabel="Position along beam x (m)",
    ylabel="Bending Moment M (kNm)",
    lw=2,
    color=:red,
    legend=:topright
)
hline!([0], color=:black, linestyle=:dash, label="")

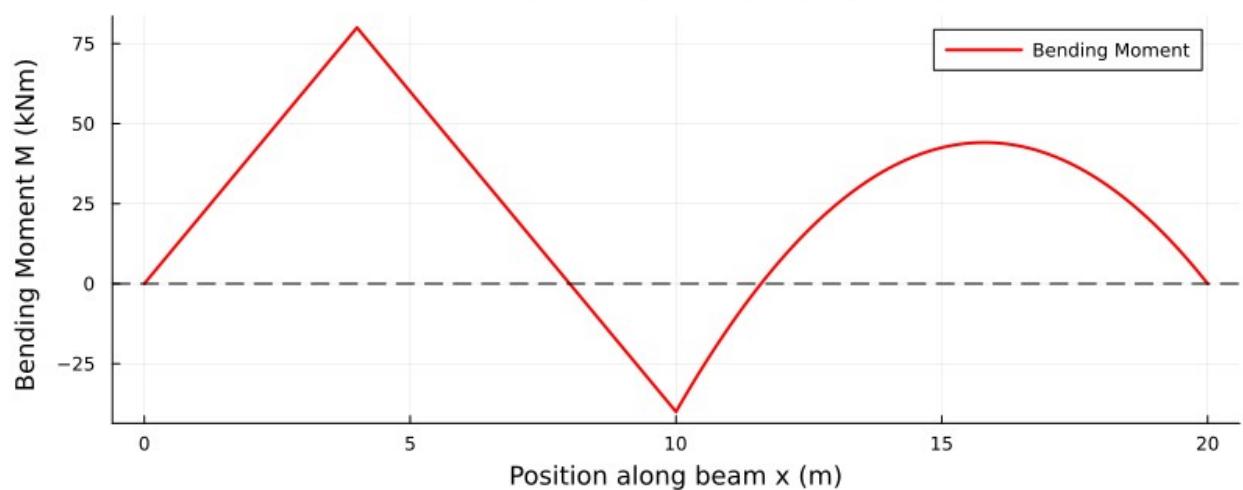
plot(sfd_plot, bmd_plot, layout=(2, 1), size=(800, 700))

Reaction at A (R_A): 20.0 kN
Reaction at B (R_B): 49.0 kN
Reaction at C (R_C): 21.0 kN
Net reactions: 90.0 kN
Total load: 90.0 kN

```



Bending Moment Diagram (BMD)



Calculations of Support Reactions

For the beam under a uniformly distributed load (q):

Equilibrium Equation

$$\begin{aligned}\sum F_y = 0 \Rightarrow R_A + R_B + R_C - 1.8ql &= 0 \\ R_A + R_B + R_C &= 1.8ql \quad (1)\end{aligned}$$

Moment About D (First Equation)

$$\sum M_D = 0 \Rightarrow R_A(0.8l) - (0.8ql)(0.4l) = 0$$

$$R_A = 0.4ql \quad (2)$$

Moment About D (Second Equation)

$$\sum M_D = 0 \Rightarrow R_B(0.2l) + R_C(1.2l) - 0.7ql = 0$$

$$R_B(0.2l) + R_C(1.2l) = 0.7ql \quad (3)$$

Solving Equations (1), (2), and (3):

$$R_A = 0.4ql$$

$$R_B = 0.98ql$$

$$R_C = 0.42ql$$

Shear Force and Bending Moment Equations

Let (x) be the distance measured from the left support (A).

For ($0 \leq x \leq 0.4l$):

$$V(x) = R_A$$

$$M(x) = R_A x$$

For ($0.4l \leq x \leq 0.8l$):

$$V(x) = R_A - 0.8ql$$

$$M(x) = R_A x - 0.8ql(x - 0.4l)$$

For ($0.8l \leq x \leq l$):

$$V(x) = R_A - 0.8ql$$

$$M(x) = R_A x - P(x - 0.4l)$$

For ($x > l$):

$$V(x) = R_A - P + R_B - q(x - l)$$

$$M(x) = R_A x - 0.8ql(x - 0.4l) + R_B(x - l) - \frac{q(x - l)^2}{2}$$

