HUDM 6026

04 – Multivariate Optimization

MLE example (11.12 on p. 339)

Find the MLE of the Gamma distribution

Let $X_1, \ldots, X_n$ be a random sample from $\Gamma(r, \lambda)$ distribution

Then the likelihood function is:

$$L(r, \lambda) = \frac{\lambda^{nr}}{\Gamma(r)^n} \prod_{i=1}^{n} x_i^{r-1} \exp\left(-\lambda \sum_{i=1}^{n} x_i\right)$$

and the log-likelihood is:

$$l(r, \lambda) = nr \log(\lambda) - n \log \Gamma(r) + (r-1) \sum_{i=1}^{n} \log x_i - \lambda \sum_{i=1}^{n} x_i$$

MLE example (cont.)

We need to find the simultaneous solution to:

$$\frac{\partial}{\partial \lambda} l(\lambda, r) = \frac{nr}{\lambda} - \sum_{i=1}^{n} x_i = 0$$

$$\frac{\partial}{\partial r} l(\lambda, r) = n \log(\lambda) - n \frac{\Gamma'(r)}{\Gamma(r)} + \sum_{i=1}^{n} \log(x_i) = 0$$

The first equation implies that:

$$\hat{\lambda} = \frac{\hat{r}}{\bar{x}}$$

and substituting in the second equation we obtain

$$n \log\left(\frac{\hat{r}}{\bar{x}}\right) - n \frac{\Gamma'(\hat{r})}{\Gamma(\hat{r})} + \sum_{i=1}^{n} \log(x_i) = 0$$

The Gradient

Definition of *gradient* for a real-valued function of:

one variable: $$\nabla f_1(x_1) = \left(\frac{df}{dx_1} \right)$$

two variables: $$\nabla f_2(x_1, x_2) = \left(\frac{df}{dx_1}, \frac{df}{dx_2} \right)$$

three variables: $$\nabla f_3(x_1, x_2, x_3) = \left(\frac{df}{dx_1}, \frac{df}{dx_2}, \frac{df}{dx_3} \right)$$

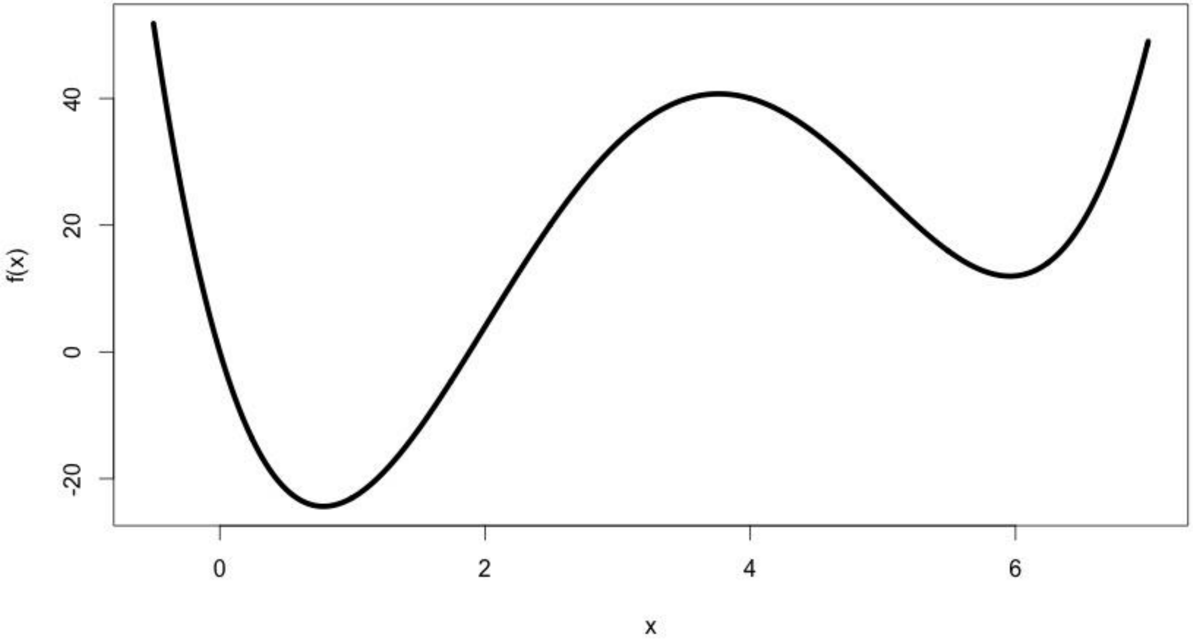
k variables: $$\nabla f_k(x_1, x_2, \ldots, x_k) = \left(\frac{df}{dx_1}, \frac{df}{dx_2}, \ldots, \frac{df}{dx_k} \right)$$

The Gradient

The gradient at x_0 represents the direction in the domain along which the function f increases the most rapidly at x_0.

The Gradient

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x, \quad \text{where } x \in [-0.5, 7]$$

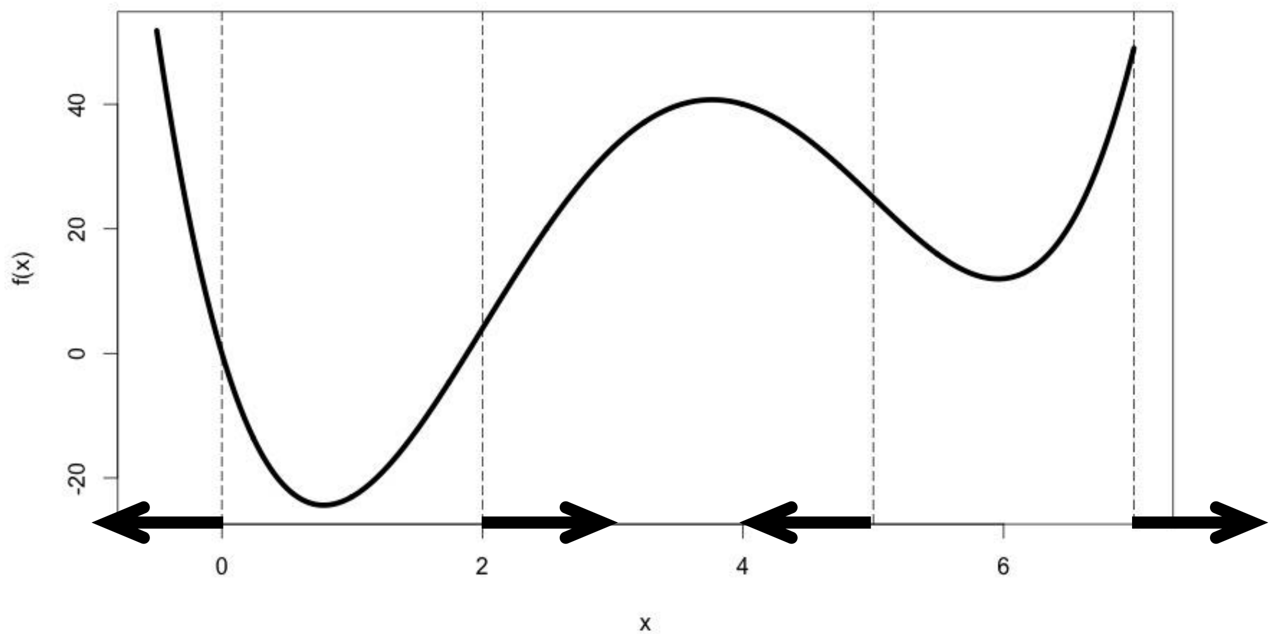


The gradient of f at x_0 is the vector that starts at x_0 and ends at the point $x_0 + f'(x_0)$.

The Gradient – Function of 1 Variable

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

$$f'(x) = 4x^3 - 42x^2 + 120x - 70$$



$$x_0 = 0,2,5,7 \qquad f'(x_0) = -70,34,-20,84$$

Or, expressed as unit vectors from the point of interest, -1, 1, -1, 1

The Gradient – Function of 2 Variables

To find the max of a surface we move in the direction where the surface appears to be increasing most steeply. That is, where it has the largest positive slope.

The gradient vector at (x_0, y_0) tells us the slope of the line tangent to the surface $f(x, y)$ at (x_0, y_0) in the x-direction and y-direction.

Examples:

$$\nabla f(x_0, y_0) = (5,5)$$
$$\nabla f(x_0, y_0) = (1,3)$$
$$\nabla f(x_0, y_0) = (-5,5)$$

The Gradient – Function of 2 Variables

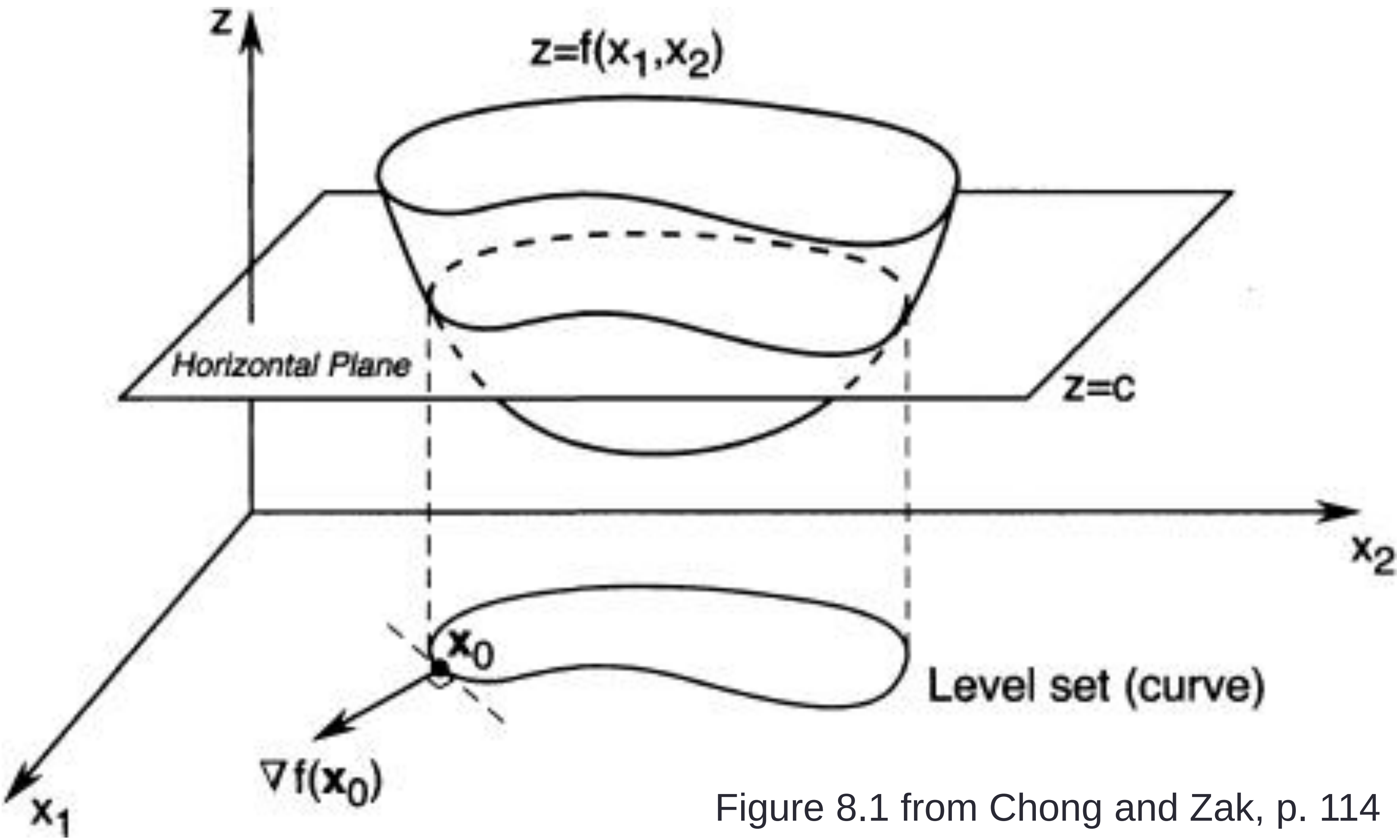


Figure 8.1 from Chong and Zak, p. 114

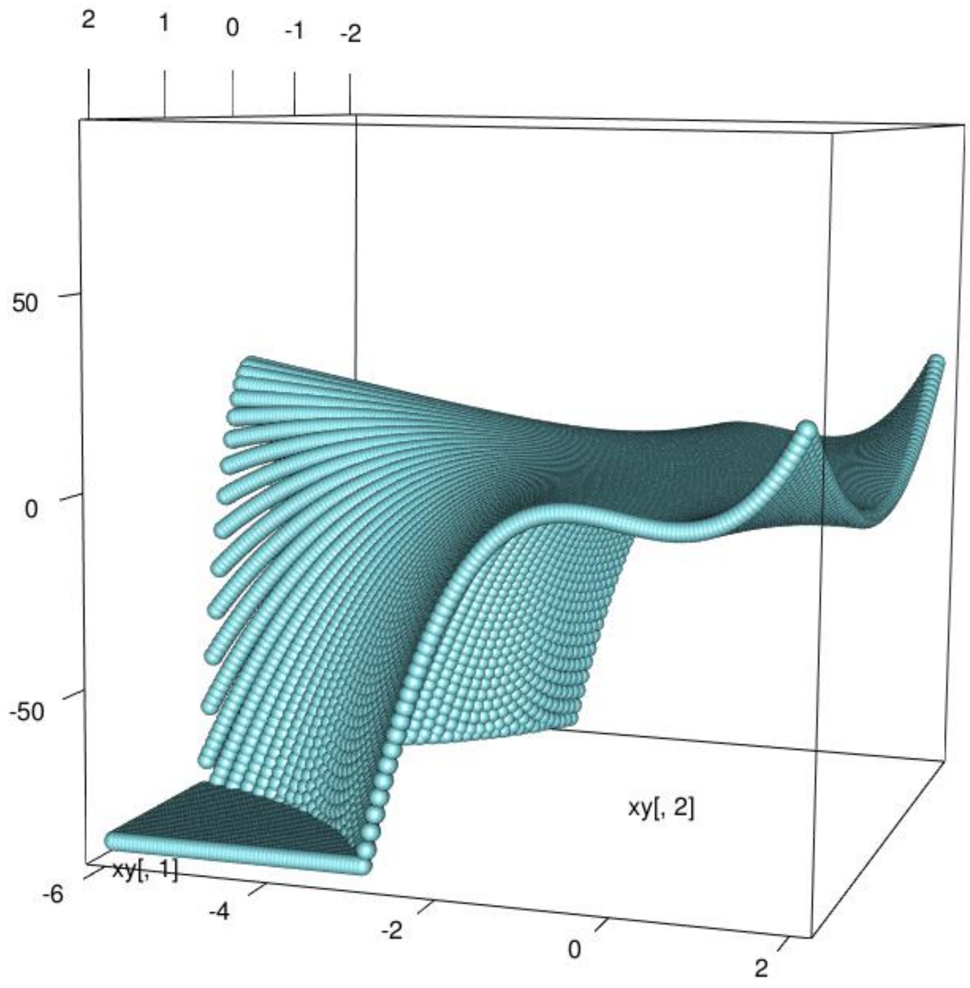
The Gradient – Function of 2 Variables

Let $f(x, y) = x^2 y^3 - 4y$

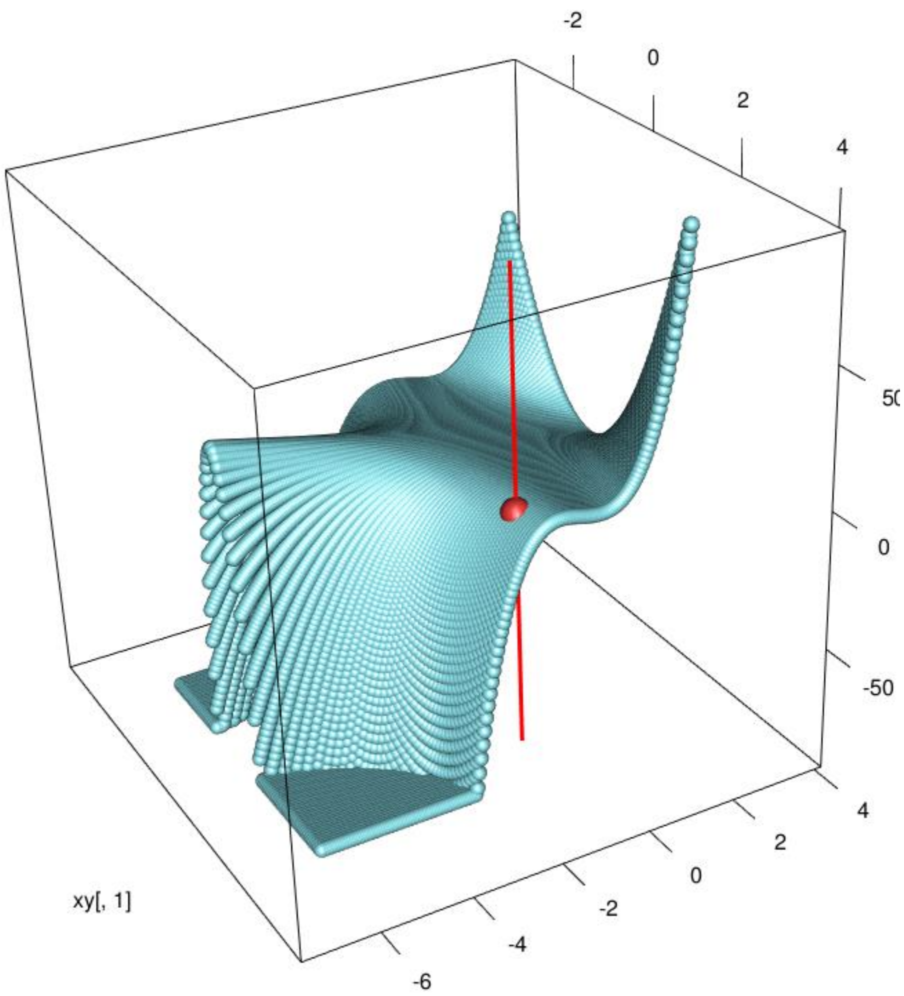
Question 1: $\nabla f(2,-1) = ?$

Question 2: how far should we move in that direction? Call the distance a 'step size' and denote it by alpha (α).

The Gradient – Function of 2 Variables

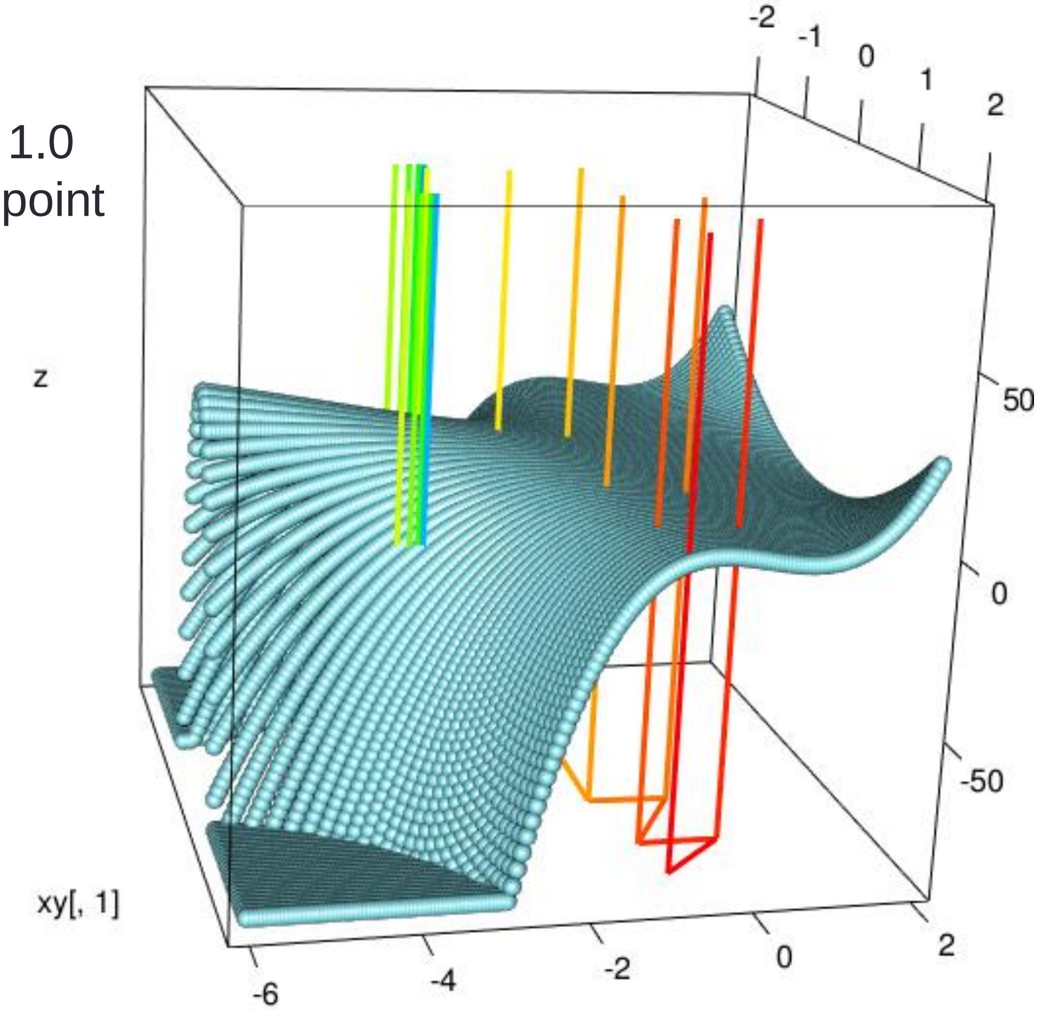


The Gradient – Function of 2 Variables



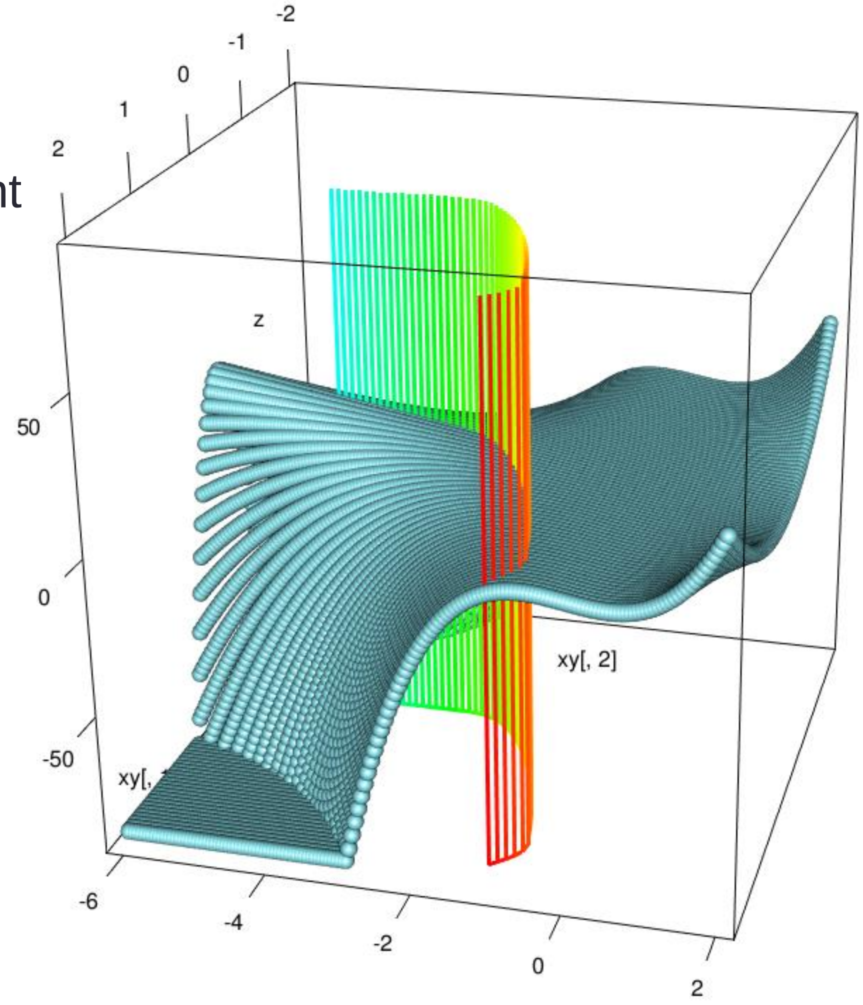
The Gradient – Function of 2 Variables

8 iterations
with alpha = 1.0
and starting point
(2, -1).



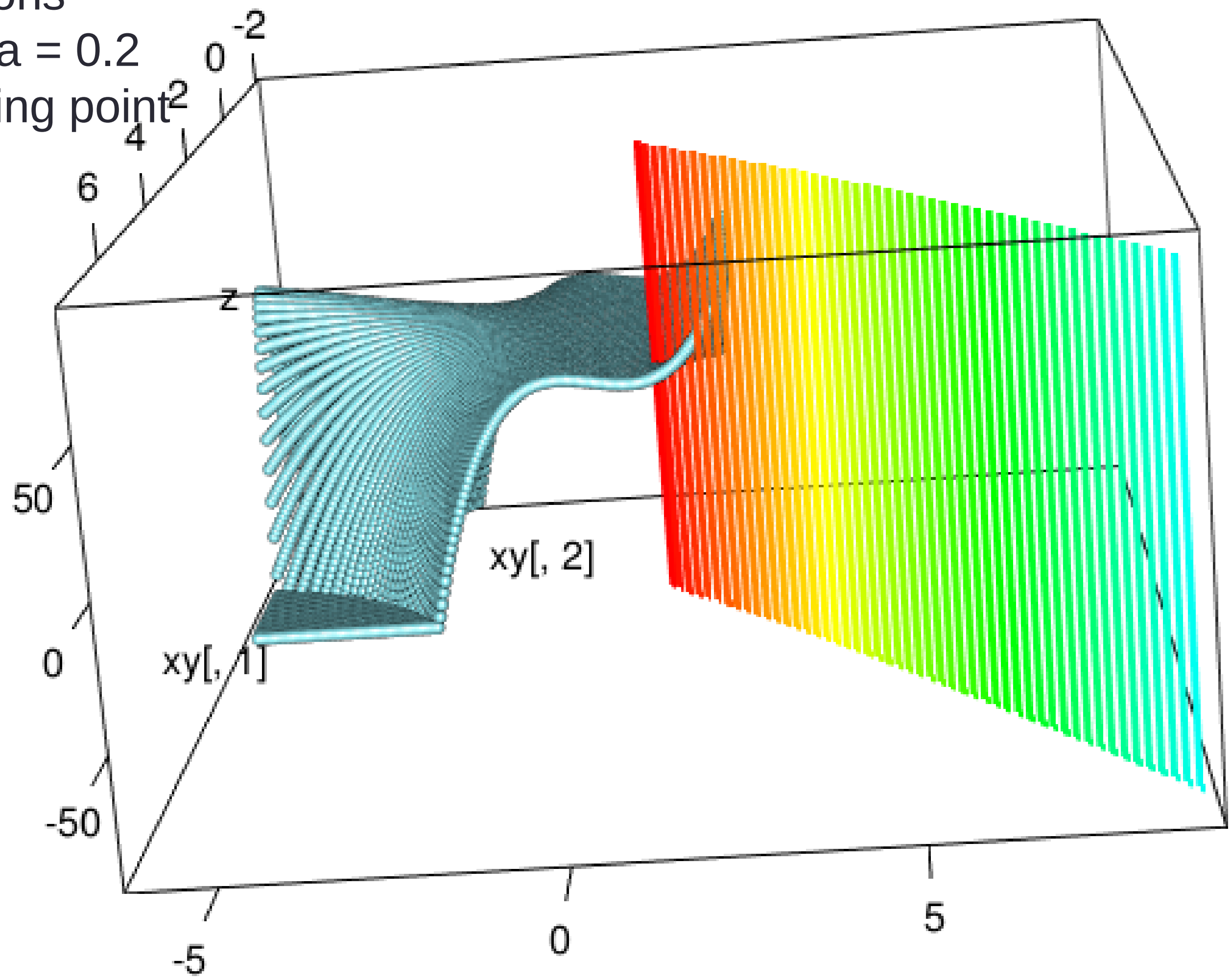
The Gradient – Function of 2 Variables

50 iterations
with alpha = 0.1
and starting point
(2, -1).



The Gradient – Function of 2 Variables

50 iterations
with alpha = 0.2
and starting point
(1.5, 1).



Gradient Descent (or Ascent)

The algorithm, then, consists of moving the **x** vector along the direction of the gradient, by a step-size alpha.

So, for ascent: $$x_{k+1} = x_k + \alpha \cdot \nabla f\left(x_k\right)$$

and, for descent: $$x_{k+1} = x_k - \alpha \cdot \nabla f\left(x_k\right)$$

Steepest Descent

The step size α can be set equal to a constant for all iterations, or it may be allowed to vary iteration by iteration.

Definition:

If we choose the step size so that it maximizes the decrease (for minimization) in *f* at each step, the algorithm is called *steepest descent*.

That is, at each iteration *k*,

$$\alpha_k = \underset{\alpha \geq 0}{\operatorname{argmin}}\ f\left(x_k - \alpha \cdot \nabla f\left(x_k\right)\right)$$

Steepest Descent

$$g(\alpha) = f\left(x_k - \alpha \cdot \nabla f\left(x_k\right)\right)$$

The problem of finding the best step size, alpha, is a univariate optimization problem.

Thus, we may use one of our univariate optimizers such as the secant method or the golden section method.