<html>

<head>

<title>function</title>

<style>

#container {

background: #fff;

width: 600px;

height: 600px;

}

</style>

<meta charset=utf-8 />

<style id="jsbin-css"></style>

</head>

<body>

<script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/109/three.min.js"></script>

<script src="https://unpkg.com/three@0.85.0/examples/js/controls/OrbitControls.js"></script>

<script src="https://unpkg.com/three@0.85.0/examples/js/controls/TransformControls.js"></script>

<script type="text/javascript">

var scene = new THREE.Scene();

var scene = new THREE.Scene();

//setting the background

scene.background = new THREE.Color(0xf0f0ff);

var camera = new THREE.PerspectiveCamera(60, window.innerWidth / window.innerHeight, 1, 1000);

camera.position.set(-4, 2, 5);

var renderer = new THREE.WebGLRenderer({

antialias: true

});

renderer.setSize(window.innerWidth, window.innerHeight);

document.body.appendChild(renderer.domElement);

var controls = new THREE.OrbitControls(camera, renderer.domElement);

//now creating the plane

var planeGeometry = new THREE.PlaneBufferGeometry(50, 50, 50);

var planeMaterial = new THREE.MeshStandardMaterial({color: 0x4E4E62, side: THREE.DoubleSide, wireframe: true});

var plane = new THREE.Mesh(planeGeometry, planeMaterial);

plane.receiveShadow = true;

plane.rotation.x = 0.5 \* Math.PI;

scene.add(plane);

//add coloured directional lights to give colourful effects to the surface

//a combination of red, blue ,green and mixed lights are used

var light = new THREE.DirectionalLight( 0xff0000 );

light.position.set( 10, 10, 10 );

scene.add( light );

light = new THREE.DirectionalLight( 0x00ff00 );

light.position.set( 10, -10, 10 );

scene.add( light );

light = new THREE.DirectionalLight( 0x0000ff );

light.position.set( -10, -10, 10 );

scene.add( light );

light = new THREE.DirectionalLight( 0x555555 );

light.position.set( 0, 0, -10 );

scene.add( light );

//add a mild ambient light

light = new THREE.AmbientLight( 0x222222 );

scene.add( light );

var planeGeom = new THREE.PlaneBufferGeometry(4, 4, 20, 20);

planeGeom.rotateX(-Math.PI \* 0.5);

var v = new THREE.Vector3();

var positions = planeGeom.attributes.position;

for (var i = 0; i < positions.count; i++) {

v.fromBufferAttribute(positions, i);

//here i use the function (-(x^2 - (2y^2)) \* 0.25)

// here x = v.x and y = v.z

//you can use any other function and the graph will change accordingly

positions.setY(i, (-(v.x \* v.x) - (2 \* v.z \* v.z)) \* 0.25);

}

planeGeom.center();

planeGeom.computeVertexNormals();

var ellipticParaboloidSurface = new THREE.Mesh(planeGeom, new THREE.MeshNormalMaterial({

side: THREE.DoubleSide

}));

scene.add(ellipticParaboloidSurface);

contrls = new THREE.TransformControls(camera,renderer.domElement);

contrls.attach(ellipticParaboloidSurface);

scene.add(contrls)

renderer.setAnimationLoop(() => {

renderer.render(scene, camera);

})

function animate() {

camc.update();

renderer.render(scene, camera);

requestAnimationFrame(animate);

}

animate();

</script>

</body>

</html>