Gradient Descent in Three Dimensions: Generalization to $f(x, y, z)$

Group 2 – Ferschl Martin, Reiter Roman, Zenkic Mirza

December 12, 2025

1 Task specification

We generalize the setting of Task 4 to a function of three variables

$$f : \mathbb{R}^3 \to \mathbb{R}, \qquad f(x, y, z),$$

and consider the simplest convex example

$$f(x, y, z) = x^2 + y^2 + z^2.$$

This function has a unique global minimum at $(x, y, z) = (0, 0, 0)$.

The goal is to:

- describe the geometry of the level sets of f and the associated gradient vector,
- formulate and justify the gradient descent iteration in $\mathbb{R}^3$ for this model problem,
- (optionally) implement and test the iteration numerically.

2 Geometry: level sets and gradient in $\mathbb{R}^3$

For the function

$$f(x, y, z) = x^2 + y^2 + z^2,$$

the level sets are given by

$$f(x, y, z) = c \quad \Longleftrightarrow \quad x^2 + y^2 + z^2 = c,$$

for constants $c \geq 0$. These are spheres in $\mathbb{R}^3$ with centre at the origin and radius $\sqrt{c}$.

The partial derivatives are

$$f_x(x, y, z) = 2x, \qquad f_y(x, y, z) = 2y, \qquad f_z(x, y, z) = 2z,$$

so the gradient is

$$\nabla f(x, y, z) = \begin{pmatrix} 2x \\ 2y \\ 2z \end{pmatrix} = 2 \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

At a point (x, y, z) on the sphere $x^2 + y^2 + z^2 = c$, the vector $(x, y, z)^\top$ points radially outwards from the origin. Hence $\nabla f(x, y, z) = 2(x, y, z)^\top$ is also radial and orthogonal to the tangent plane of the sphere at that point. Thus, in three dimensions, the same geometric principle holds as in Task 4: the gradient is normal to the level surfaces (here, spheres).

The direction of steepest descent is given by $-\nabla f(x, y, z)$, which points radially inward toward the minimum at $(0, 0, 0)$.

3 Gradient descent iteration in $\mathbb{R}^3$

We now consider the gradient descent iteration

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma \, \nabla f(\boldsymbol{x}_k),$$

where $\boldsymbol{x}_k = (x_k, y_k, z_k)^\top \in \mathbb{R}^3$ and $\gamma > 0$ is a step size.

For our model function, the gradient is

$$\nabla f(x, y, z) = (2x, 2y, 2z)^\top,$$

so the update reads

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma(2x_k, 2y_k, 2z_k)^\top = (1 - 2\gamma) \, \boldsymbol{x}_k.$$

Written componentwise:

$$x_{k+1} = (1 - 2\gamma)x_k, \quad y_{k+1} = (1 - 2\gamma)y_k, \quad z_{k+1} = (1 - 2\gamma)z_k.$$

By induction, we obtain

$$\boldsymbol{x}_k = (1 - 2\gamma)^k \boldsymbol{x}_0.$$

Hence the iterates converge to the minimum $\mathbf{0}$ if and only if

$$|1 - 2\gamma| < 1,$$

i.e.

$$0 < \gamma < 1.$$

As in the two-dimensional case:

- If $0 < \gamma < \frac{1}{2}$, then $1 - 2\gamma \in (0, 1)$. The iterates move monotonically towards the origin without changing direction.
- If $\frac{1}{2} < \gamma < 1$, then $1 - 2\gamma \in (-1, 0)$. The iterates converge to the origin but alternate sign, i.e. they "zigzag" across the origin.
- If $\gamma = 1$, then $1 - 2\gamma = -1$, so $\boldsymbol{x}_k$ alternates between $\boldsymbol{x}_0$ and $-\boldsymbol{x}_0$ and never converges.
- If $\gamma > 1$, then $|1 - 2\gamma| > 1$ and the iterates diverge.

Thus, the convergence condition and the qualitative behaviour of the method are exactly the same as in $\mathbb{R}^2$; only the dimension of the vector changes.

4 Python implementation (optional)

To illustrate the three-dimensional gradient descent, we can generalize the code from Task 4. We now work with vectors in $\mathbb{R}^3$:

```python
import numpy as np

def f3(x):
    """
    Objective function in R^3:
    f(x, y, z) = x^2 + y^2 + z^2
    x is a numpy array of shape (3,).
    """
    return np.dot(x, x)  # x^T x = x^2 + y^2 + z^2

def grad_f3(x):
    """Gradient: f(x) = 2x."""
    return 2.0 * x

def gradient_descent_3d(x0, gamma, n_steps):
    """
    Gradient descent in R^3 starting from x0 (shape (3,)),
    step size gamma, for n_steps iterations.
    """
    x = np.array(x0, dtype=float)
    trajectory = [x.copy()]
    for k in range(n_steps):
        g = grad_f3(x)
        x = x - gamma * g
        trajectory.append(x.copy())
    return trajectory

# Example usage
x0 = np.array([1.0, -1.0, 0.5])
gamma = 0.25
n_steps = 10

traj = gradient_descent_3d(x0, gamma, n_steps)
print("Gradient descent in R^3 with gamma =", gamma)
for k, xk in enumerate(traj):
    print(f"k={k:2d}: x = {xk}, f(x) = {f3(xk): .6e}")
```

For $0 < \gamma < 1$, the sequence $\boldsymbol{x}_k$ converges to the origin and the function values $f(\boldsymbol{x}_k)$ decrease to zero, confirming the theoretical analysis.

5 Conclusion

The generalization of Task 4 from $\mathbb{R}^2$ to $\mathbb{R}^3$ is straightforward:

- For $f(x, y, z) = x^2 + y^2 + z^2$, the level sets are spheres and the gradient $\nabla f = (2x, 2y, 2z)^\top$ points radially outwards, orthogonal to the level surfaces.
- The gradient descent iteration $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma \nabla f(\boldsymbol{x}_k)$ becomes a simple scaling $\boldsymbol{x}_{k+1} = (1 - 2\gamma)\boldsymbol{x}_k$.
- Convergence to the unique minimum $\mathbf{0}$ occurs if and only if $0 < \gamma < 1$, with the same interpretation of step-size choices as in the two-dimensional case.

This simple example illustrates that the core ideas of gradient descent extend naturally from two to higher dimensions.