

Assignment 1

22ce01024_Thanvi

```
In [7]: using Plots
using CalculusWithJulia

# Question-1
# (a)

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

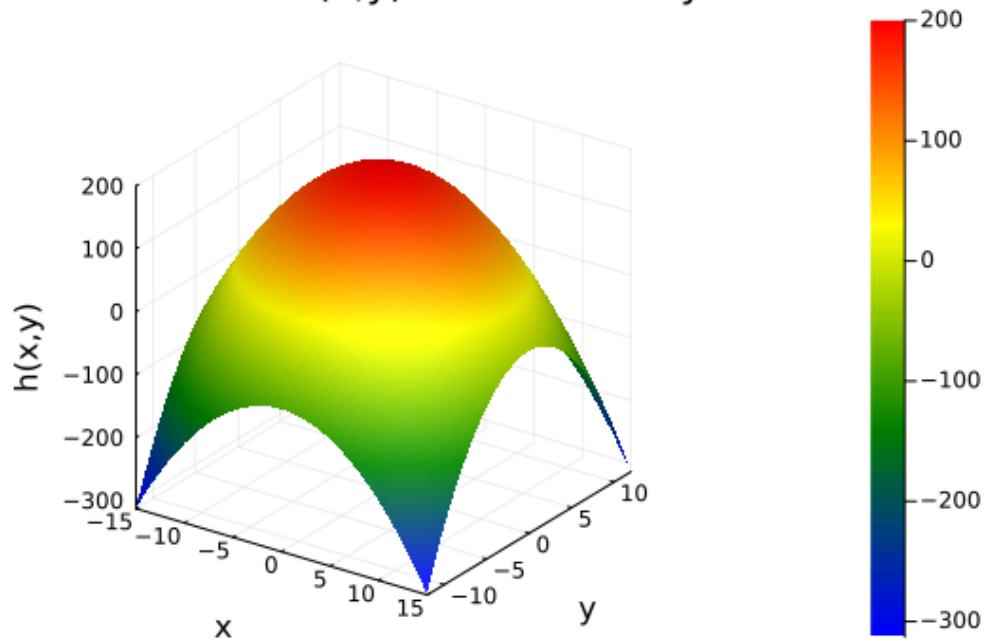
x1 = -15:0.1:150)
y1 = -12:0.1:12

# 3D Surface Plot with better colors
color_grad = cgrad([:blue, :green, :yellow, :red])
p1 = surface(x1, y1, h,
             c = color_grad,
             xlabel = "x",
             ylabel = "y",
             zlabel = "h(x,y)",
             title = "3D Surface:  $h(x,y) = 200 - x^2 - 2y^2$ ",
             camera = (35, 25))

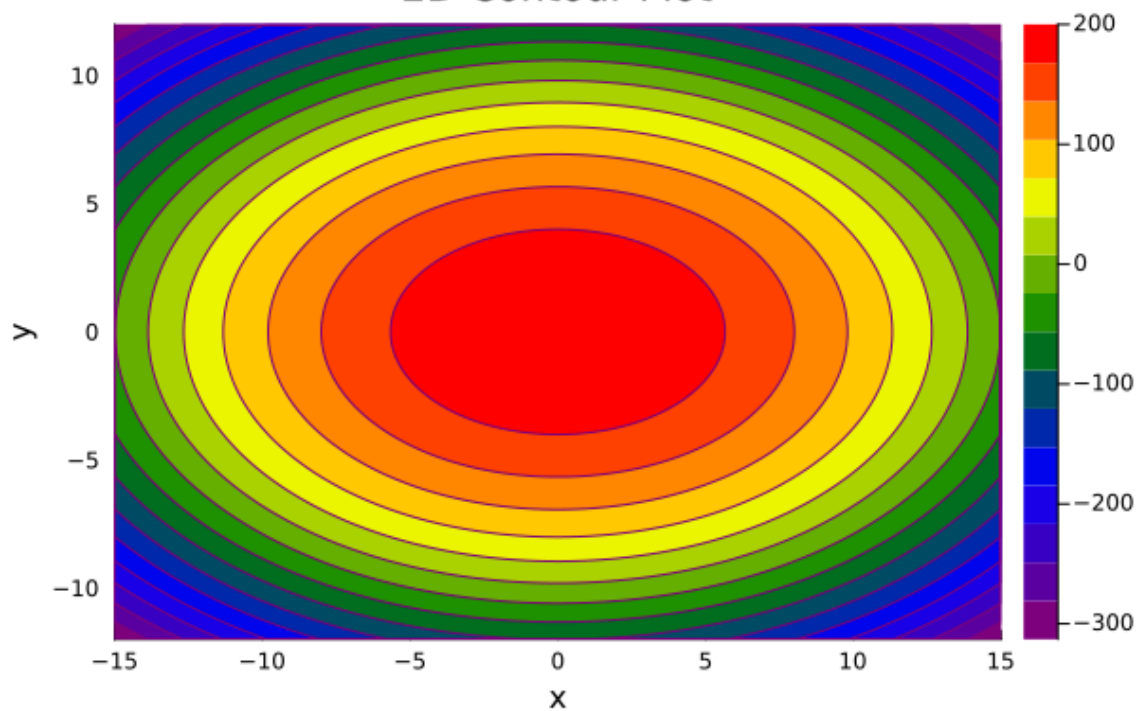
# 2D Contour Plot with improved colors
p2 = contour(x1, y1, h,
             fill = true,
             c = cgrad([:purple, :blue, :green, :yellow, :red]),
             xlabel = "x",
             ylabel = "y",
             title = "2D Contour Plot")

# Display both plots
display(p1)
display(p2)
```

3D Surface: $h(x,y) = 200 - x^2 - 2y^2$



2D Contour Plot



```
In [4]: h(1,1)

gradient(u -> h(u[1],u[2]),[1,1])
```

```
Out[4]: 2-element Vector{Int64}:
 -2
 -4
```

```
In [5]: # Question 1
# (b) & (c)

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]

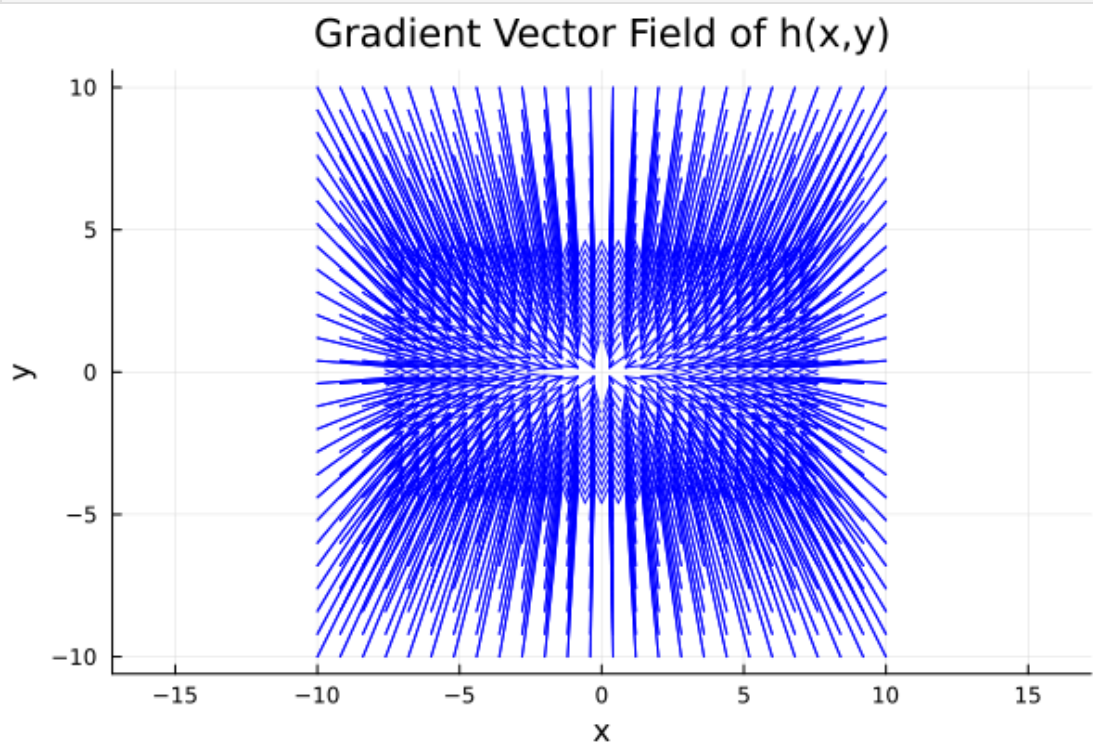
X = [x for x in x2, y in y2]
Y = [y for x in x2, y in y2]

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

quiver(X, Y, quiver = (U, V),
        linealpha = 0.9,
        arrowsize = 0.5,
        aspect_ratio = :equal,
        title = "Gradient Vector Field of h(x,y)",
        xlabel = "x",
        ylabel = "y",
        color = :blue)

```

Out[5]:



```

In [6]: # Question 2
# (a)

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

```

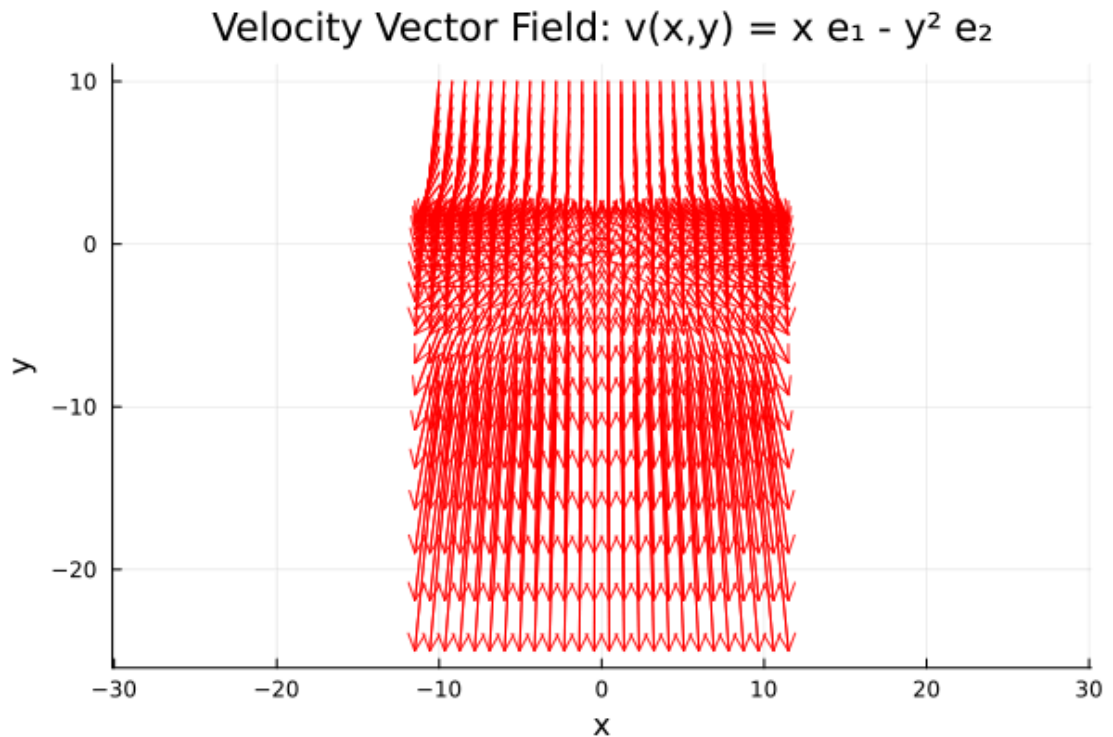
```

V .*= scale

quiver(X, Y, quiver = (U, V),
       title = "Velocity Vector Field:  $v(x,y) = x e_1 - y^2 e_2$ ",
       xlabel = "x",
       ylabel = "y",
       linealpha = 0.9,
       arrowsize = 0.2,
       aspect_ratio = :equal,
       color = :red)

```

Out[6]:



```

In [27]: # Question 2
# (b) (using automatic divergence calculation)

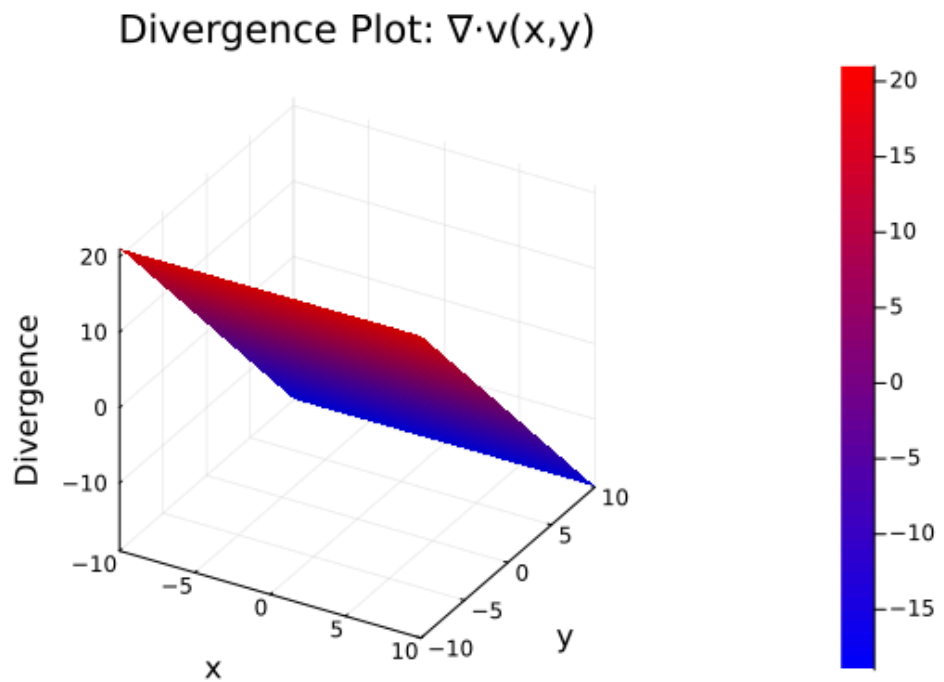
div(x,y) = divergence(u -> velocity_vector(u[1],u[2]),[x,y])

x4 = -10:0.8:10
y4 = -10:0.8:10

color_grad = cgrad([:blue, :red])
surface(x4, y4, div,
       c = color_grad,
       title = "Divergence Plot:  $\nabla \cdot v(x,y)$ ",
       xlabel = "x",
       ylabel = "y",
       zlabel = "Divergence")

```

Out[27]:

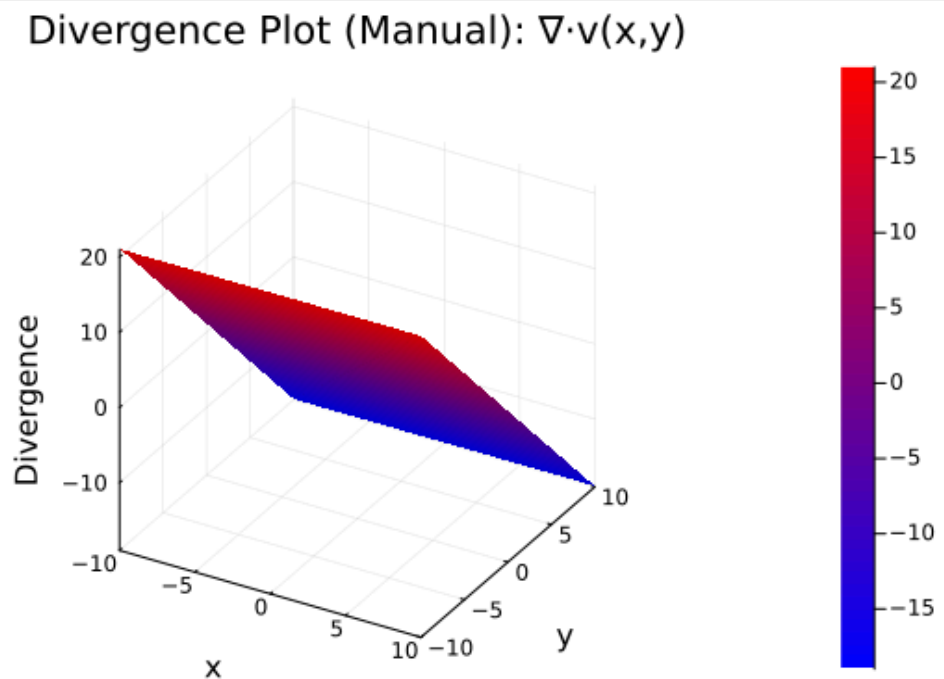


```
In [28]: # Question 2
# (b) (detailed calculation)

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

color_grad = cgrad([:blue, :red])
surface(x4, y4, div_manual,
        c = color_grad,
        title = "Divergence Plot (Manual):  $\nabla \cdot \mathbf{v}(x,y)$ ",
        xlabel = "x",
        ylabel = "y",
        zlabel = "Divergence")
```

Out[28]:



```

In [30]: # Question 2
# (c) (using automatic curl calculation)

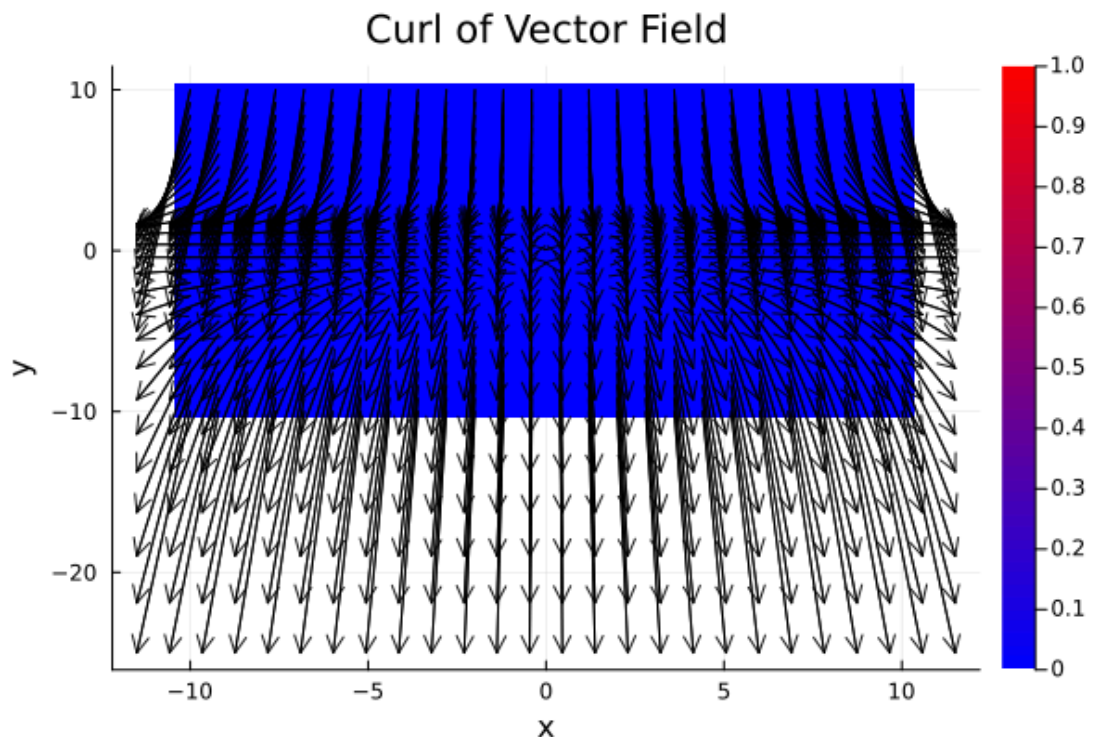
curl_auto(x,y) = curl(u -> velocity_vector(u[1],u[2]),[x,y])

color_grad = cgrad([:blue, :red])
p = heatmap(x3, y3, curl_auto,
            c = color_grad,
            title = "Curl of Vector Field",
            xlabel = "x",
            ylabel = "y")

quiver!(p, X, Y, quiver = (U, V),
        color = :black,
        arrowsize = 0.2,
        linealpha = 0.9)

```

Out[30]:



```

In [31]: # Question 2
# (c) (using manual curl calculation)

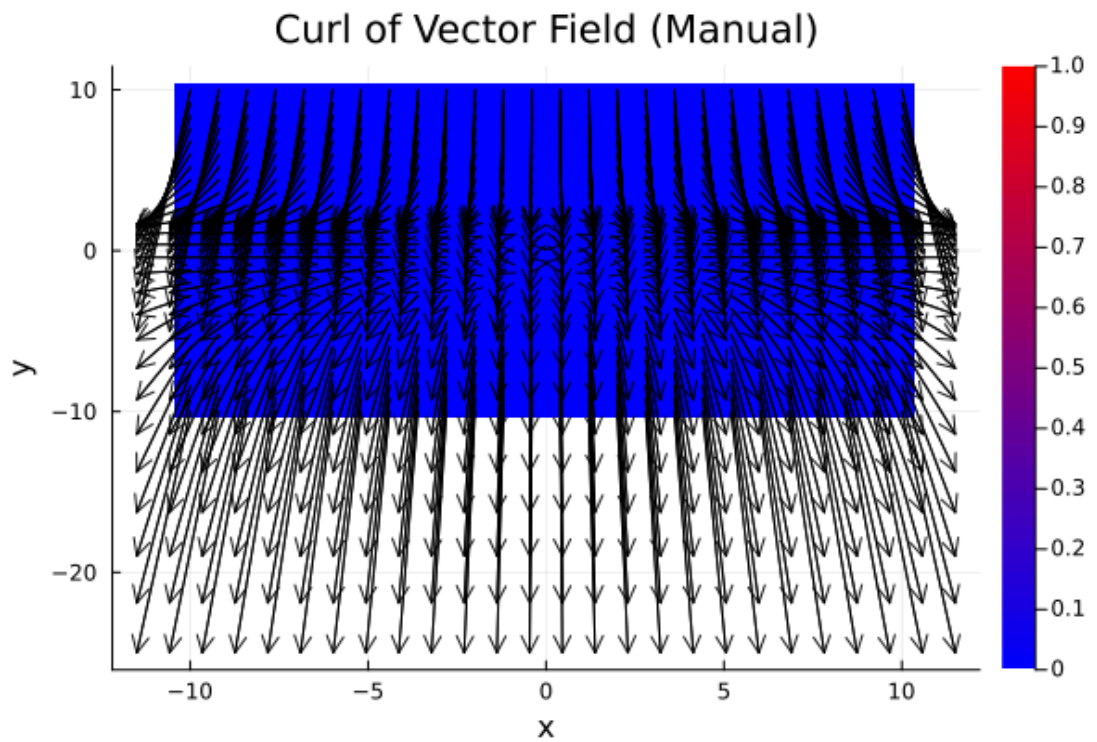
curl_manual(x,y) = (gradient(u -> velocity_vector(u[1],u[2])[1],[x,y])[2]) - (gr

color_grad = cgrad([:blue, :red])
p = heatmap(x3, y3, curl_manual,
            c = color_grad,
            title = "Curl of Vector Field (Manual)",
            xlabel = "x",
            ylabel = "y")

quiver!(p, X, Y, quiver = (U, V),
        color = :black,
        arrowsize = 0.2,
        linealpha = 0.9)

```

Out[31]:



In [32]:

```
# question 3
# (a)

function velocity_vector(x,y)
    return [exp(x)*y^2, x + 2*y]
end

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

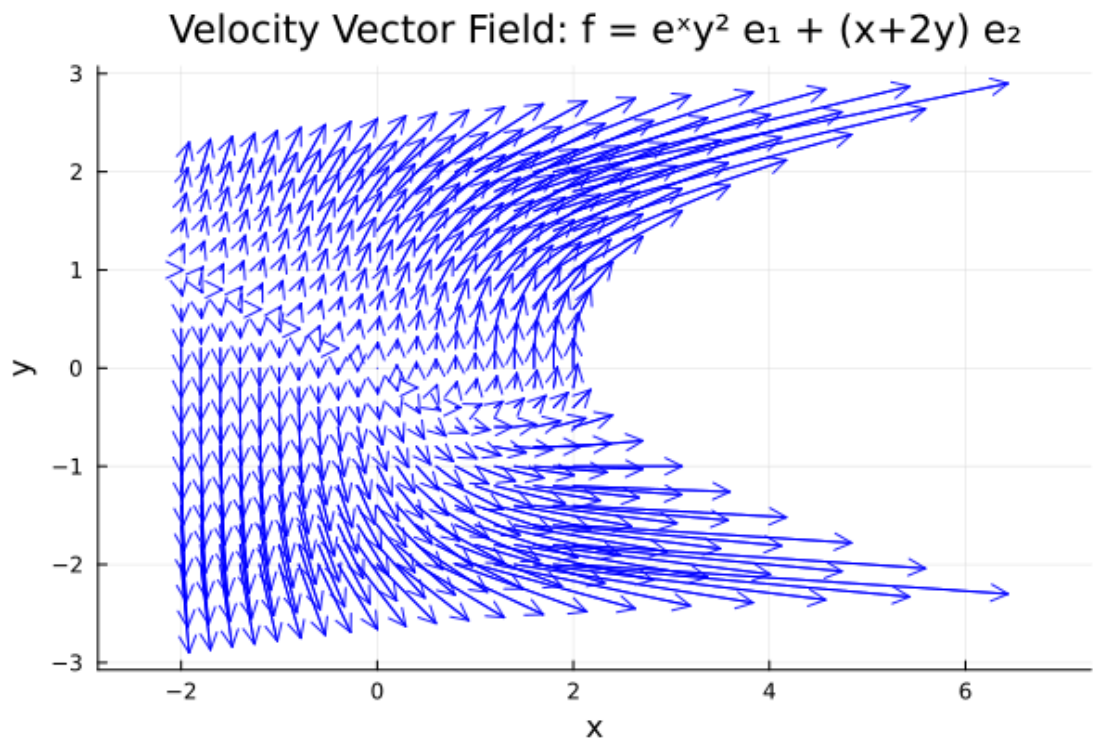
U = [velocity_vector(x,y)[1] for x in x3, y in y3]
V = [velocity_vector(x,y)[2] for x in x3, y in y3]

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 = "Velocity Vector Field:  $f = e^{xy^2} e_1 + (x+2y) e_2$ ",
        xlabel = "x",
        ylabel = "y",
        linealpha = 0.9,
        arrowsize = 0.2,
        aspect_ratio = :equal,
        color = :blue)
```

Out[32]:



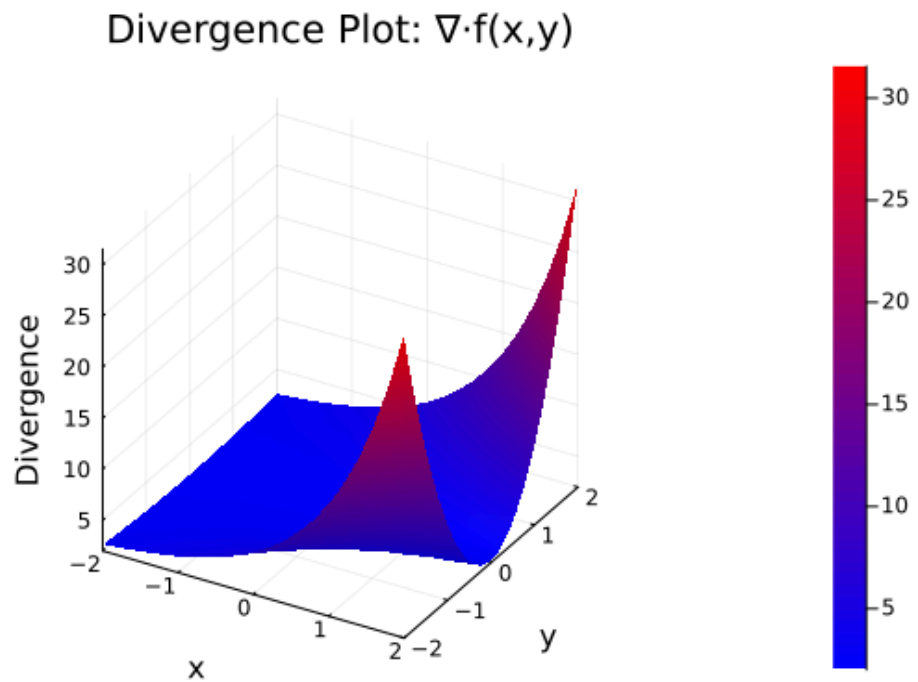
```
In [33]: # Question 3
# (b) (using automatic divergence calculation)

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 Plot:  $\nabla \cdot f(x,y)$ ",
        xlabel = "x",
        ylabel = "y",
        zlabel = "Divergence")
```

Out[33]:

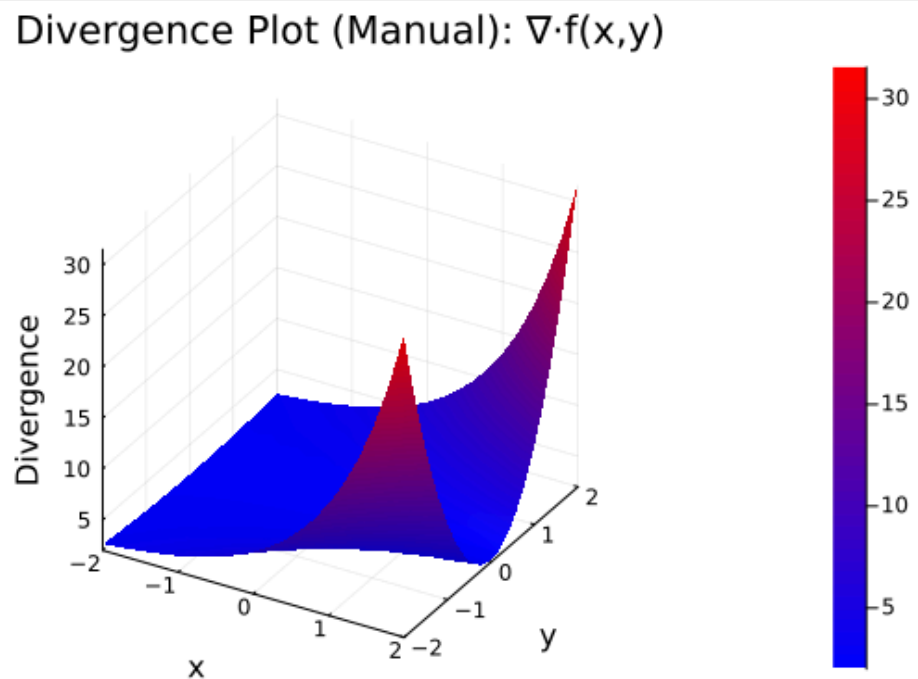


```
In [34]: # Question 3
# (b) (detailed calculation)

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

color_grad = cgrad([:blue, :red])
surface(x4, y4, div_manual,
        c = color_grad,
        title = "Divergence Plot (Manual):  $\nabla \cdot f(x,y)$ ",
        xlabel = "x",
        ylabel = "y",
        zlabel = "Divergence")
```

Out[34]:



```

In [35]: # Question 3
# (c) (using automatic curl calculation)

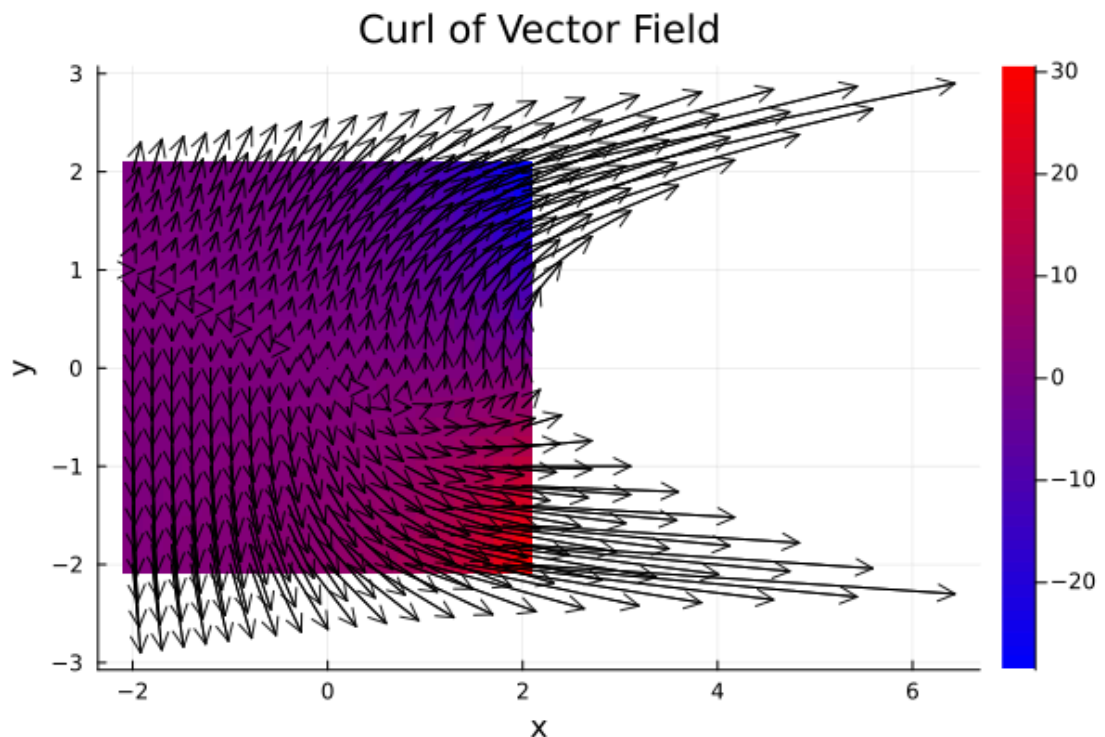
curl_auto(x,y) = curl(u -> velocity_vector(u[1],u[2]),[x,y])

color_grad = cgrad([:blue, :red])
p = heatmap(x3, y3, curl_auto,
            c = color_grad,
            title = "Curl of Vector Field",
            xlabel = "x",
            ylabel = "y")

quiver!(p, X, Y, quiver = (U, V),
        color = :black,
        arrowsize = 0.2,
        linealpha = 0.9)

```

Out[35]:



```

In [36]: # Question 3
# (c) (using manual curl calculation)

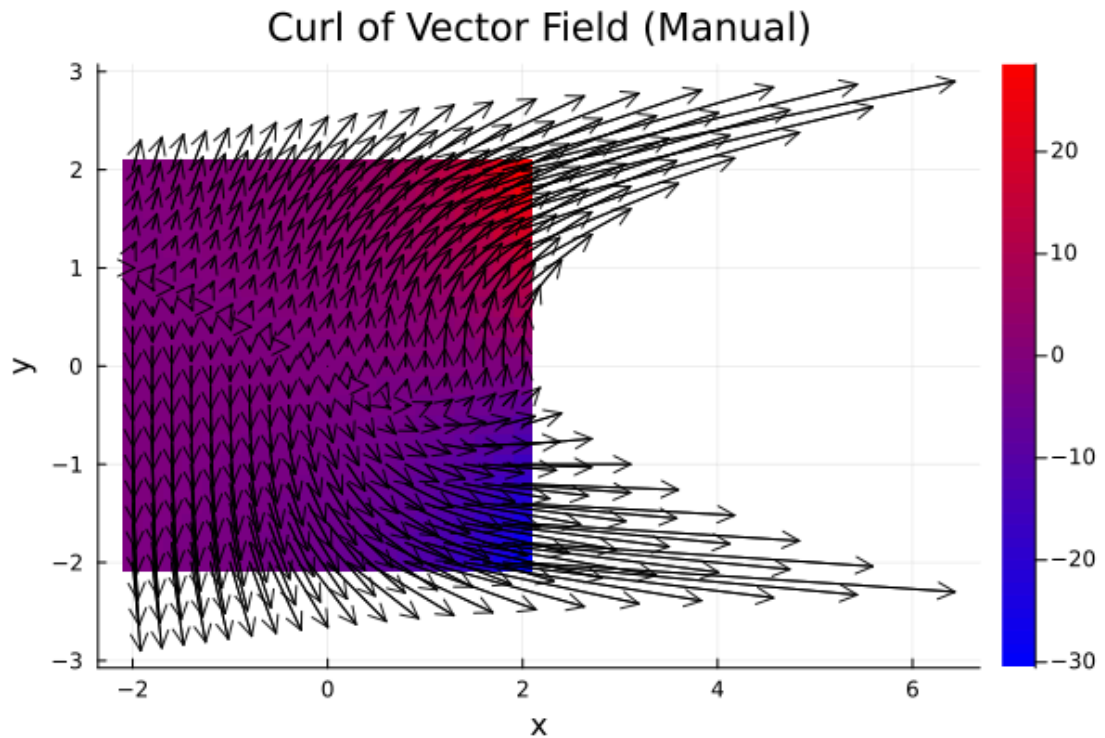
curl_manual(x,y) = (gradient(u -> velocity_vector(u[1],u[2])[1],[x,y])[2]) - (gr

color_grad = cgrad([:blue, :red])
p = heatmap(x3, y3, curl_manual,
            c = color_grad,
            title = "Curl of Vector Field (Manual)",
            xlabel = "x",
            ylabel = "y")

quiver!(p, X, Y, quiver = (U, V),
        color = :black,
        arrowsize = 0.2,
        linealpha = 0.9)

```

Out[36]:



In [37]: **using** Plots

```
# Question 4: Beam Problem (Generic function)
function solve_beam_problem(l=10.0, 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 <= 1
            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

    function bending_moment(x)
        if 0 <= x <= 1
            return 15 * q * l * x / 32 - (q * x^2) / 2
        else
            return 15 * q * l / 32 * x + (25 * q * l / 32) * (x - 1) - (q * x^2)
        end
    end

    total_length = 1.25 * l
    x_vals = 0:0.01:total_length
    V_vals = shear_force.(x_vals)
    M_vals = bending_moment.(x_vals)

    # Shear Force Diagram (SFD)
    sfd_plot = plot(x_vals, V_vals,
        label="Shear Force",
        title="Shear Force Diagram (SFD)\n(l=$l m, q=$q kN/m)",
        xlabel="Position along beam (m)",
```

```

        ylabel="Shear Force (kN)",
        lw=2,
        color=:blue,
        legend=:topright
    )
    hline!([0], color=:black, linestyle=:dash, label="")

    # Bending Moment Diagram (BMD)
    bmd_plot = plot(x_vals, M_vals,
        label="Bending Moment",
        title="Bending Moment Diagram (BMD)\n(l=$l m, q=$q kN/m)",
        xlabel="Position along beam (m)",
        ylabel="Bending Moment (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))
end

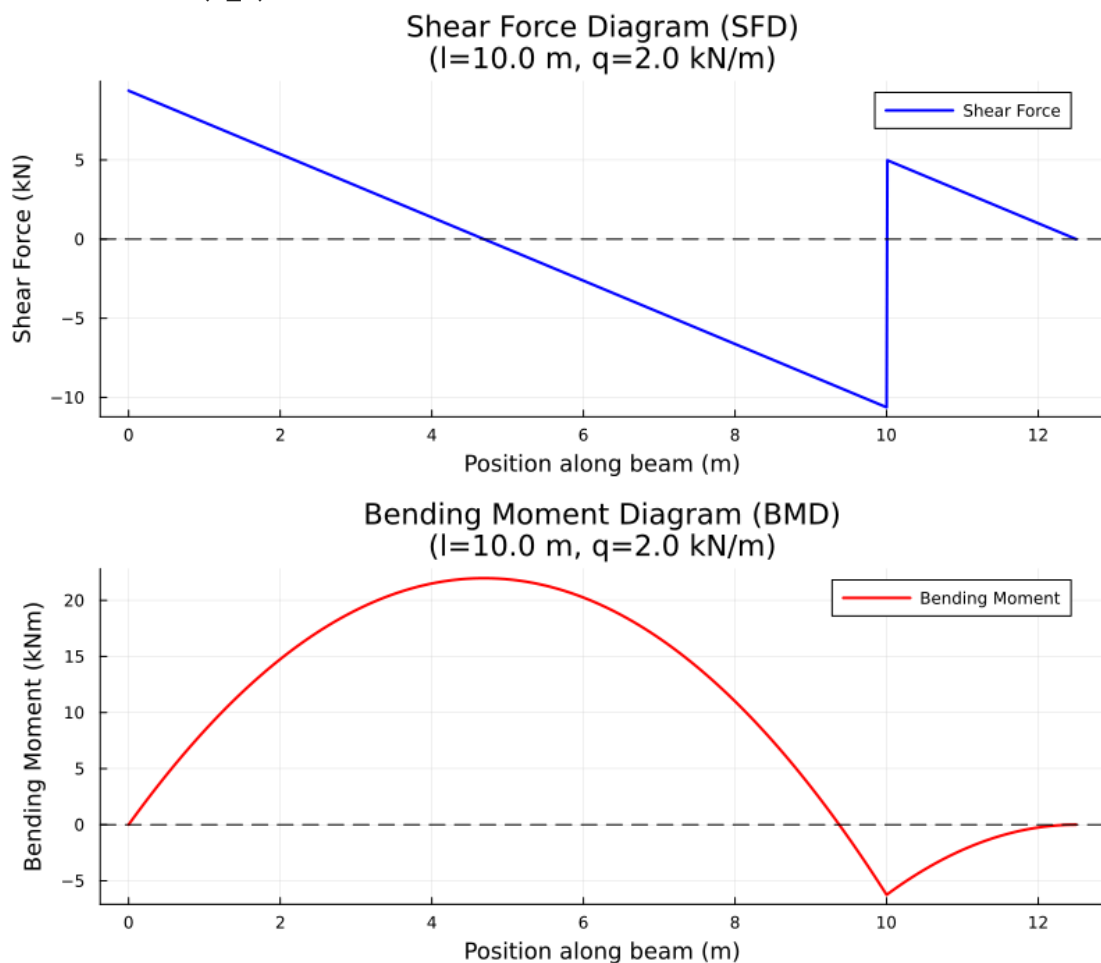
# Solve for default values
solve_beam_problem()

```

Reaction at A (R_A): 9.375 kN

Reaction at B (R_B): 15.625 kN

Out[37]:



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

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

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

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):

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

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

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

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

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

In [38]: **using** Plots

```
# Question 5: Beam Problem (Generic function)
function solve_beam_problem_2(l=10.0, q=5.0)
    # Reactions calculation
    RA = (0.8 * q * l * (0.4 * l)) / (0.8 * l)
    RC = (RA * l - (0.8 * q * l) * (0.6 * l) + (q * l) * (0.5 * l)) / l
    RB = (0.8 * q * l) + (q * l) - RA - RC

    println("Reaction at A (RA): $RA kN")
    println("Reaction at B (RB): $RB kN")
    println("Reaction at C (RC): $RC kN")
    println("Net reactions: $(RA+RB+RC) 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 RA
        elseif 0.4*l <= x < l      # Segments E-D and D-B
            return RA - (0.8 * q * l)
        elseif l <= x <= 2*l       # Segment B-C
            return RA - (0.8 * q * l) + RB - q * (x - l)
        else
            return 0.0
        end
    end

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

    total_length = 2 * l
    x_vals = 0:total_length/1000:total_length
    V_vals = shear_force.(x_vals)
    M_vals = bending_moment.(x_vals)

    # Shear Force Diagram (SFD)
    sfd_plot = plot(x_vals, V_vals,
        label="Shear Force",
        title="Shear Force Diagram (SFD)\n(l=$l m, q=$q kN/m)",
        xlabel="Position along beam (m)",
        ylabel="Shear Force (kN)",
        lw=2,
        color=:blue,
        legend=:topright,
    )
    hline!([0], color=:black, linestyle=:dash, label="")

    # Bending Moment Diagram (BMD)
    bmd_plot = plot(x_vals, M_vals,
        label="Bending Moment",
```

```

    title="Bending Moment Diagram (BMD)\n(l=$l m, q=$q kN/m)",
    xlabel="Position along beam (m)",
    ylabel="Bending Moment (kNm)",
    lw=2,
    color=:red,
    legend=:topright
)
hline!([0], color=:black, linestyle=:dash, label="")

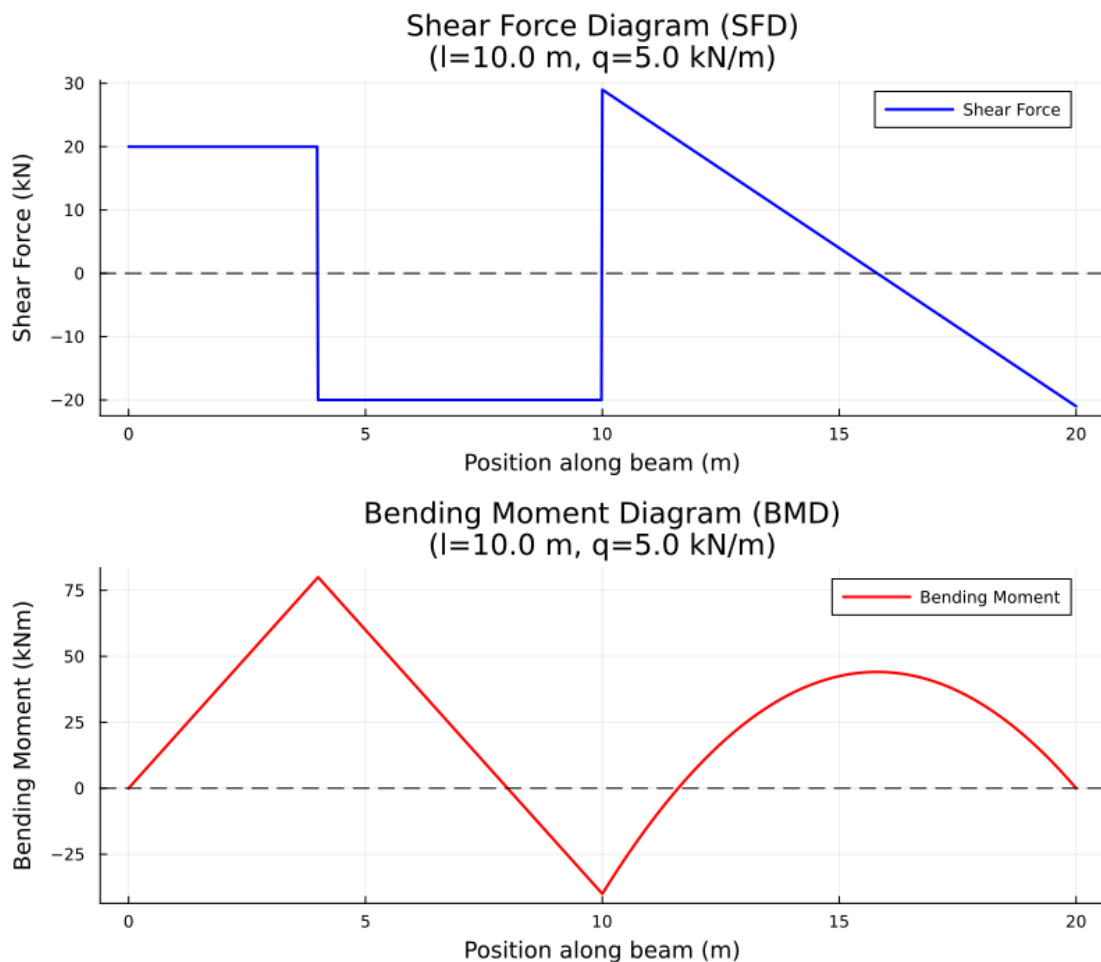
plot(sfd_plot, bmd_plot, layout=(2, 1), size=(800, 700))
end

# Solve for default values
solve_beam_problem_2()

```

Reaction at A (RA): 20.0 kN
 Reaction at B (RB): 49.0 kN
 Reaction at C (RC): 21.0 kN
 Net reactions: 90.0 kN
 Total load: 90.0 kN

Out[38]:



Calculations of Support Reactions

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

Equilibrium Equation

$$\sum F_y = 0 \Rightarrow R_A + R_B + R_C - 1.8ql = 0$$

$$R_A + R_B + R_C = 1.8ql \quad (1)$$

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 $(x > l)$:

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

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