IIT Hyderabad

Assignment 2

Submitted by:
ME21BTECH11001 Abhishek Ghosh

ME5053: Soft Robotics
Mechanical Engineering
24.04.2025

Submitted to:
Dr. Prabhat Kumar

1 Connectivity Matrix for Quadrilateral Mesh

In a structured 2D mesh of quadrilateral elements, each element connects 4 corner nodes. For a mesh of $nx \times ny$ quadrilateral elements, the total number of elements is $nx \cdot ny$.

The node indexing follows a left-to-right, bottom-to-top pattern. For each element, the node ordering is:

- node1: bottom-left
- node2: bottom-right
- node3: top-right
- node4: top-left

MATLAB Code:

```
function conn = connectivity2D(nx, ny)
    conn = zeros(nx * ny, 4);
    element_index = 1;

    for j = 1:ny
        for i = 1:nx
            node1 = (j - 1) * (nx + 1) + i;
            node2 = node1 + 1;
            node3 = node2 + (nx + 1);
            node4 = node3 - 1;

            conn(element_index, :) = [node1, node2, node3, node4];
            element_index = element_index + 1;
        end
    end
end
```

1	2	6	5
2	3	7	6
3	4	8	7
5	6	10	9
6	7	11	10
7	8	12	11

Figure 1: Connectivity Matrix for nx = 3 and ny = 2

2 Gradient and Direction of Maximum Increase

Let $f(\vec{x})$ be a scalar function, where $\vec{x} \in \mathbb{R}^n$. The gradient ∇f is defined as:

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

The directional derivative in the direction of a unit vector $\vec{u}$ is:

$$D_{\vec{u}} f = \nabla f \cdot \vec{u} = \|\nabla f\| \cos\theta$$

where θ is the angle between ∇f and $\vec{u}$.

Conclusion:

The directional derivative is maximum when $\cos\theta = 1$, i.e., when $\vec{u}$ points in the direction of ∇f. Therefore, the gradient points in the direction of maximum increase of the function.The red arrows show the gradient vectors pointing in the direction of maximum increase of the function f.

2D and 3D Gradient Visualization in MATLAB

2D Visualization:

Function: $f(x, y) = x^2 + y^2$

```matlab
% Define the scalar function
f = @(x, y) x.^2 + y.^2;

% Generate a grid
[x, y] = meshgrid(-2:0.2:2, -2:0.2:2);

% Compute gradients
[fx, fy] = gradient(f(x, y), 0.2, 0.2);

% Plot the scalar field
contour(x, y, f(x, y), 20); hold on;
quiver(x, y, fx, fy, 'r'); % Gradient vectors
title('Gradient vectors of f(x,y) = x^2 + y^2');
xlabel('x'); ylabel('y');
axis equal; grid on;
```

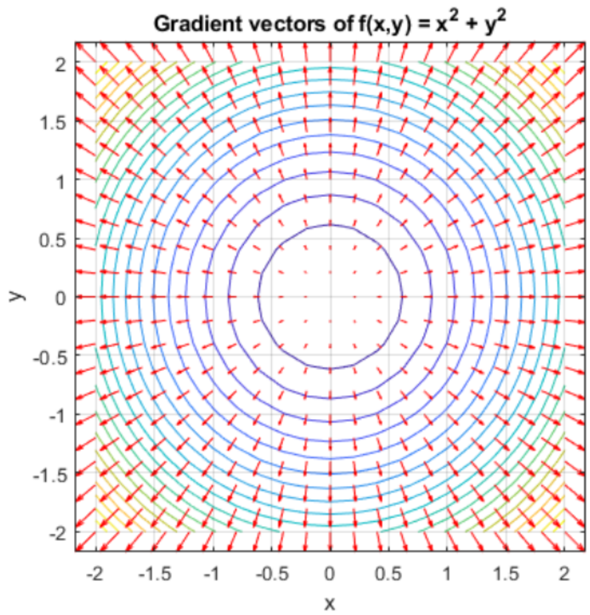

Figure 2: Visualization of the gradient of a scalar field in 2D

3D Visualization:

```
% Define grid
[x, y] = meshgrid(-2:0.4:2, -2:0.4:2);

% Define scalar function
f = x.^2 + y.^2;

% Compute gradients
[fx, fy] = gradient(f, 0.4, 0.4);

% Plot 3D surface
figure;
surf(x, y, f);
shading interp
colormap jet
hold on;

% Plot gradient vectors
quiver3(x, y, f, fx, fy, zeros(size(f)), 0.8, 'k');  % Arrows lie
    on surface

title('3D Surface and Gradient Vectors of f(x, y) = x^2 + y^2');
xlabel('x'); ylabel('y'); zlabel('f(x,y)');
axis tight;
grid on;
view(45, 30);
```

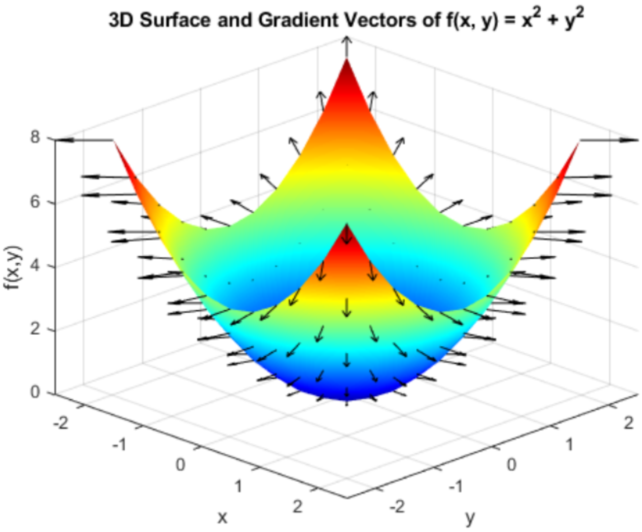

Figure 3: Visualization of the gradient of a scalar field in 3D

References

1 The code used to generate plots is in the ME21BTECH11001.m file.