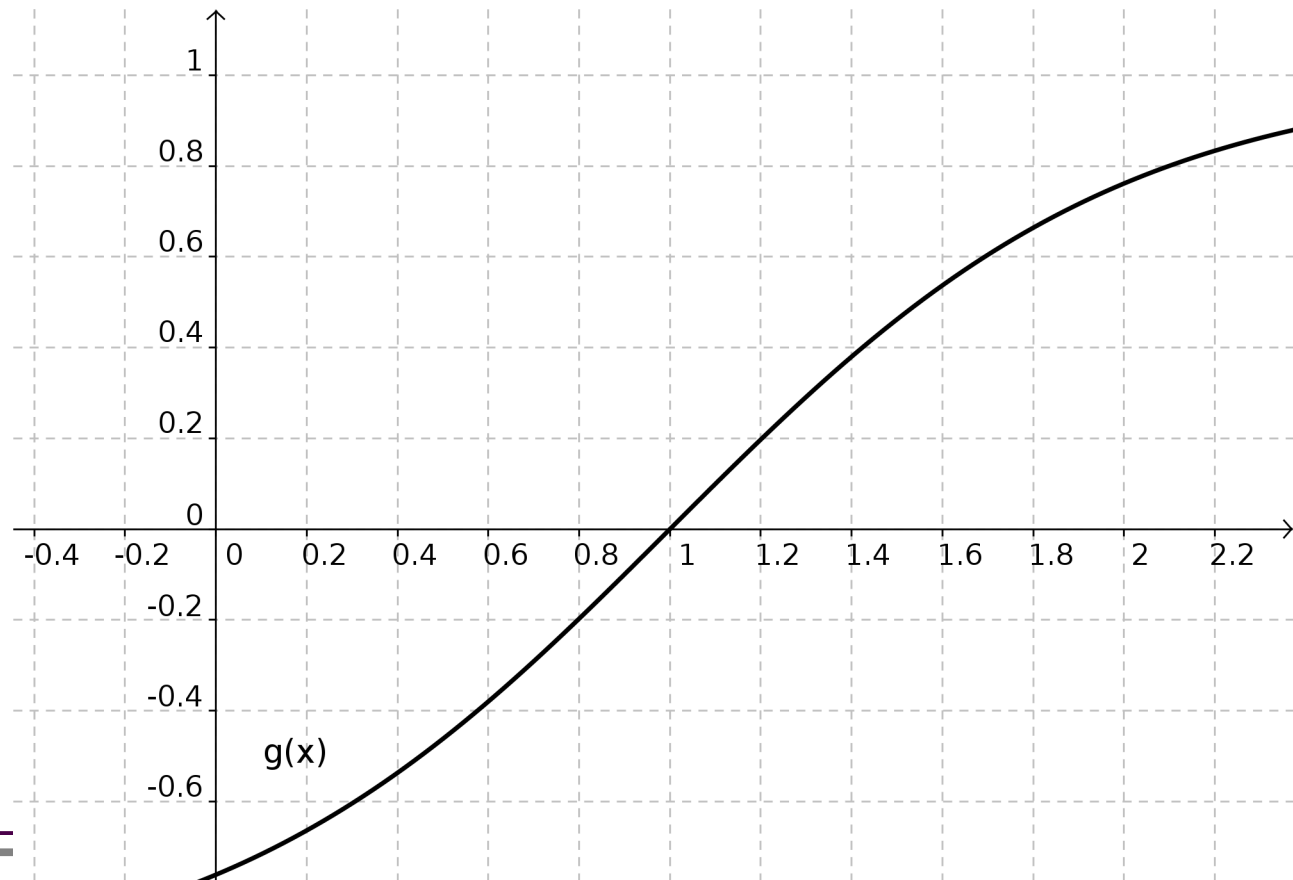


rootfinding

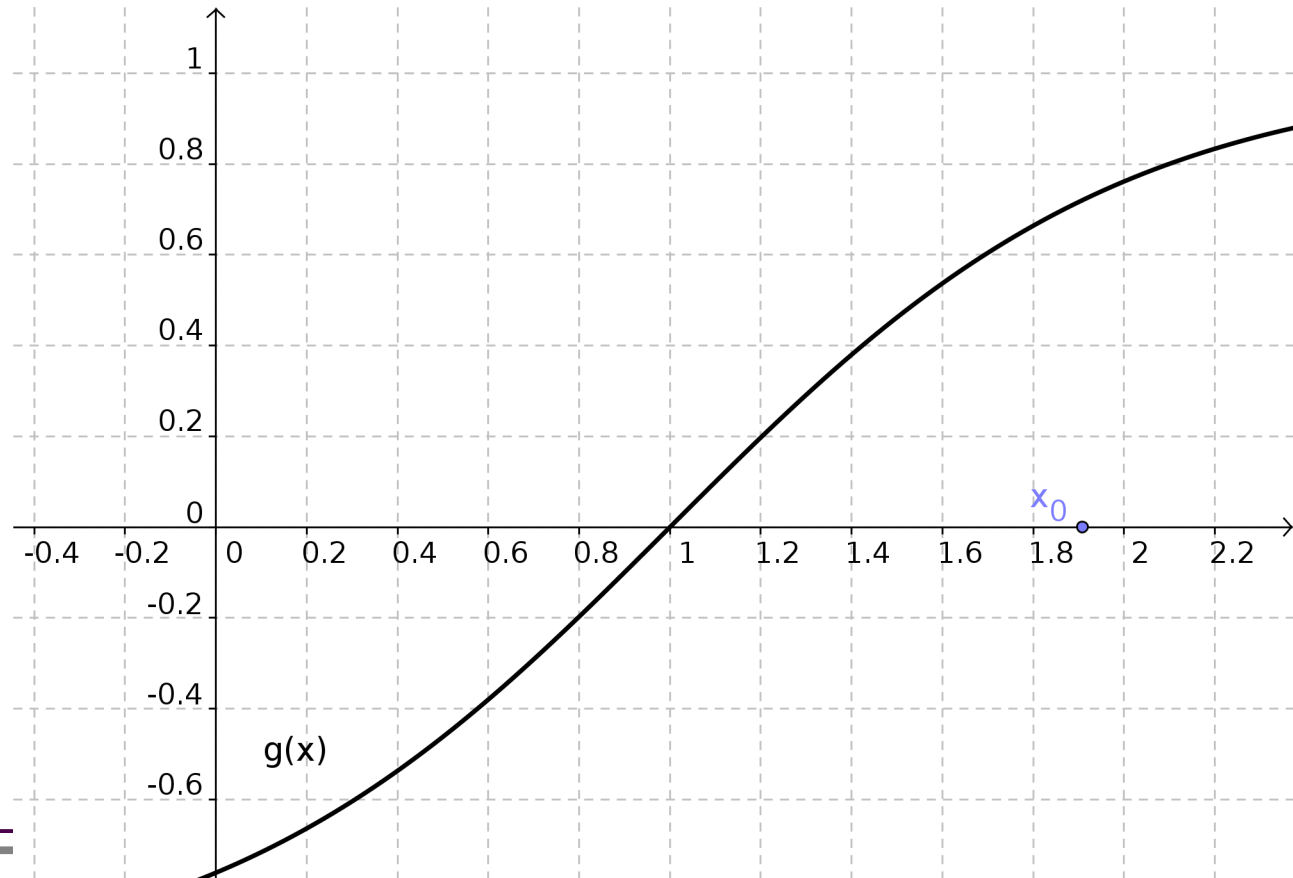
Newton's method (1D)

- Find x such that $g(x)=0$
- $x \in \mathbb{R}, g: \mathbb{R} \rightarrow \mathbb{R}$



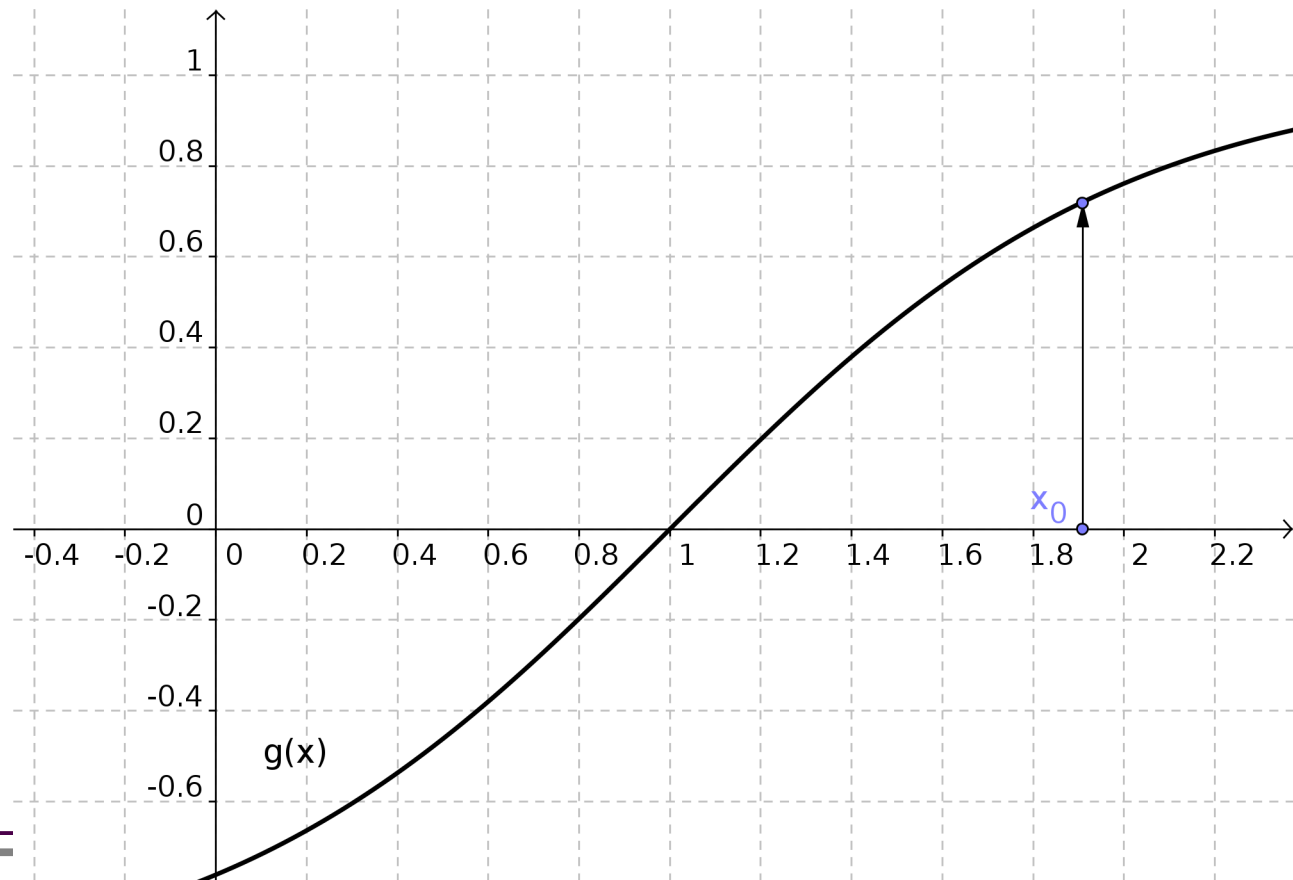
Newton's method (1D)

- Iterative process: start with a guess for x



Newton's method (1D)

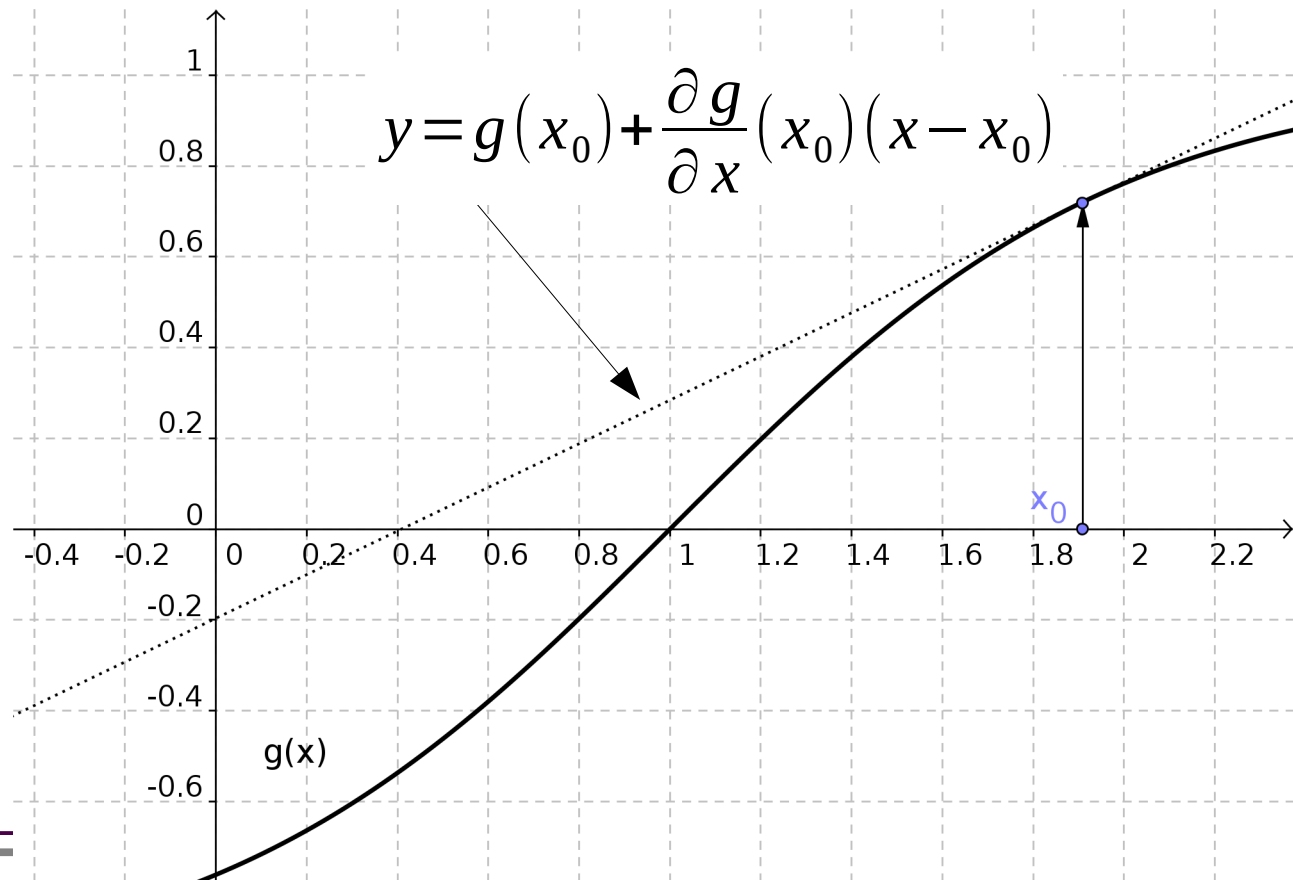
- Evaluate $g(x_0)$



Newton's method (1D)

- Compute tangent approximation

$$g(x) \approx g(x_0) + \frac{\partial g}{\partial x}(x_0)(x - x_0)$$

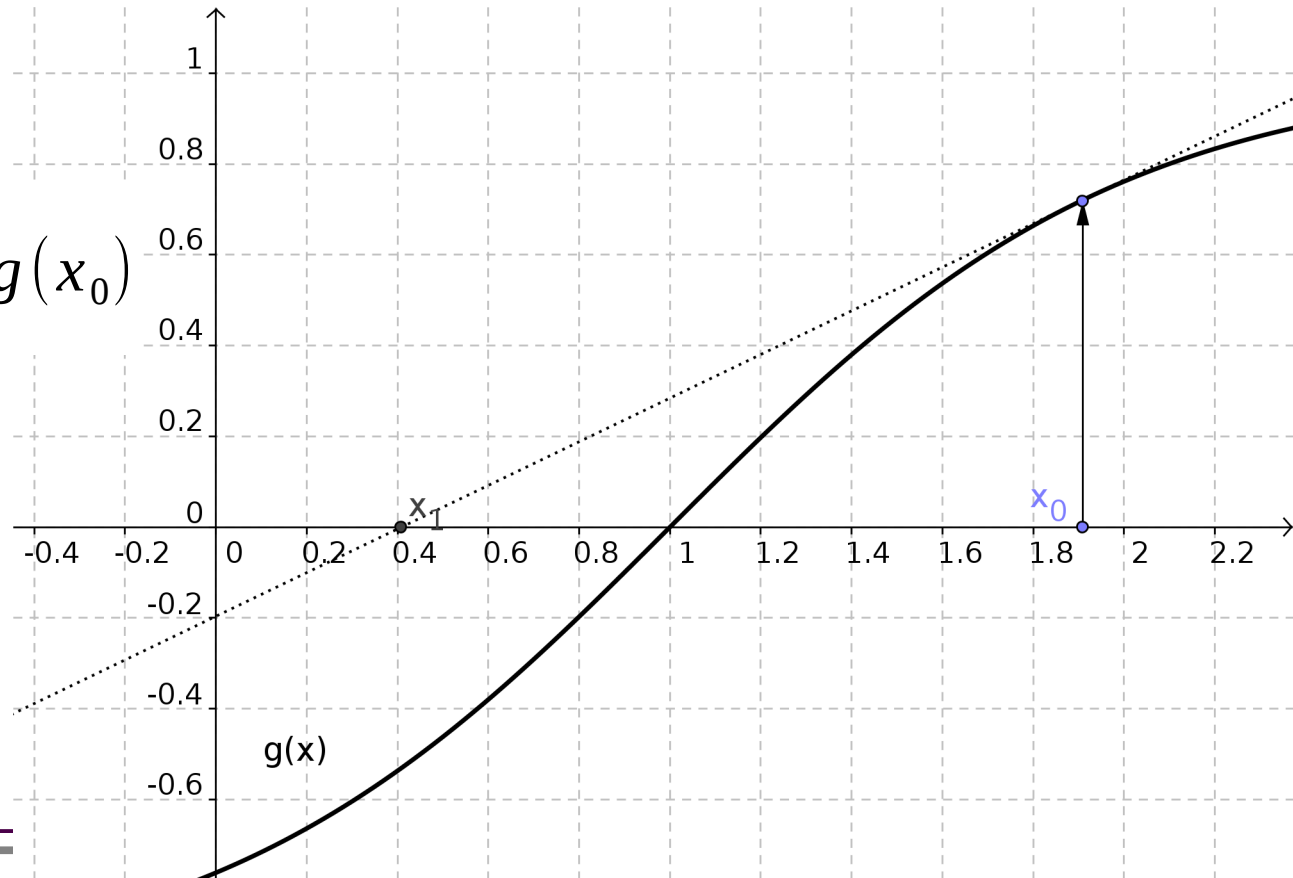


Newton's method (1D)

- Find x such that $g(x)=0$

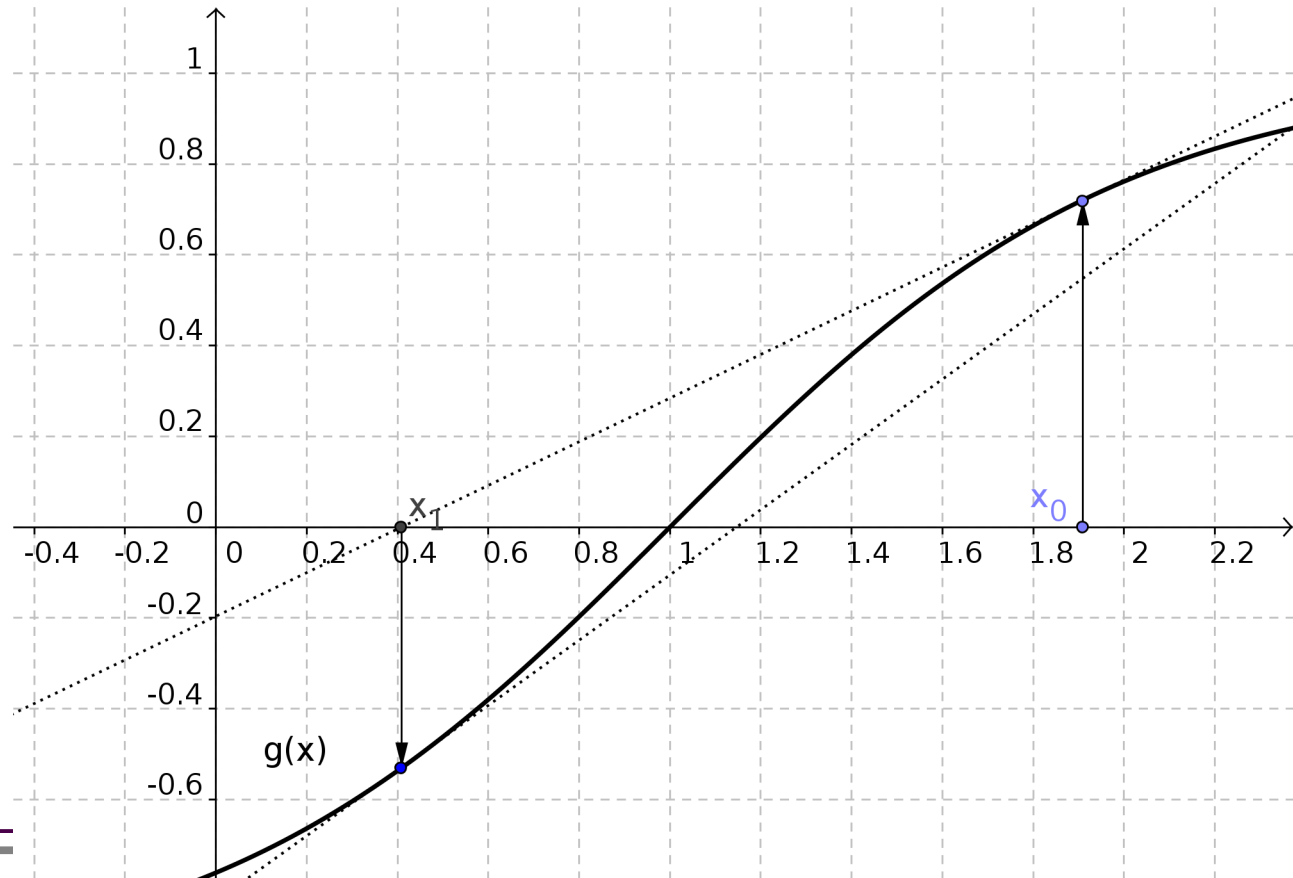
$$g(x_0) + \frac{\partial g}{\partial x}(x_0)(x - x_0) = 0$$

$$\Delta x = - \left(\frac{\partial g}{\partial x}(x_0) \right)^{-1} g(x_0)$$



Newton's method (1D)

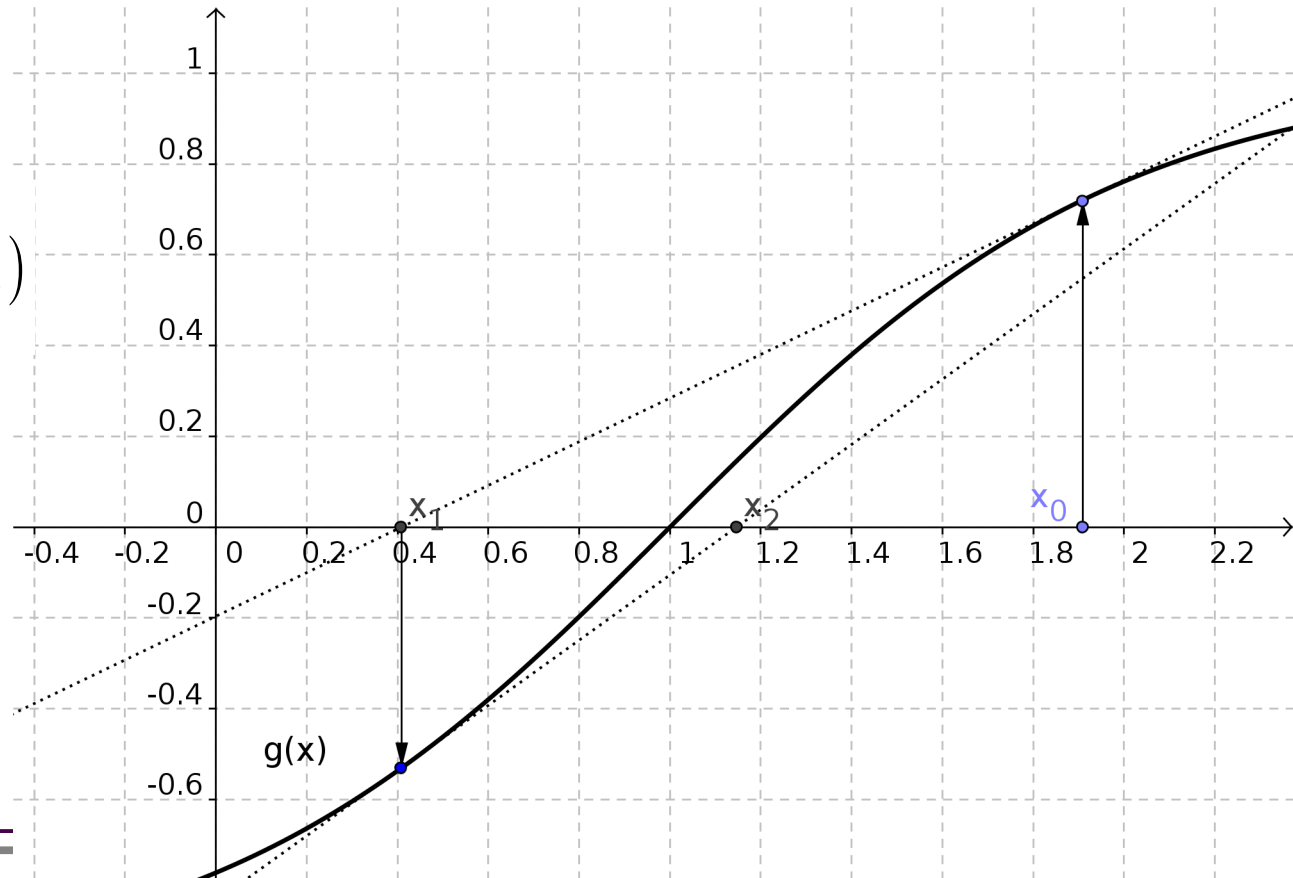
- Start again with new guess



Newton's method (1D)

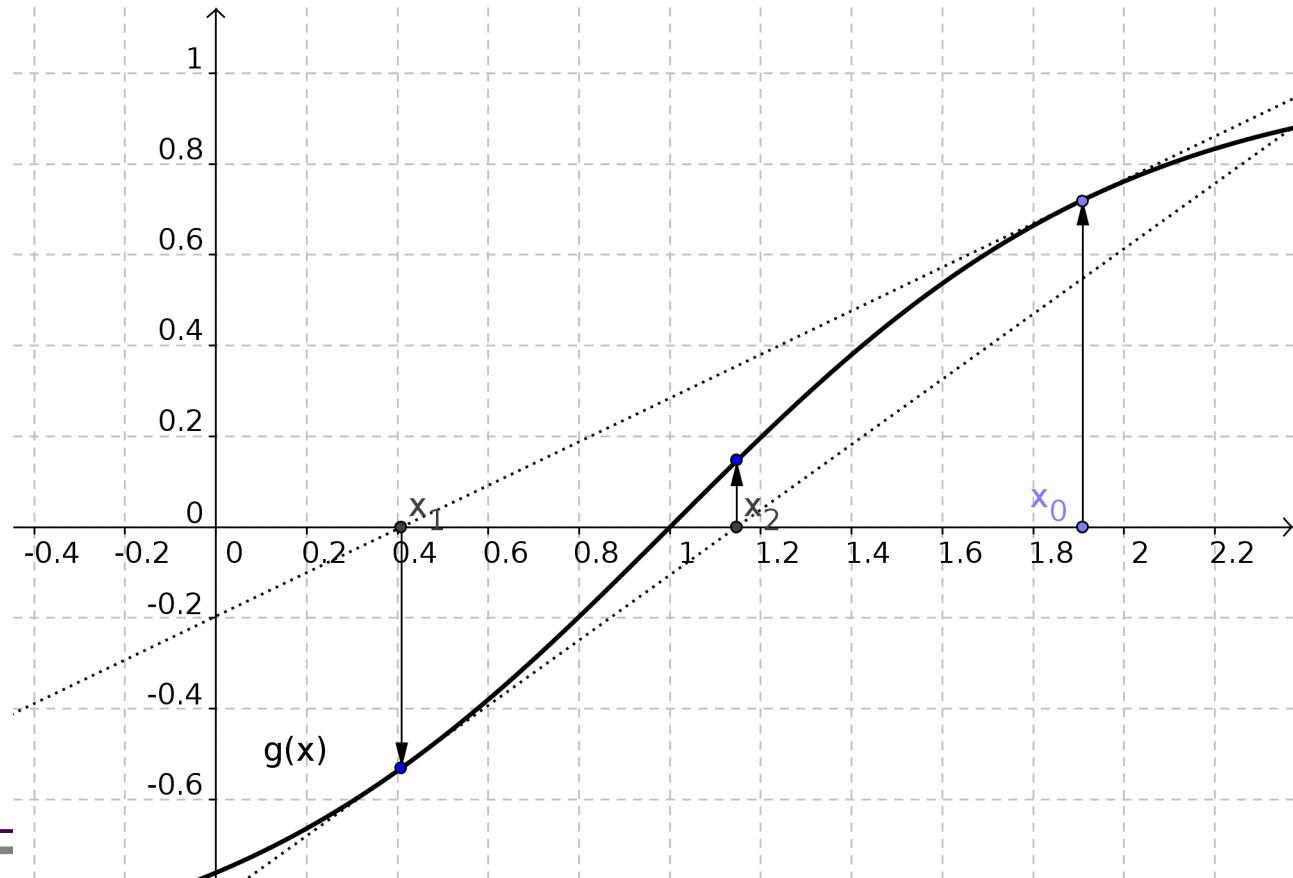
- Take a newton step

$$\Delta x = -\left(\frac{\partial g}{\partial x}(x_k)\right)^{-1} g(x_k)$$



Newton's method (1D)

- Repeat until $\|g(x)\| \leq tol$ $\|\Delta x\| \leq tol_{step}$

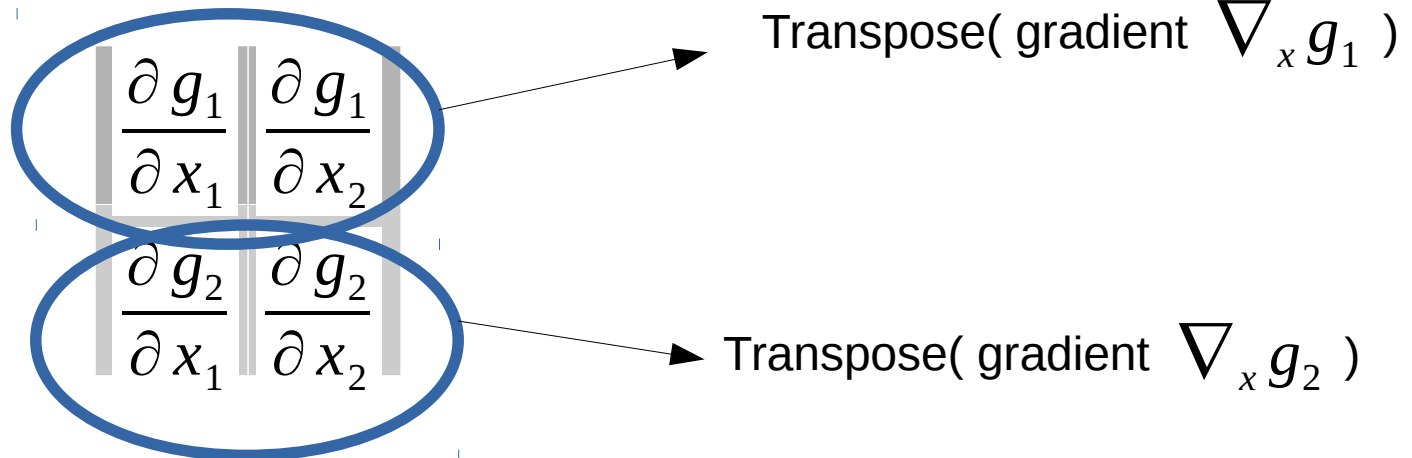


Newton's method

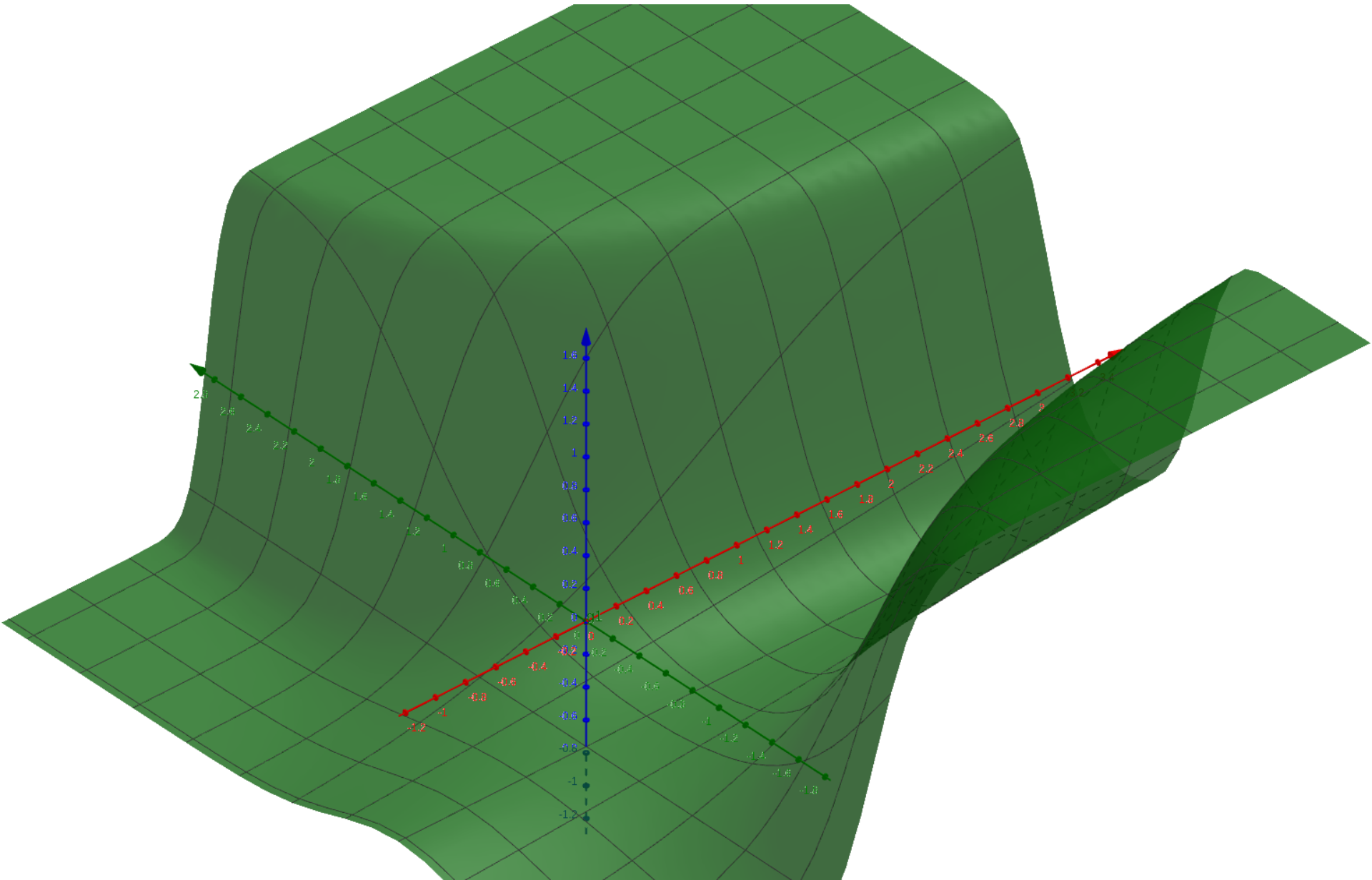
- This trick generalizes to more dimensions
- Find x such that $g(x)=0$
- $x \in \mathbb{R}^2, g: \mathbb{R}^2 \rightarrow \mathbb{R}^2$

$$g(x^0) + \frac{\partial g}{\partial x}(x^0)(x - x^0) = 0$$

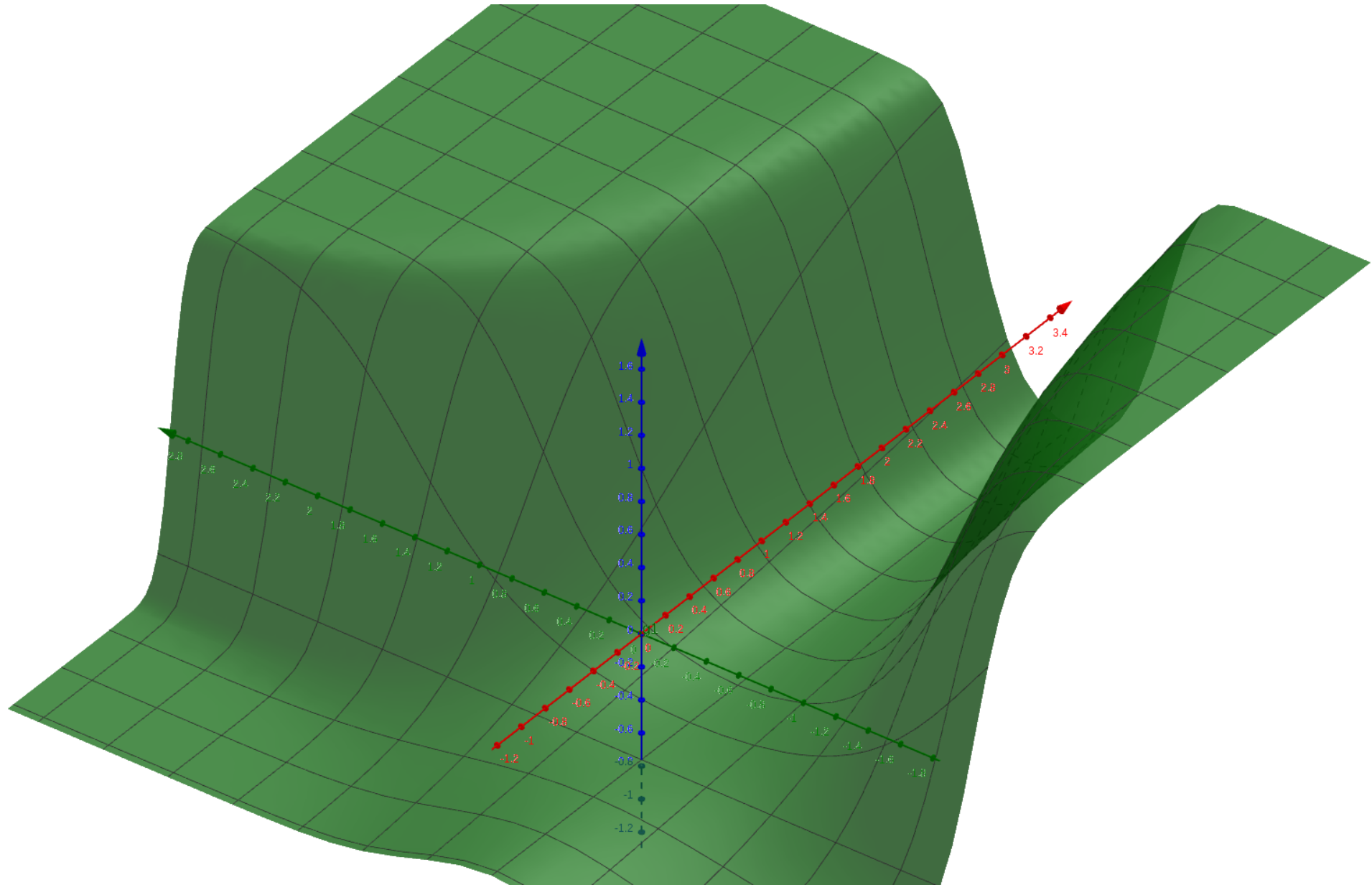
Jacobian $\frac{\partial g}{\partial x}$:



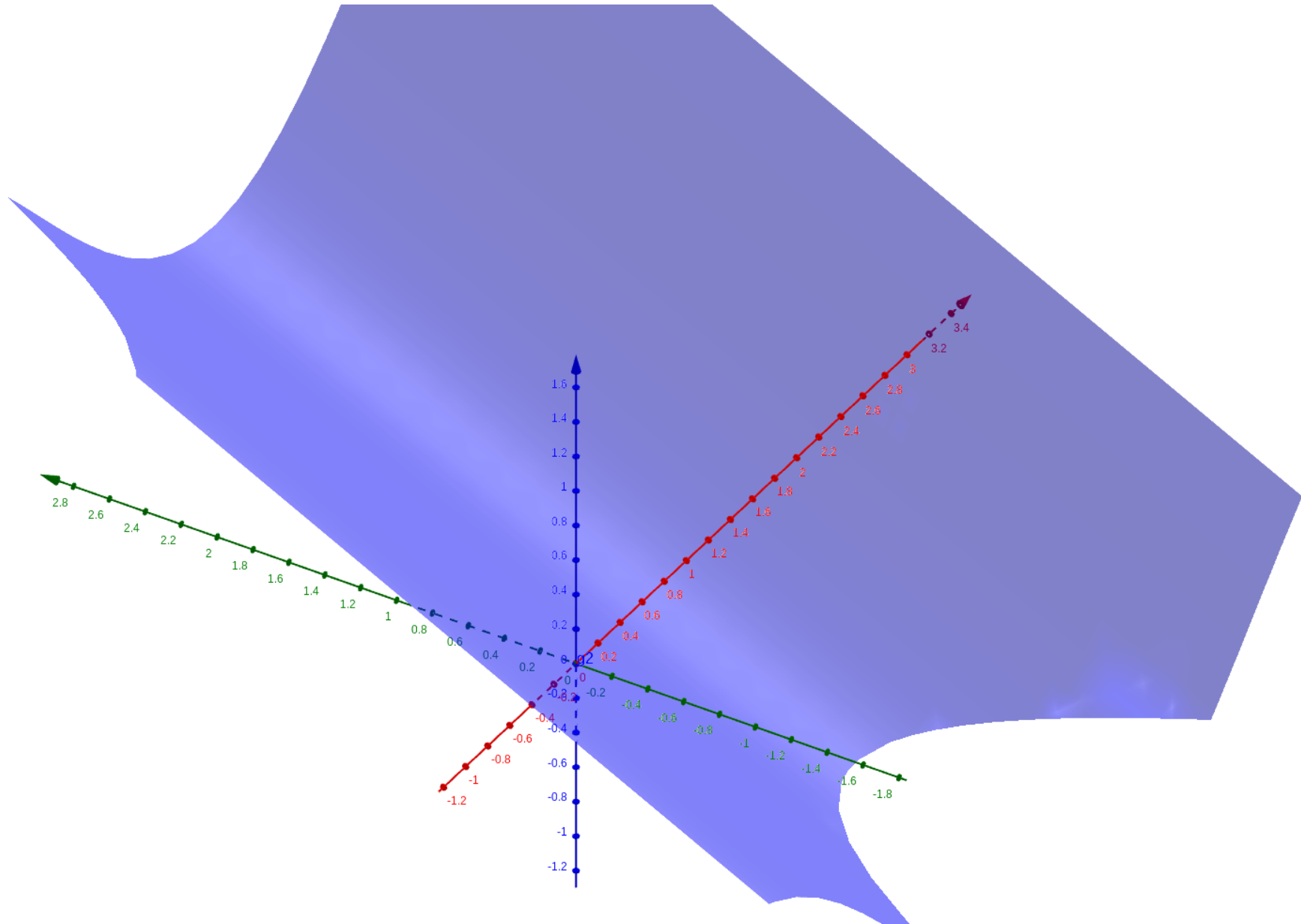
$$g_1(x, y)$$



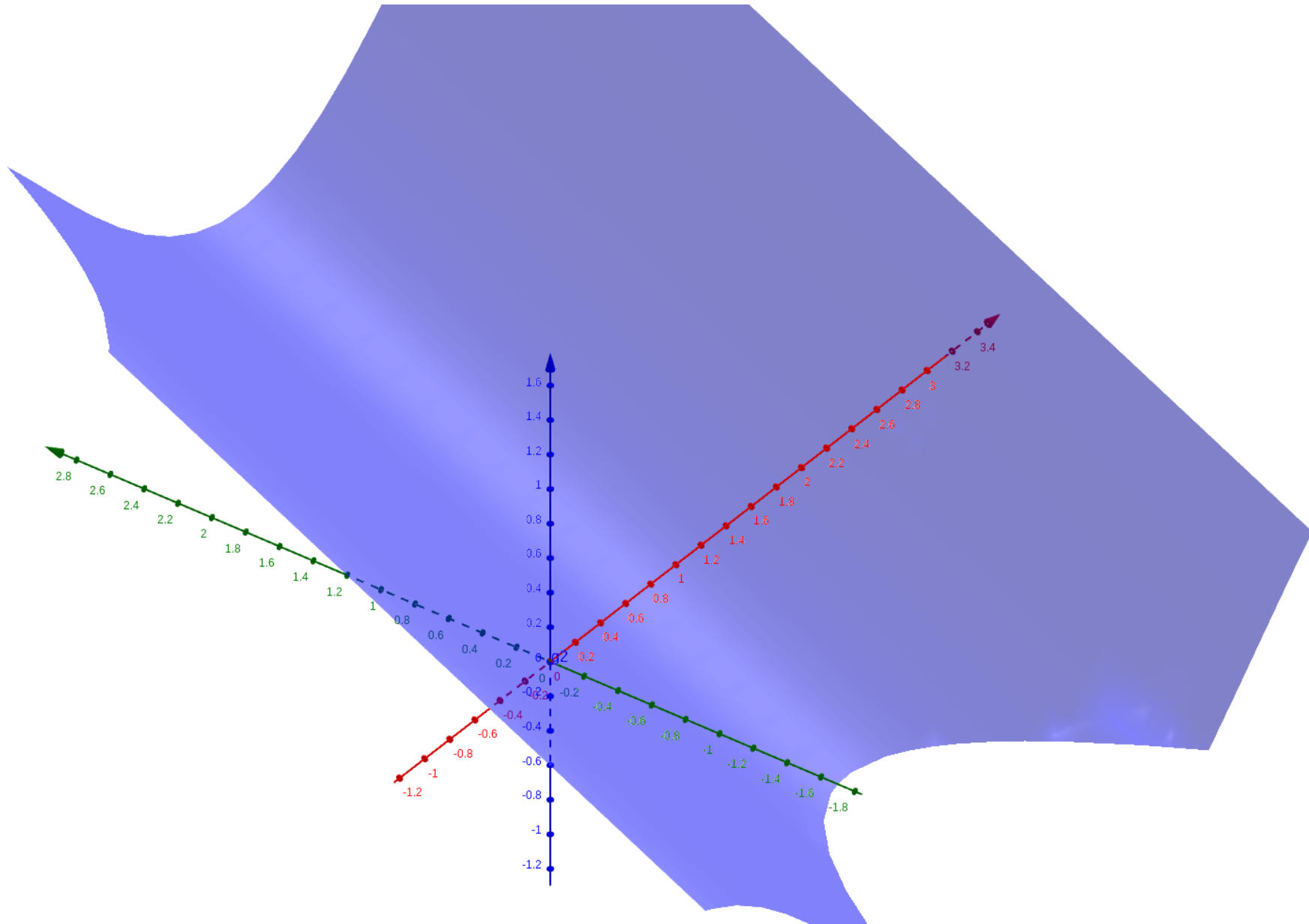
$$g_1(x, y)$$



$$g_2(x, y)$$



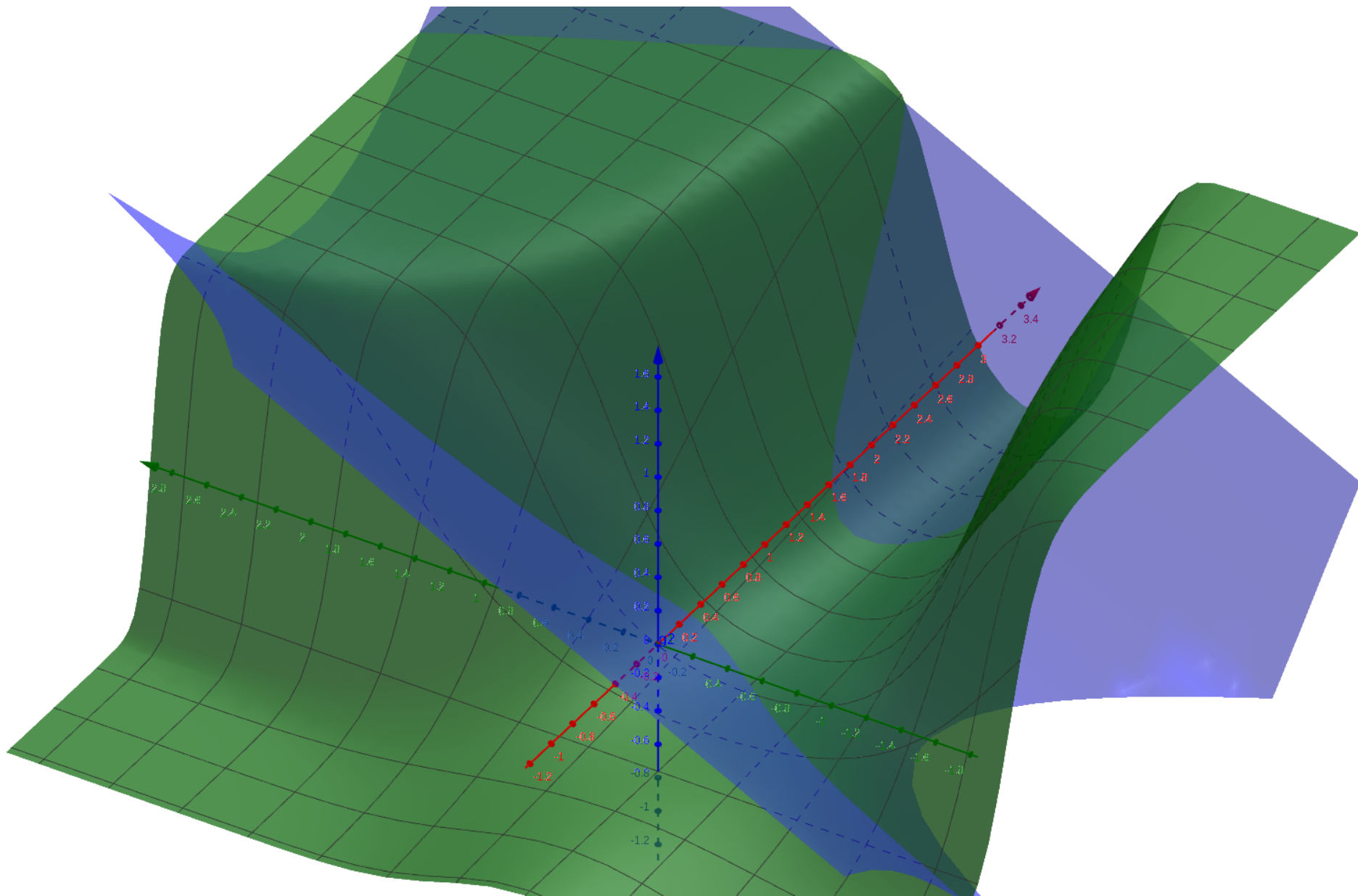
$$g_2(x, y)$$



$$g_1(x, y)$$
$$g_2(x, y)$$

$$g(x, y)$$

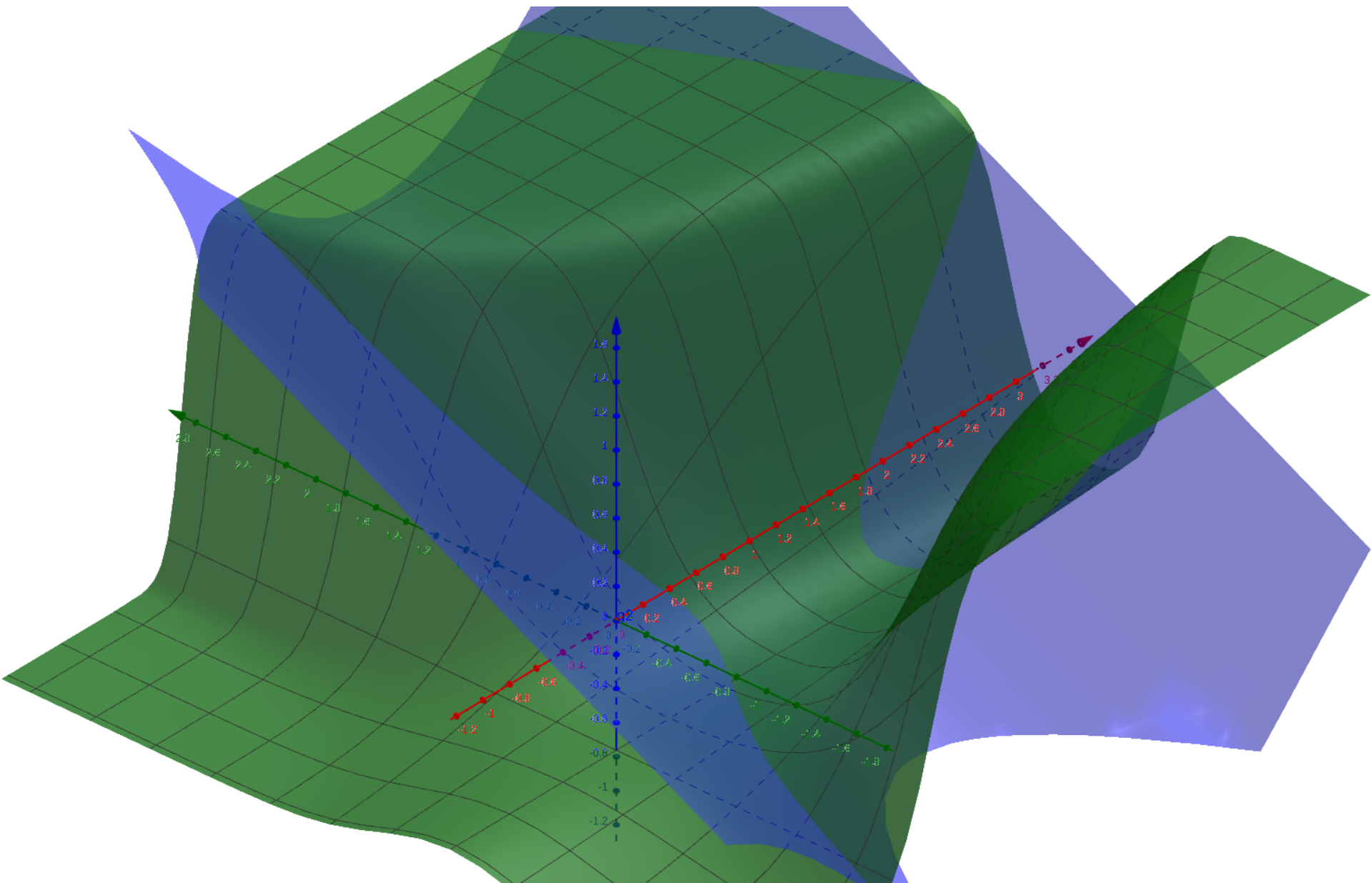
$$g: \mathbb{R}^2 \rightarrow \mathbb{R}^2$$



$$g_1(x, y)$$
$$g_2(x, y)$$

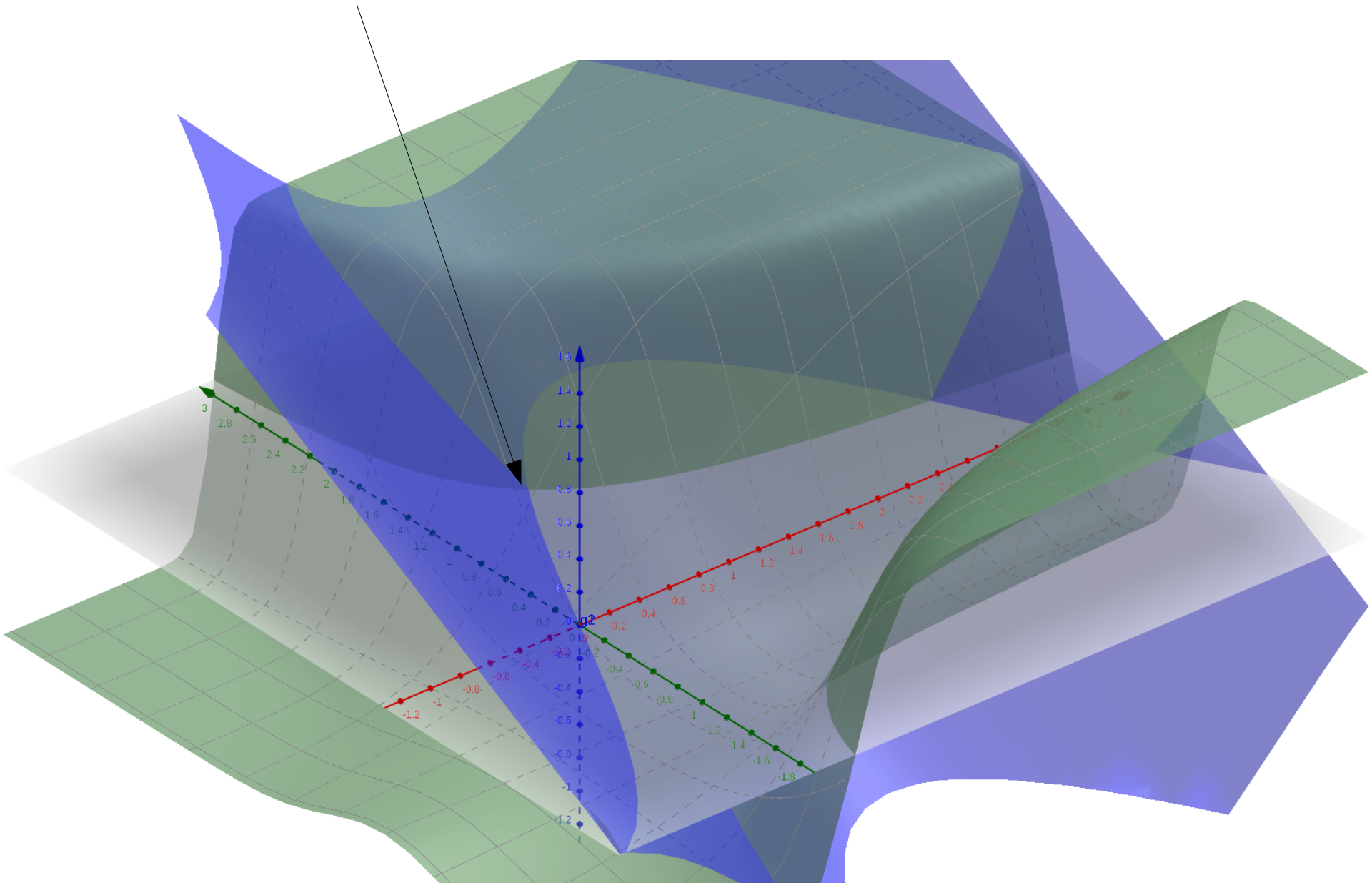
$$g(x, y)$$

$$g: \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

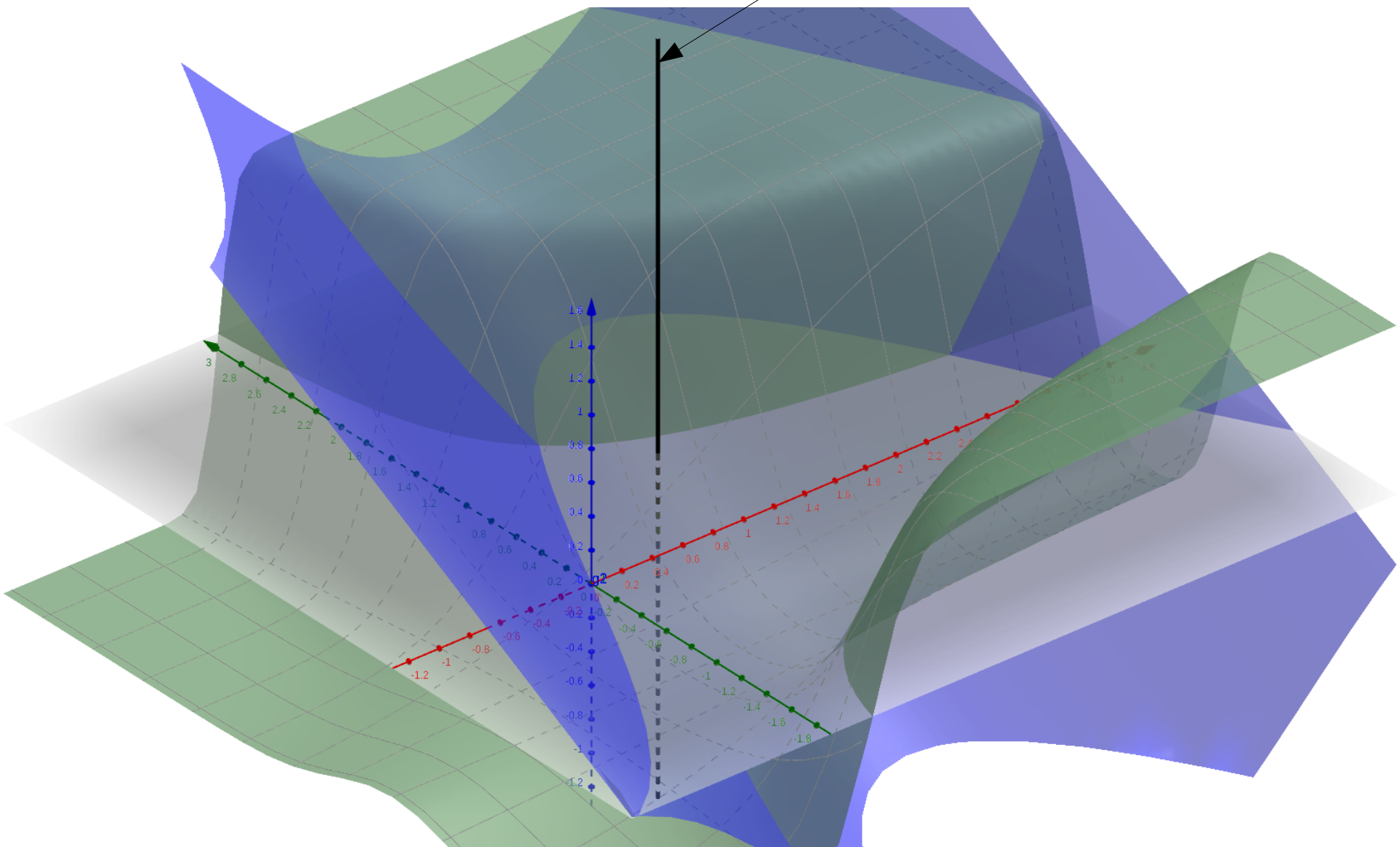


$$g(x, y) = 0$$

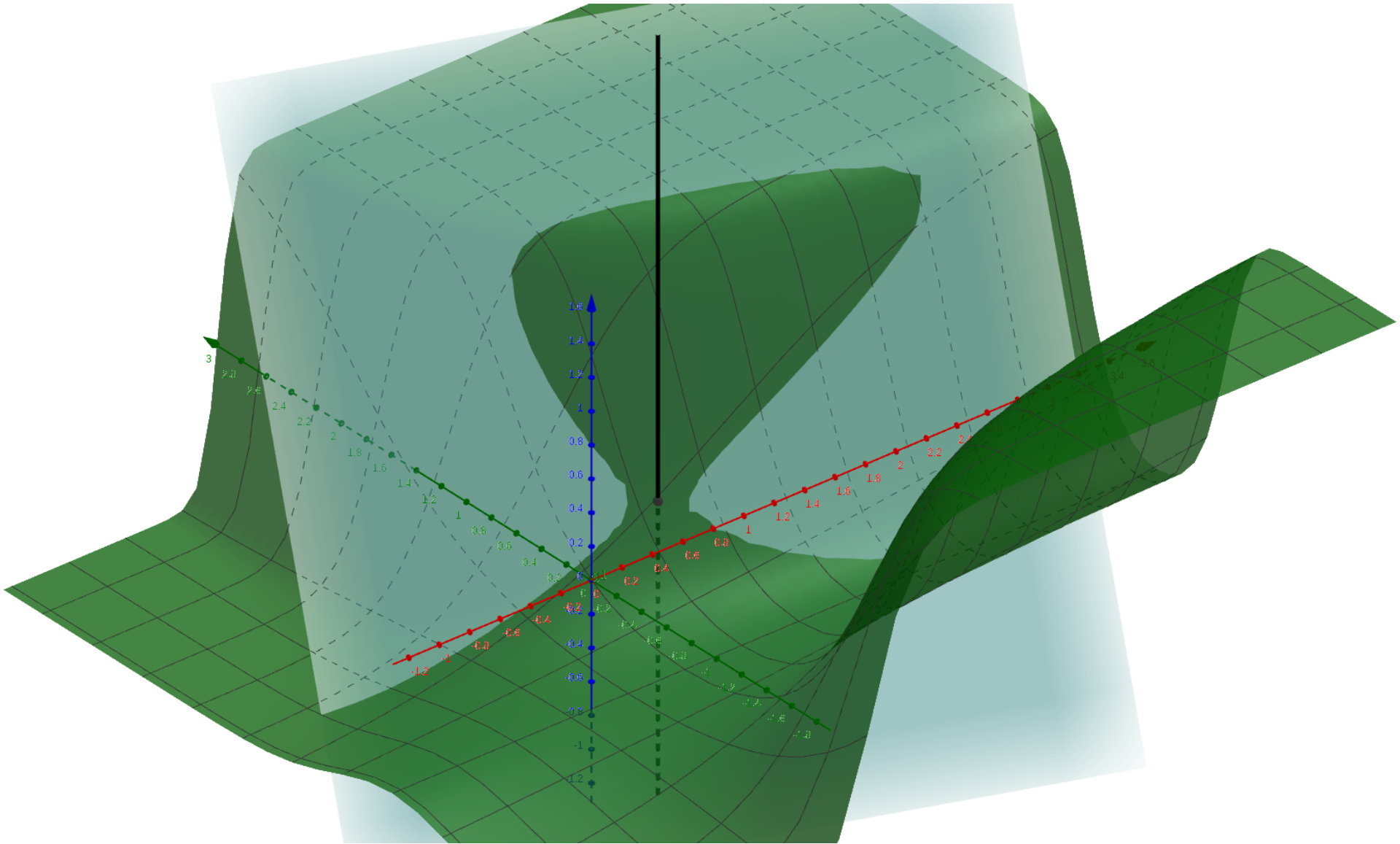
Finding this point



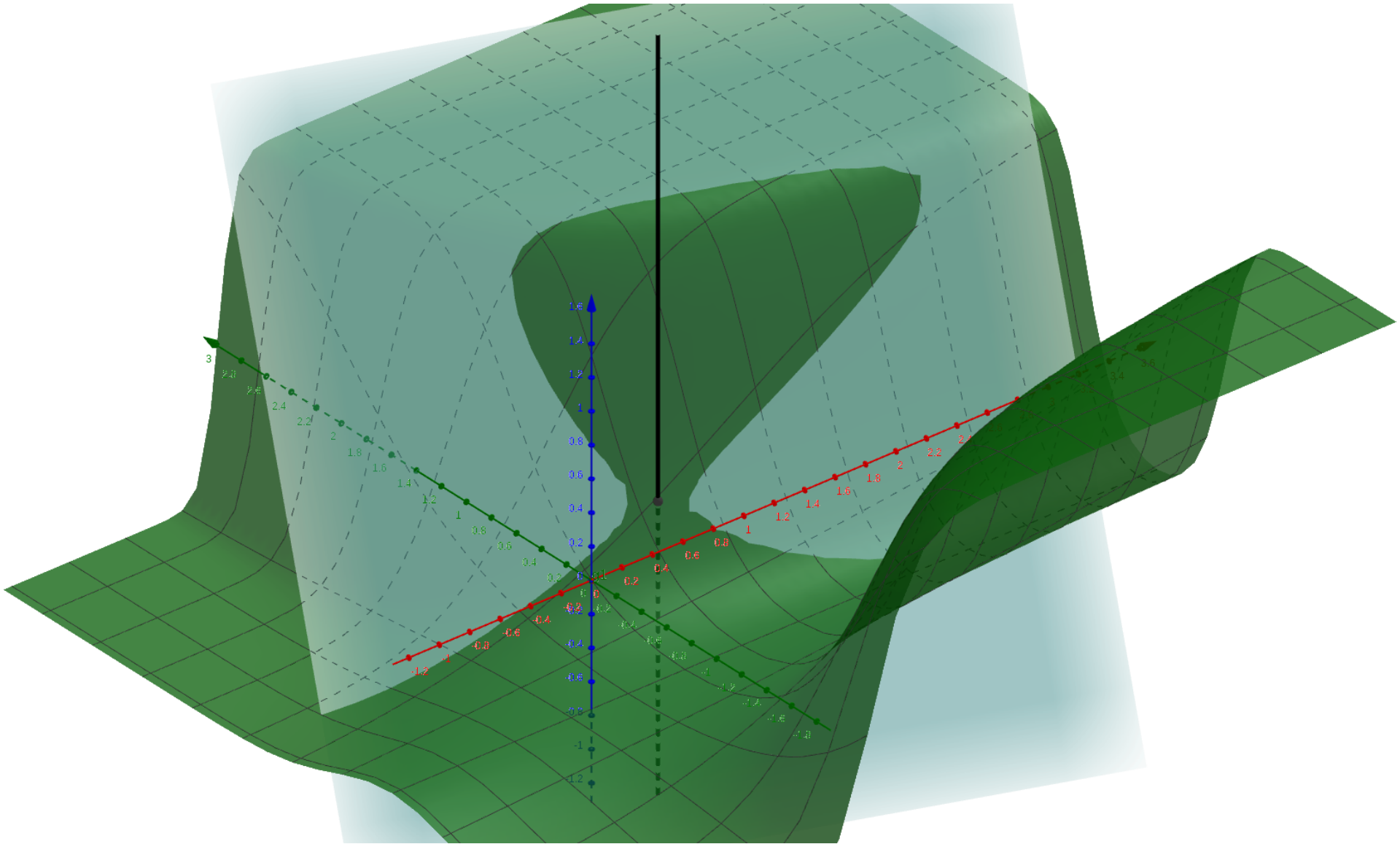
Start with (x_0, y_0)



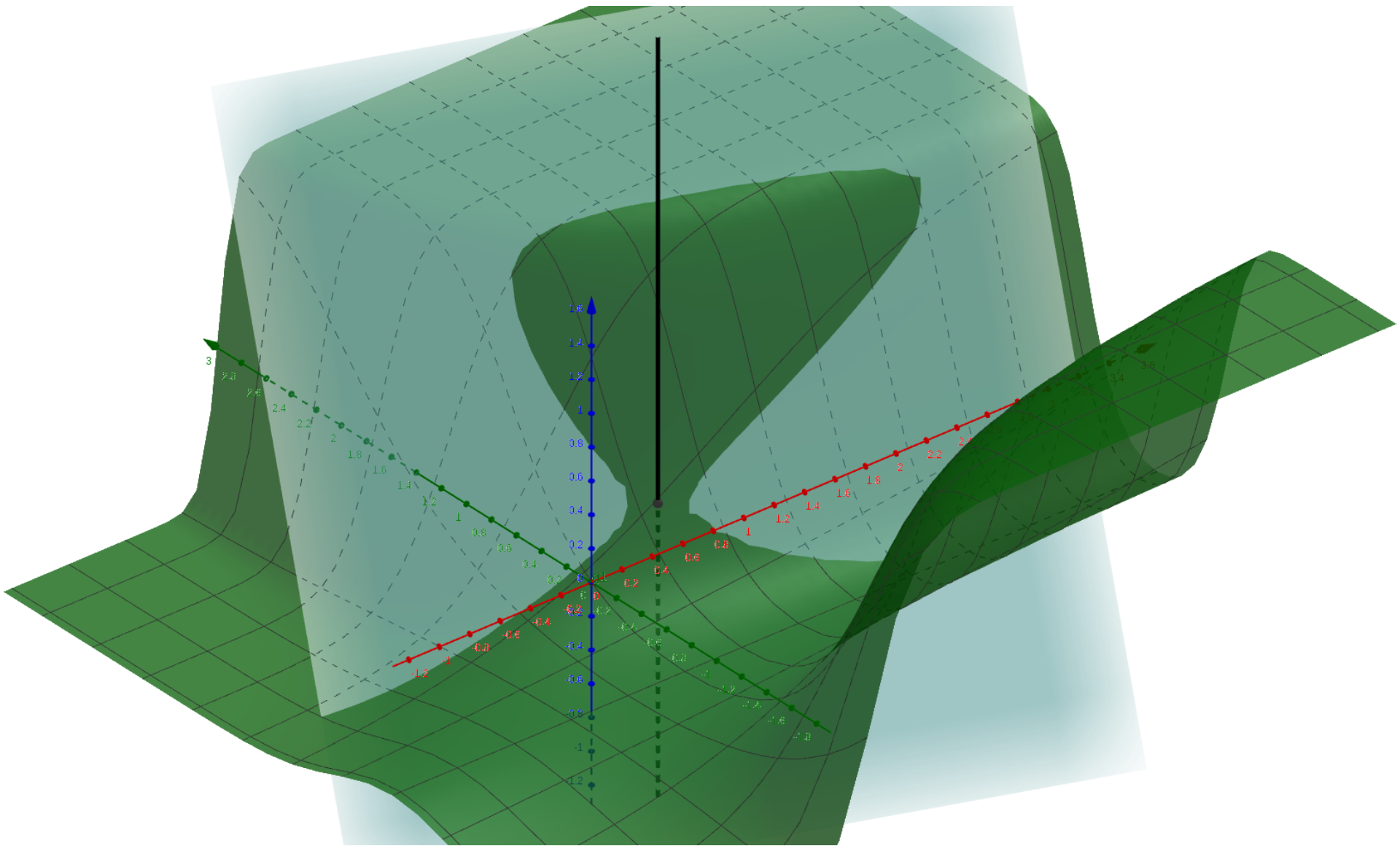
$$g_1(x, y) \approx g_1(x_0, y_0) + \frac{\partial g_1}{\partial x}(x_0, y_0)(x - x_0) + \frac{\partial g_1}{\partial y}(x_0, y_0)(y - y_0)$$



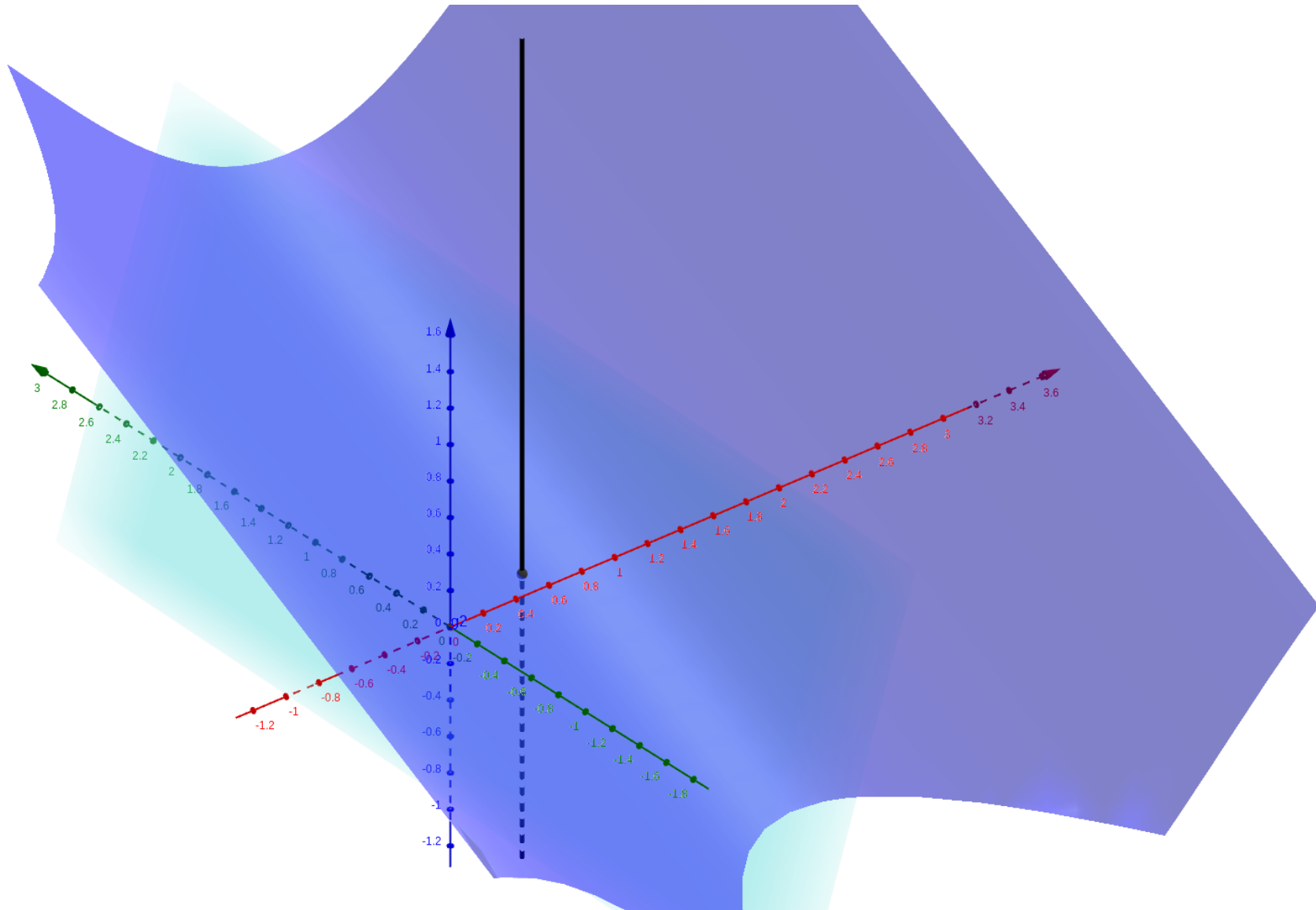
$$g_1(x, y) \approx \bar{g}_1 + \frac{\partial g_1}{\partial x}(x - x_0) + \frac{\partial g_1}{\partial y}(y - y_0)$$



$$g_1(x, y) \approx \bar{g}_1 + \left[\frac{\partial g_1}{\partial x} \quad \frac{\partial g_1}{\partial y} \right] \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

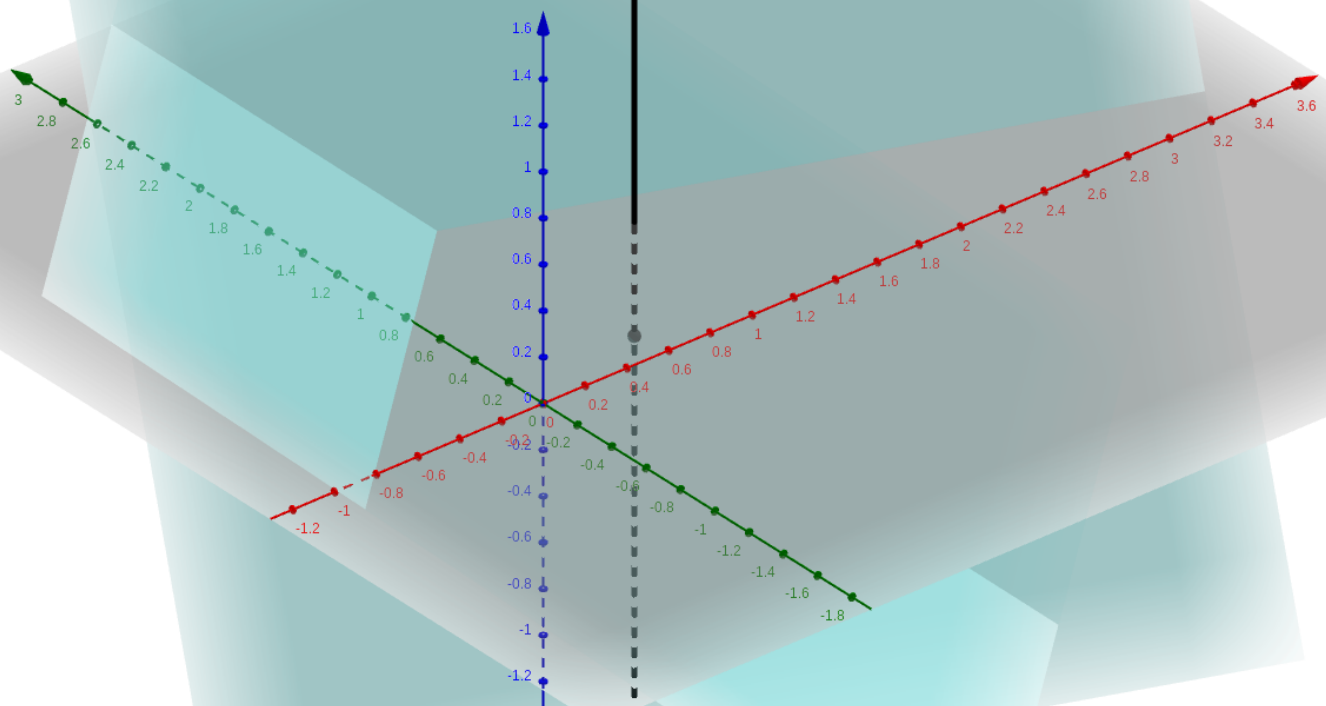


$$g_2(x, y) \approx \bar{g}_2 + \left[\frac{\partial g_2}{\partial x} \quad \frac{\partial g_2}{\partial y} \right] \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$



$$\bar{g}_1 + \begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = 0$$

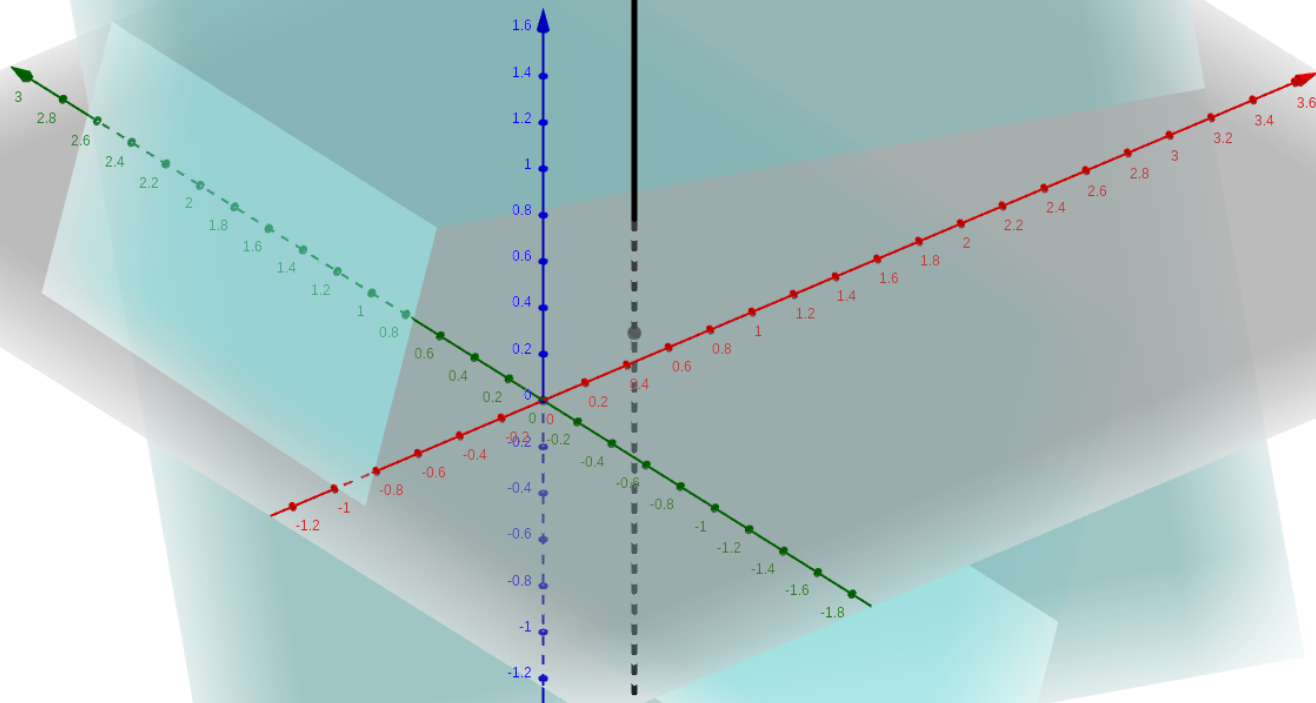
$$\bar{g}_2 + \begin{bmatrix} \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = 0$$



$$\bar{g}_1 + \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1 - x_1^0 \\ x_2 - x_2^0 \end{bmatrix} = 0$$

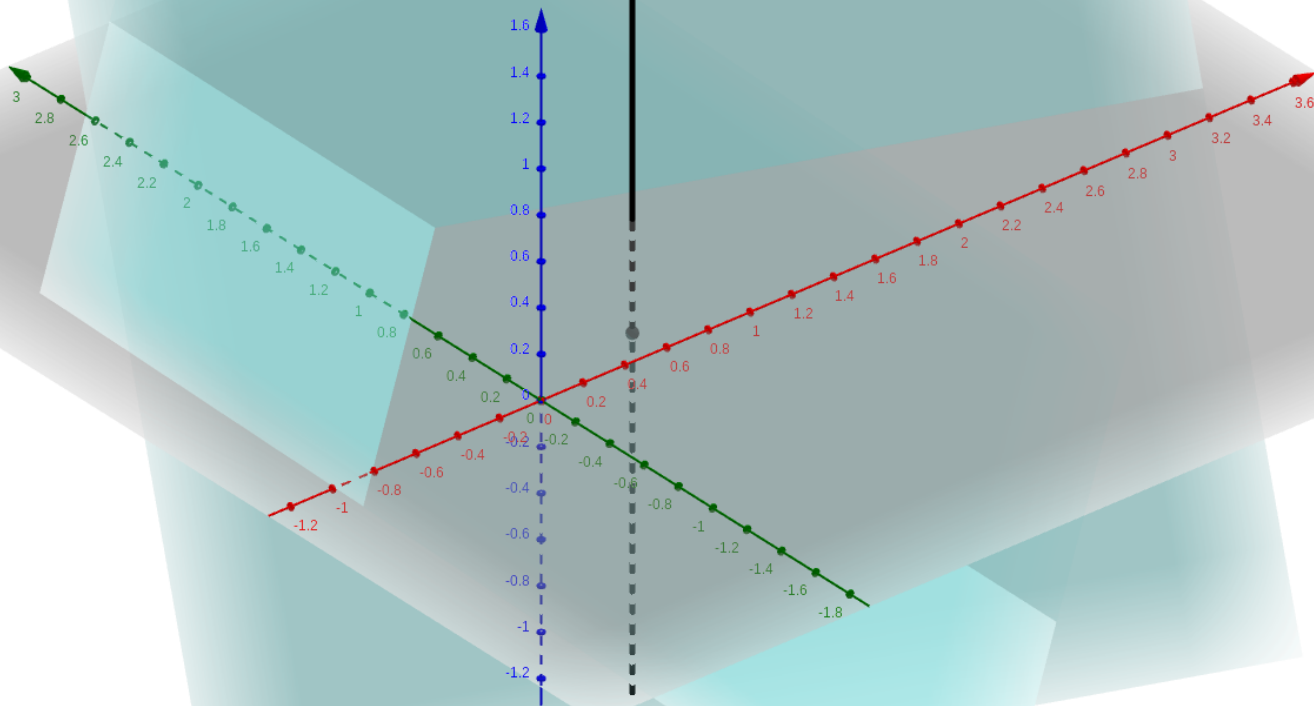
$$\bar{g}_2 + \begin{bmatrix} \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1 - x_1^0 \\ x_2 - x_2^0 \end{bmatrix} = 0$$

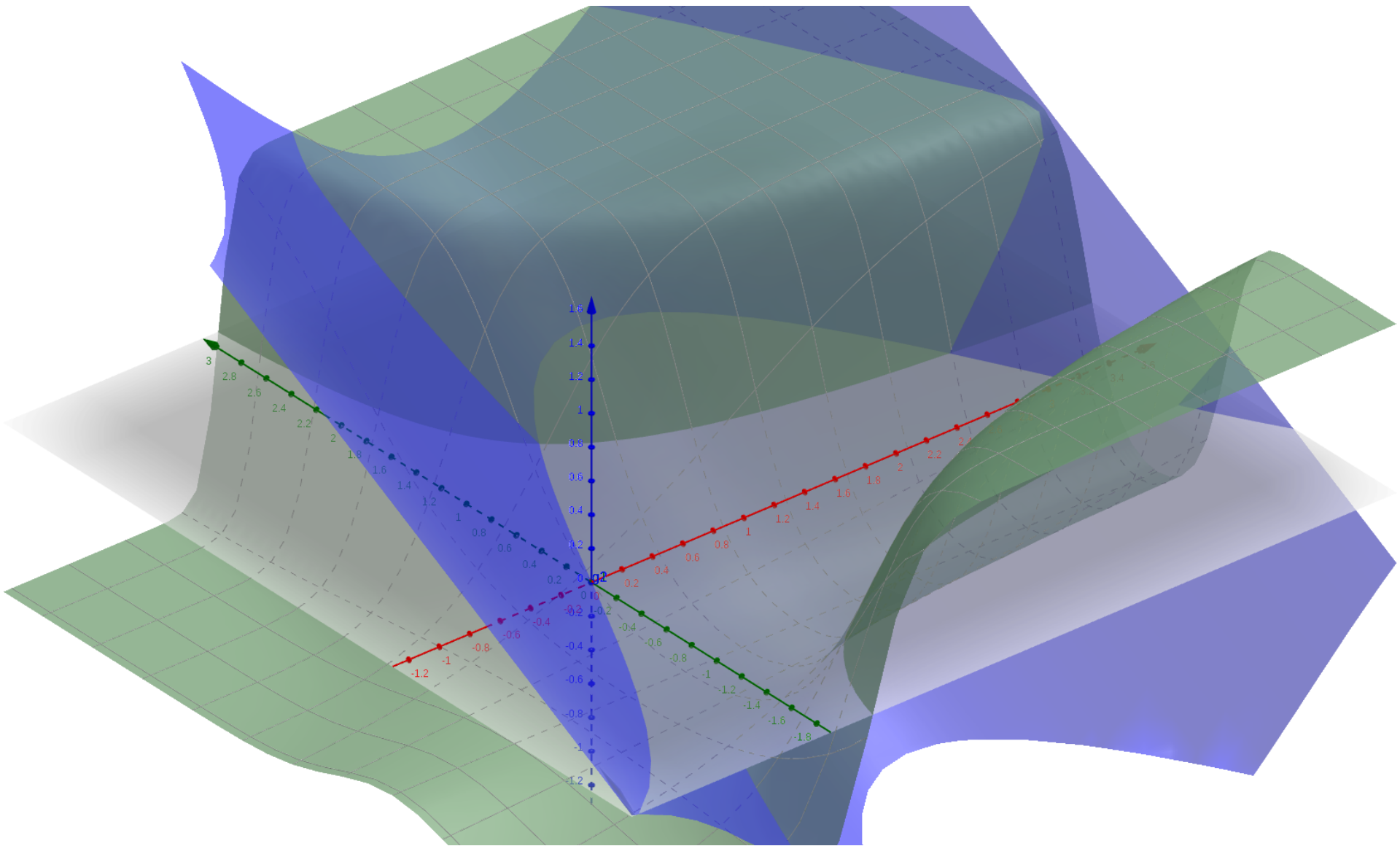
$x \in \mathbb{R}^2$



$$\begin{bmatrix} \bar{g}_1 \\ \bar{g}_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1 - x_1^0 \\ x_2 - x_2^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$g(x_0) + \frac{\partial g}{\partial x}(x_0)(x - x_0) = 0$$





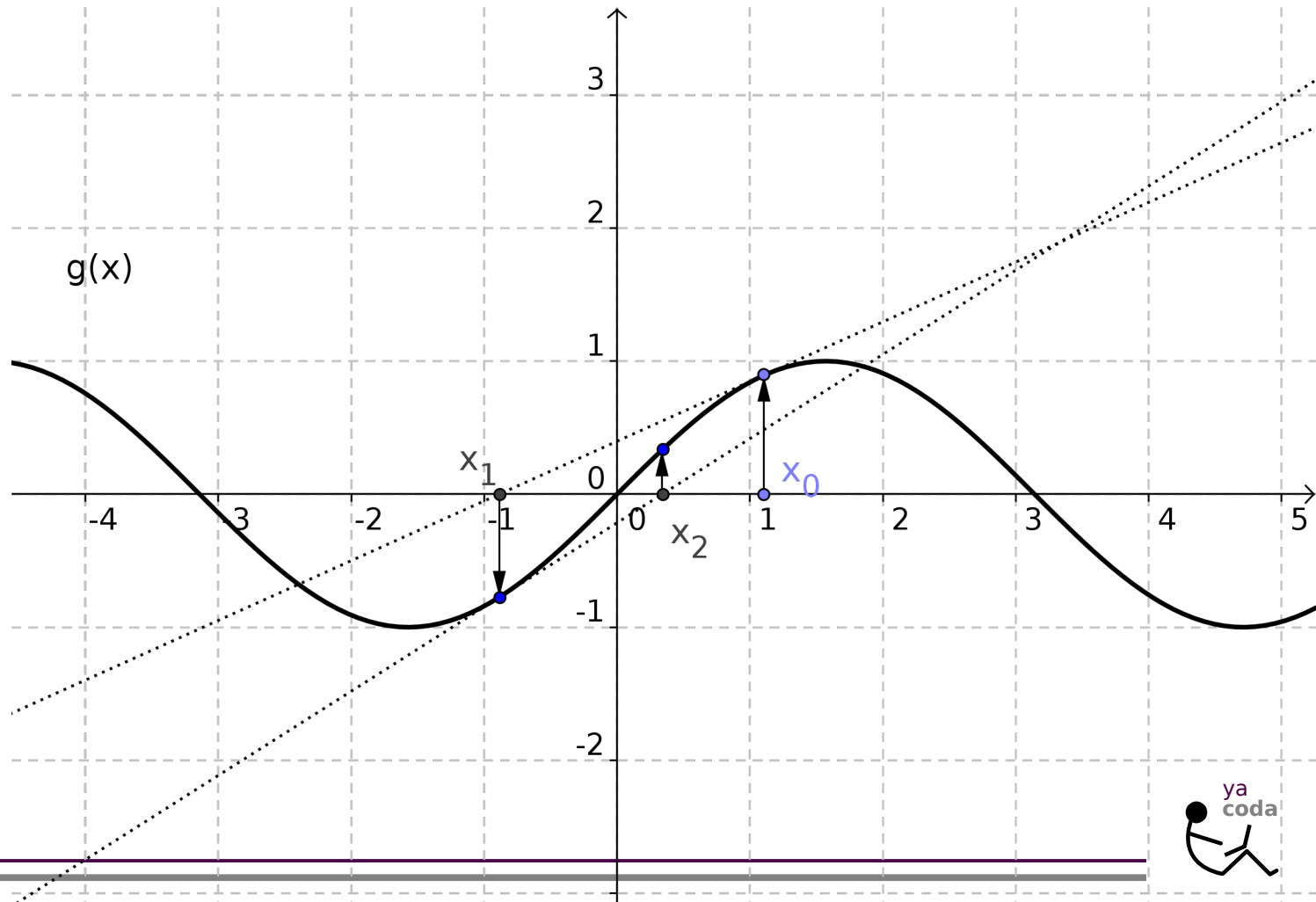
Newton step

$$\Delta x = -\left(\frac{\partial g}{\partial x}(x^k)\right)^{-1} g(x^k)$$

$$x^{k+1} = x^k + \Delta x$$

Newton's method is local

- Solution depends on initial guess

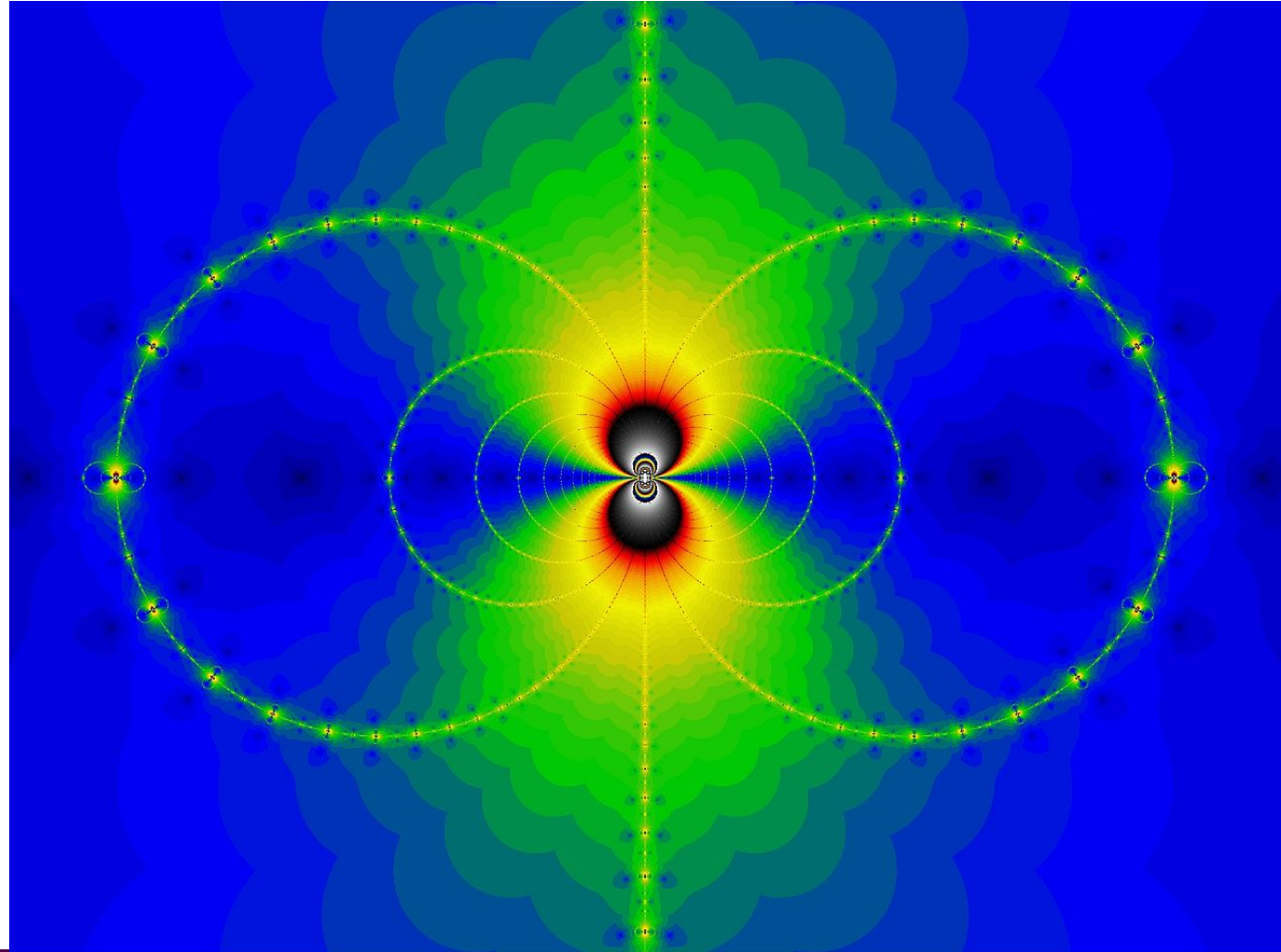


Newton's method is local

- Regions of attraction can be fractal

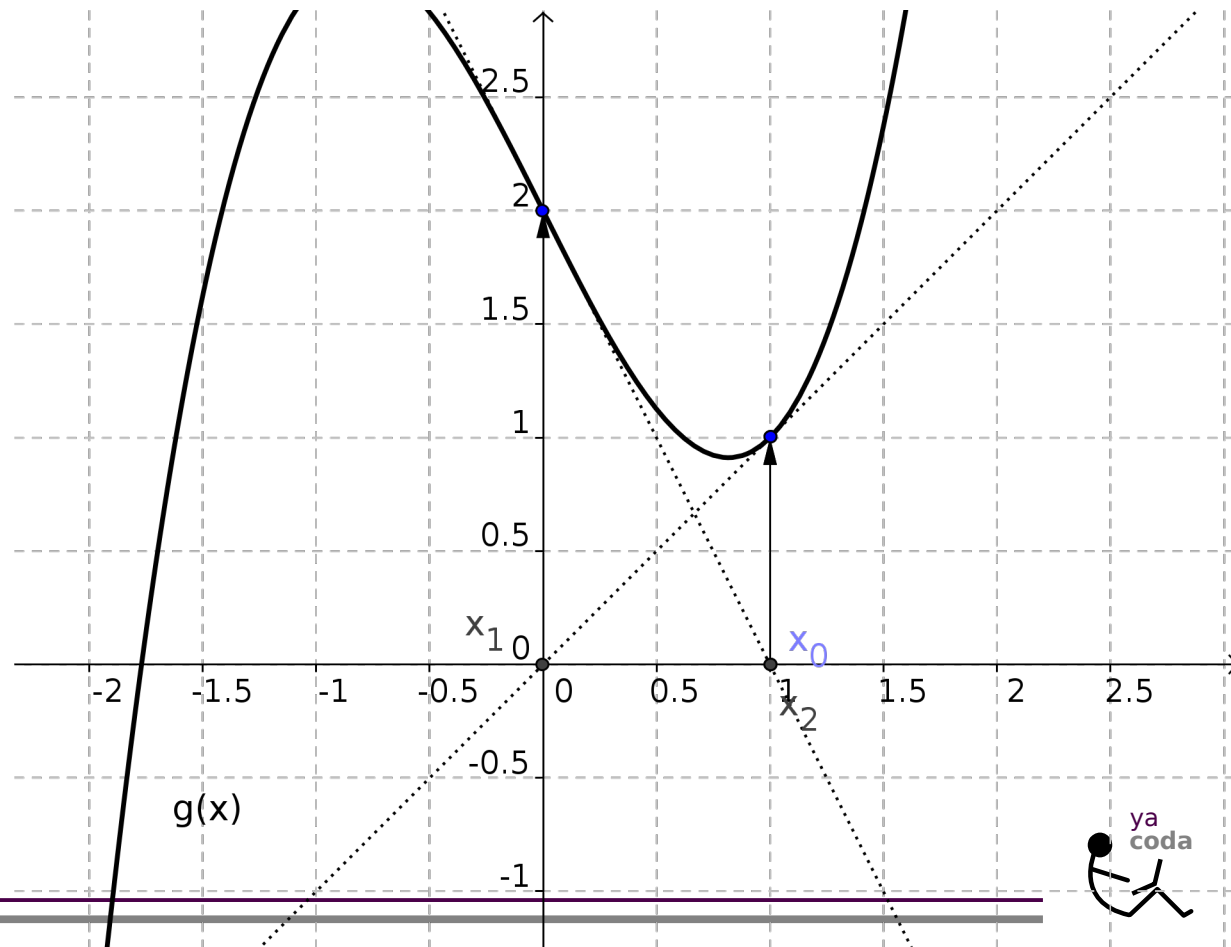
$$g(x) = \sin(x) - 1, x \in \mathbb{C}$$

$$g(x) = [\dots], x \in \mathbb{R}^2$$



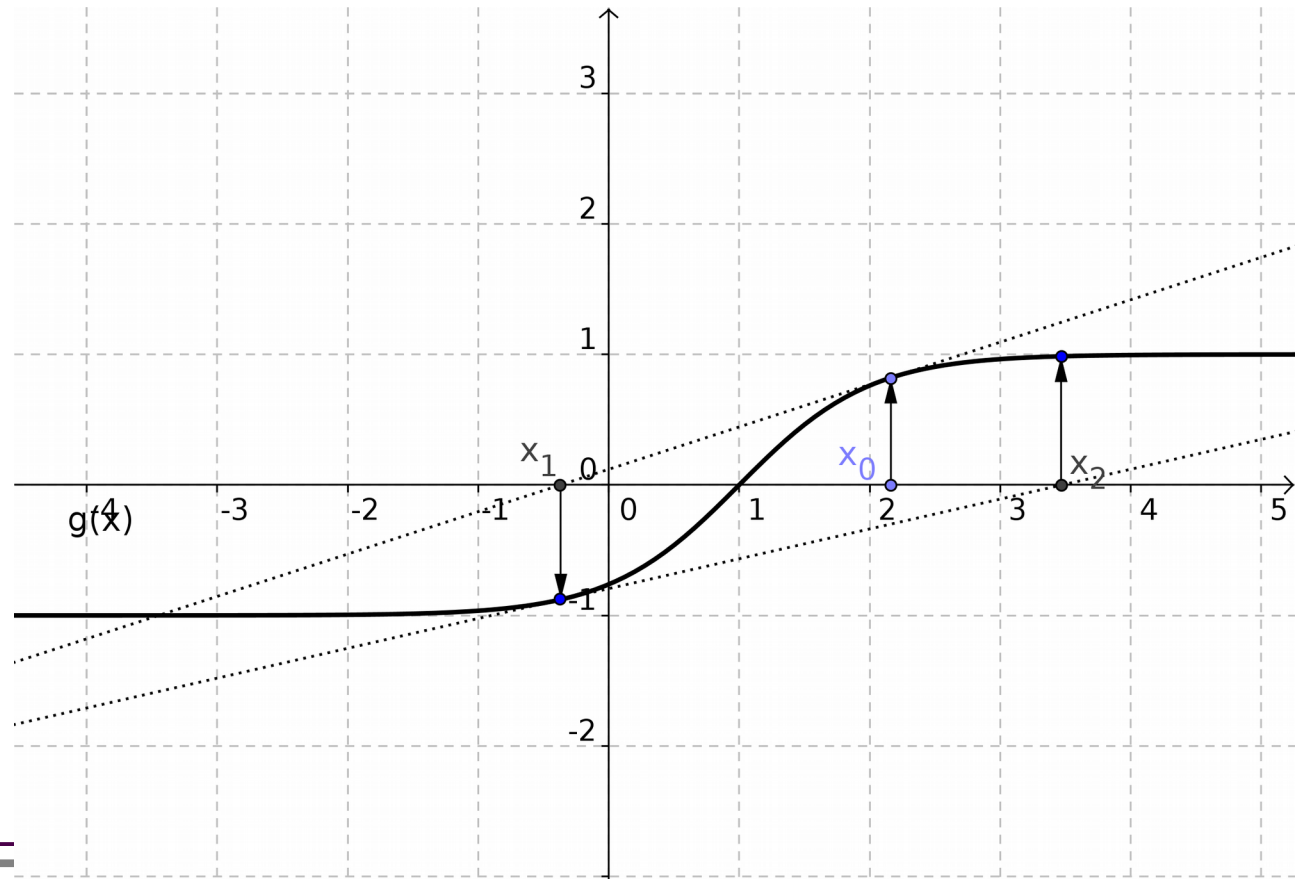
CC BY-SA 3.0
Josep M Batlle i Ferrer

Newton's method may cycle



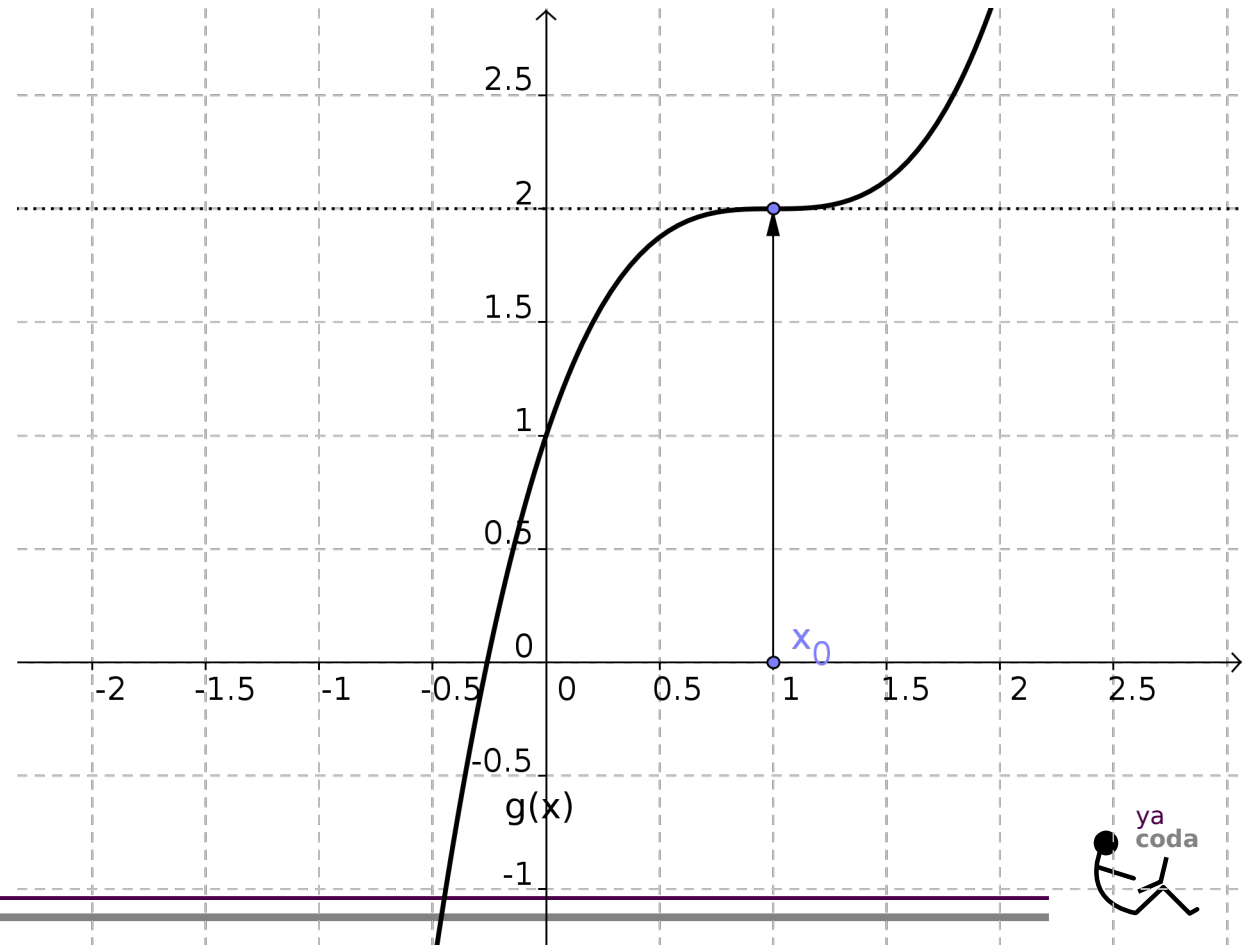
Newton's method may diverge

- “globalisation strategy”: broaden the region of convergence
- Decrease Δx if not sufficient progress made



Newton step may be invalid

- 1D: Slope 0



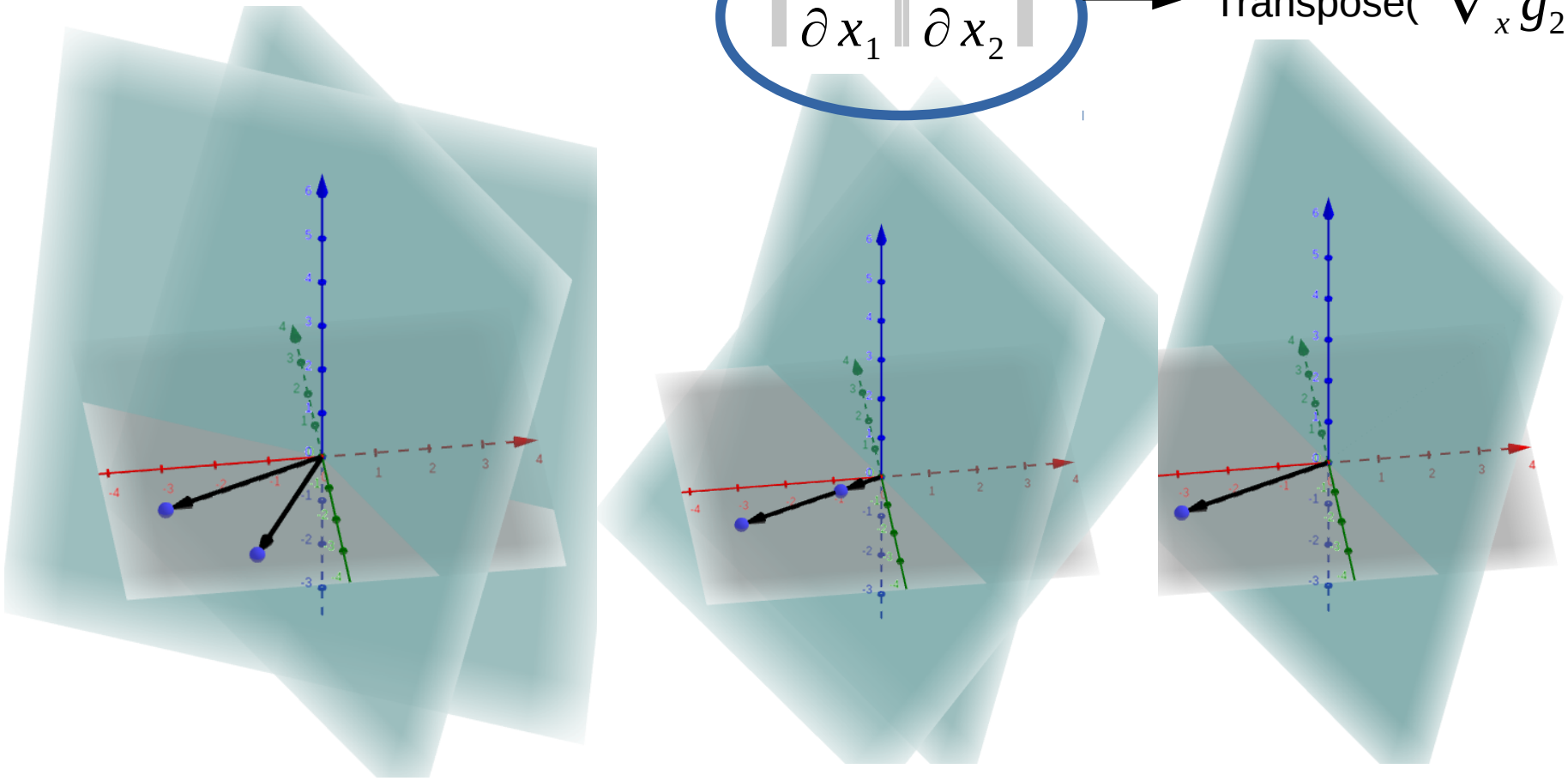
Newton step may be invalid

- 1D: Slope 0
- $\frac{\partial g}{\partial x}(x_k)$ not invertible

$$\begin{array}{|c|c|} \hline \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \hline \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \\ \hline \end{array}$$

→ Transpose($\nabla_x g_1$)

→ Transpose($\nabla_x g_2$)



Newton's method is robust in practice

- In practice, no special precautions needed
- Jacobian $\frac{\partial g}{\partial x}(x_k)$ need not be exact
 - Approximations work
 - Updating for each iteration may not be needed



Tricks for hard real-time performance

How to compute that Jacobian?

- Derive by hand
- Finite differences
- Symbolic differentiation
- Algorithmic differentiation (AD)

Re-use of subexpressions

$$x \in \mathbb{R}^4, g : \mathbb{R}^4 \rightarrow \mathbb{R}^3$$

Jacobian $\frac{\partial g}{\partial x}$:

functions

variables

$\frac{\partial g_1}{\partial x_1}$	$\frac{\partial g_1}{\partial x_2}$	$\frac{\partial g_1}{\partial x_3}$	$\frac{\partial g_1}{\partial x_4}$
$\frac{\partial g_2}{\partial x_1}$	$\frac{\partial g_2}{\partial x_2}$	$\frac{\partial g_2}{\partial x_3}$	$\frac{\partial g_2}{\partial x_4}$
$\frac{\partial g_3}{\partial x_1}$	$\frac{\partial g_3}{\partial x_2}$	$\frac{\partial g_3}{\partial x_3}$	$\frac{\partial g_3}{\partial x_4}$

Finite differences

- Cost of one column $\sim \text{cost}(g)$
- Limited precision
- Scaling issues

$$\frac{\partial g}{\partial x_1} \approx \frac{g \begin{pmatrix} x_1 + \varepsilon \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} - g \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}}{\varepsilon}$$

$\frac{\partial g_1}{\partial x_1}$	$\frac{\partial g_1}{\partial x_2}$	$\frac{\partial g_1}{\partial x_3}$	$\frac{\partial g_1}{\partial x_4}$
$\frac{\partial g_2}{\partial x_1}$	$\frac{\partial g_2}{\partial x_2}$	$\frac{\partial g_2}{\partial x_3}$	$\frac{\partial g_2}{\partial x_4}$
$\frac{\partial g_3}{\partial x_1}$	$\frac{\partial g_3}{\partial x_2}$	$\frac{\partial g_3}{\partial x_3}$	$\frac{\partial g_3}{\partial x_4}$

Finite differences

- Cost of one forward sensitivity $\sim \text{cost}(g)$
- Limited precision
- Scaling issues

$$\frac{\partial g}{\partial x} s \approx \frac{g(x + s\varepsilon) - g(x)}{\varepsilon}$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Forward seed

Forward sensitivity

$\frac{\partial g_1}{\partial x_1}$	$\frac{\partial g_1}{\partial x_2}$	$\frac{\partial g_1}{\partial x_3}$	$\frac{\partial g_1}{\partial x_4}$
$\frac{\partial g_2}{\partial x_1}$	$\frac{\partial g_2}{\partial x_2}$	$\frac{\partial g_2}{\partial x_3}$	$\frac{\partial g_2}{\partial x_4}$
$\frac{\partial g_3}{\partial x_1}$	$\frac{\partial g_3}{\partial x_2}$	$\frac{\partial g_3}{\partial x_3}$	$\frac{\partial g_3}{\partial x_4}$

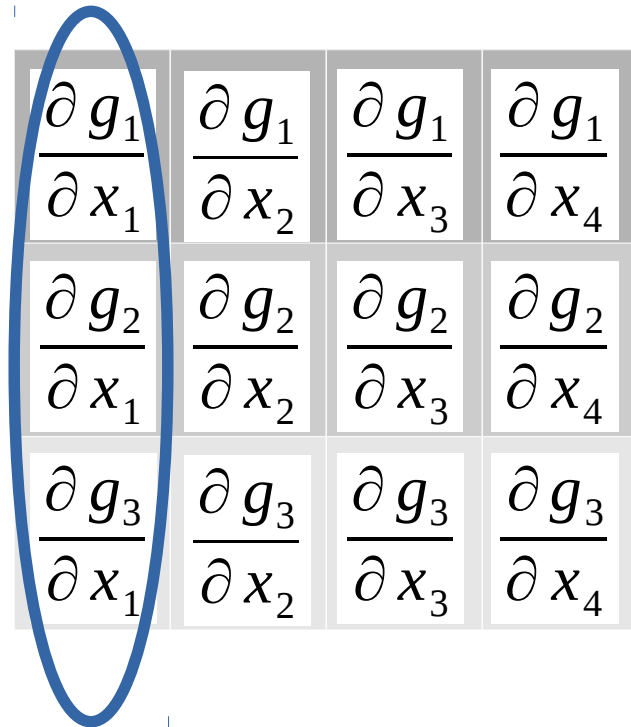
Algorithmic differentiation (AD)

- Cost of one forward sensitivity $\sim \text{cost}(g)$
- Machine precision (no worries about scaling)
- Given an algorithm, $g(x)$, can construct a forward algorithm

$$\frac{\partial g}{\partial x} s = g^{\text{forward}}(x, s)$$

Forward seed

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$\frac{\partial g_1}{\partial x_1}$	$\frac{\partial g_1}{\partial x_2}$	$\frac{\partial g_1}{\partial x_3}$	$\frac{\partial g_1}{\partial x_4}$
$\frac{\partial g_2}{\partial x_1}$	$\frac{\partial g_2}{\partial x_2}$	$\frac{\partial g_2}{\partial x_3}$	$\frac{\partial g_2}{\partial x_4}$
$\frac{\partial g_3}{\partial x_1}$	$\frac{\partial g_3}{\partial x_2}$	$\frac{\partial g_3}{\partial x_3}$	$\frac{\partial g_3}{\partial x_4}$

Algorithmic differentiation (AD)

- Cost of one forward sensitivity $\sim \text{cost}(g)$
- Machine precision (no worries about scaling)
- Given an algorithm, $g(x)$, can construct a reverse algorithm

$$\left[\frac{\partial g}{\partial x} \right]^T s = g^{\text{reverse}}(x, s)$$

Reverse seed $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

$\frac{\partial g_1}{\partial x_1}$	$\frac{\partial g_1}{\partial x_2}$	$\frac{\partial g_1}{\partial x_3}$	$\frac{\partial g_1}{\partial x_4}$
$\frac{\partial g_2}{\partial x_1}$	$\frac{\partial g_2}{\partial x_2}$	$\frac{\partial g_2}{\partial x_3}$	$\frac{\partial g_2}{\partial x_4}$
$\frac{\partial g_3}{\partial x_1}$	$\frac{\partial g_3}{\partial x_2}$	$\frac{\partial g_3}{\partial x_3}$	$\frac{\partial g_3}{\partial x_4}$

Algorithmic differentiation (AD)

- Jacobian can be composed from forward/reverse sweeps
- Reverse sweeps can be a real game-changer
 - Unconstrained optimization: deep learning
- Sparsity can be exploited
 - Can combine seeds when rows/cols structurally independent

rootfinding