Goal-Oriented Graphics Animation:
A Collision-Free Path

In this assignment, you will write a program that creates an animation of the motion of an object on a plane. The object moves from an initial position to a goal position, while avoiding collisions with various obstacles that are located between the object and its goal. The method to be implemented is described in Example 1 of Breen's paper [1]. Figure 1 shows an example output for this animation.

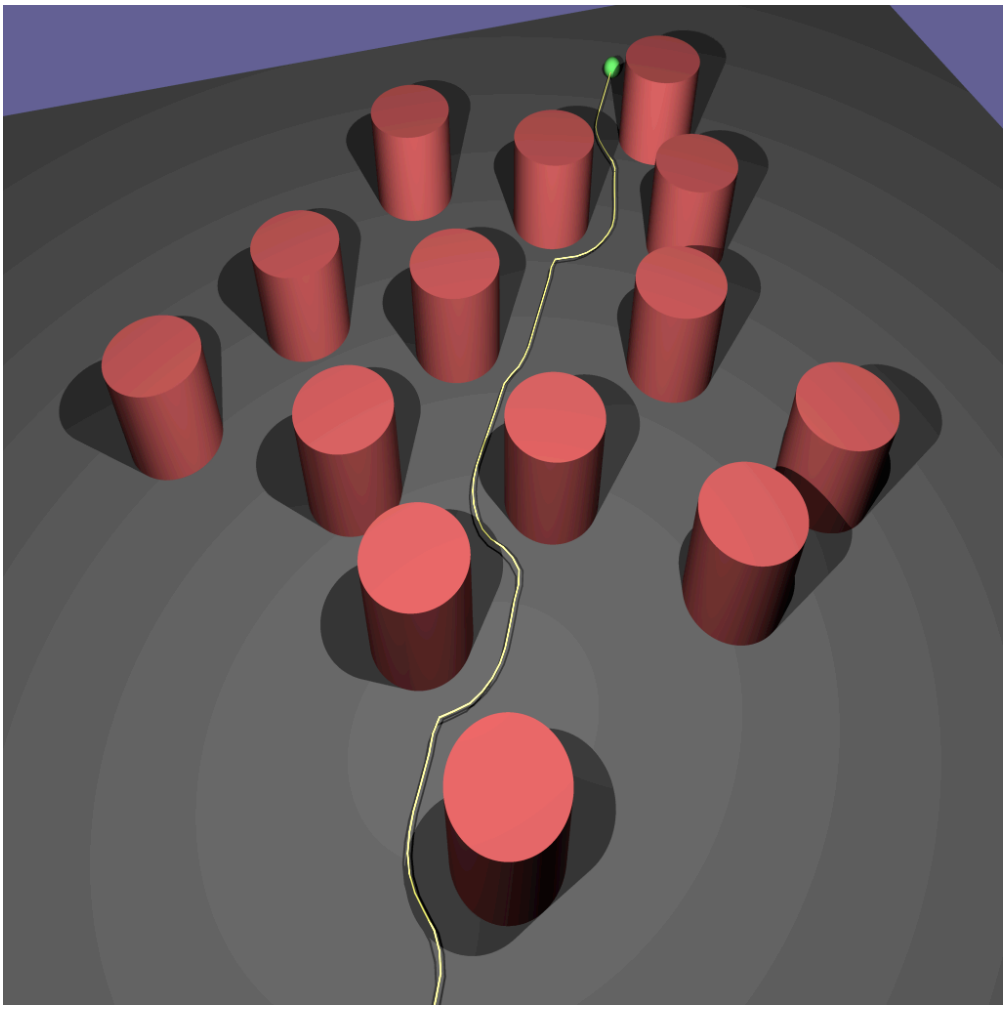


Figure 1: Goal-oriented animation: The motion path for a particle.

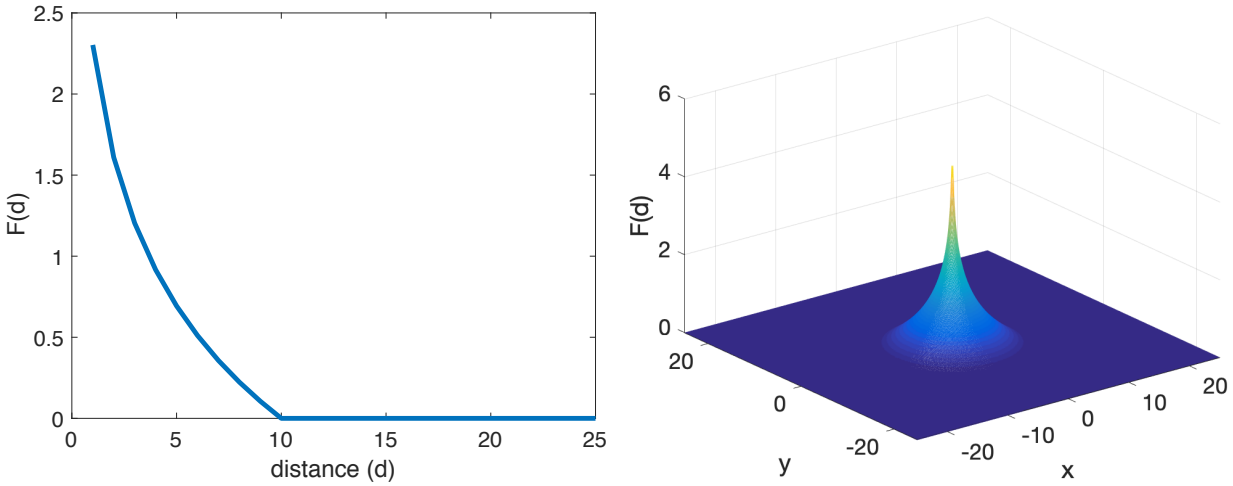
The animation is achieved by minimizing the following cost function:

$$C(\mathbf{x}) = \|\mathbf{x} - \mathbf{g}\| + \sum_{i=1}^{n} \mathcal{F}(\|\mathbf{x} - \mathbf{o}_i\|), \tag{1}$$

where $\mathbf{x}$ is the current location of the animated object, $\mathbf{g}$ is the goal location, $\mathbf{o}_i$ is the location of obstacle i, and $\mathcal{F}$ is a penalty field for collision avoidance. An example of a field function is given as follows [1]:

$$\mathcal{F}(d) = \begin{cases} \ln(R/d), & 0 < d \leq R, \\ 0, & d > R. \end{cases} \tag{2}$$

where R is the radius of the obstacle. If we set $R = 10$, the field function looks like the plot shown in Figure 2. The value of $\mathcal{F}$ decreases as the distance d increases. Once the distance is larger than R, the value of the function vanishes and so does its influence in the overall cost calculated by Equation 1.



(a) (b)

Figure 2: Penalty field function. (a) One-dimensional plot of the penalty function. Penalty value increases as particle approaches the obstacle. (b) 2-D representation of the field function for $d = \sqrt{x^2 + y^2}$, i.e., distance from any point (x, y) to the origin.

To minimize Equation 1, we can use the gradient-descent algorithm, which is given by:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \lambda \nabla f(\mathbf{x}_n), \quad n \geq 0, \tag{3}$$

where ∇C is the gradient of the cost function, i.e.:

$$\nabla C = \left[\frac{\partial C}{\partial x} \quad \frac{\partial C}{\partial y}\right]^{\mathsf{T}} \tag{4}$$

The gradient can be approximated by $\Delta C/\Delta P$, using finite differences. Here, simply evaluate the function at a local grid centered at the current location of the particle and calculate the gradient vector.

1 Calculus review

1.1 Derivative of a scalar function of a single scalar variable

Let f be a scalar function of a single variable x. The derivative of the function w.r.t. x is:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{\Delta f}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}. \tag{5}$$

The derivative of a scalar function describes the slope (i.e., rate of change) of the function at a given point x. Figure 3 shows the derivative of the function $f(x) = x \sin x^2 + 1$ at $x = -1$, -0.5, 1, 0.5, and 1. The derivative of the function is $\dfrac{df}{dx}(x) = 2x^2 \cos x^2 + \sin x^2$.

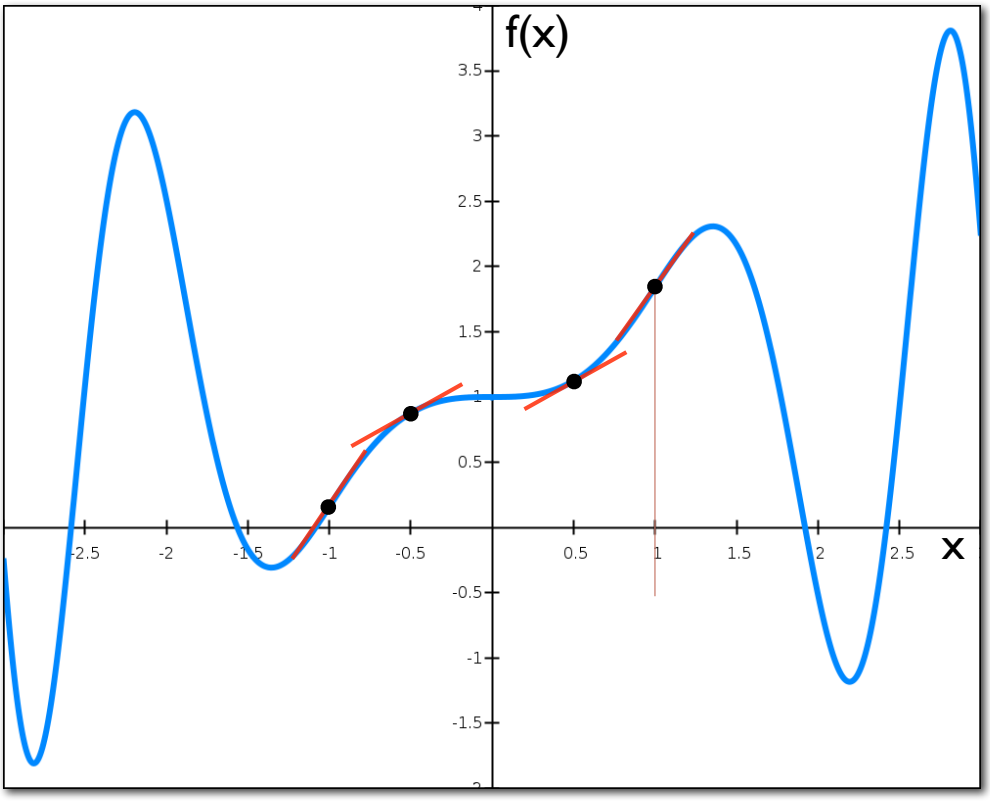


Figure 3: The derivative of the function $f(x) = x \sin x^2 + 1$ at $x = -1$, -0.5, 1, 0.5, and 1.

1.2 Derivative approximation

Often, calculating the derivative analytically can be very hard. In these cases, we can make progress by approximating the derivative numerically by using discrete differences. In fact, as long as we can evaluate the function, we can always approximate the derivative, i.e.:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{\Delta f}{\Delta x} = \frac{f\left(x + \Delta x\right) - f\left(x\right)}{\Delta x}, \tag{6}$$

for a small Δx. Figure 4 shows the approximate derivative of the function for a small Δx.

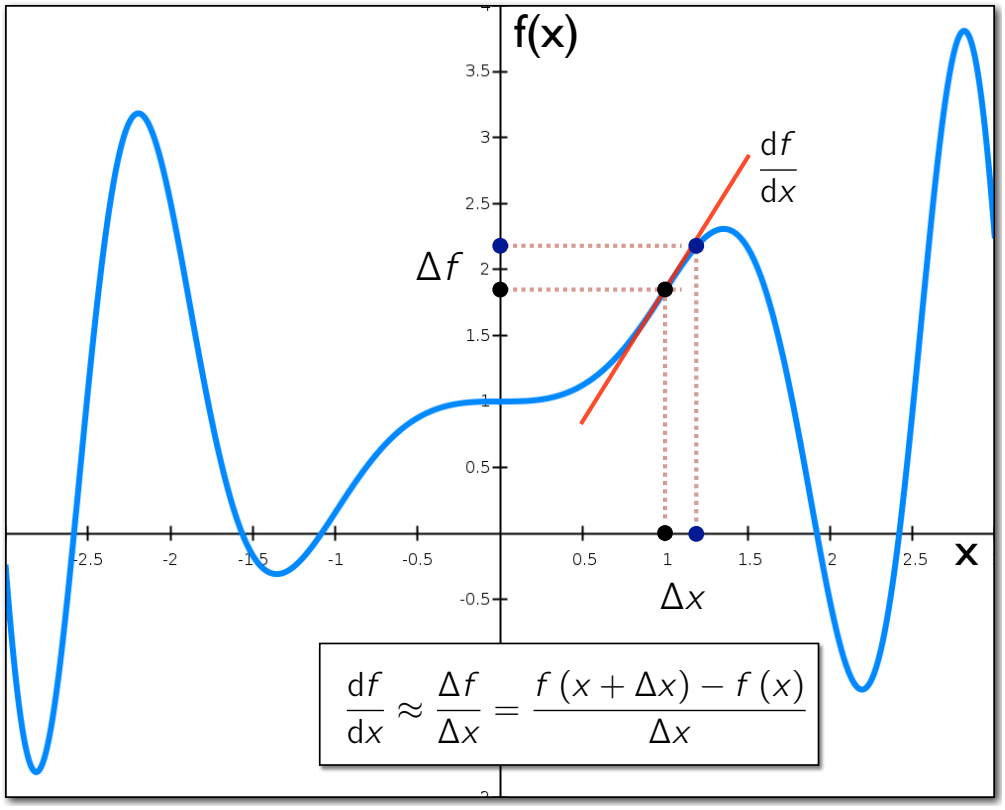


Figure 4: The approximate derivative of the function $f(x) = x \sin x^2 + 1$ at $x = -1$.

1.3 Calculating the value of functions at nearby points

Another useful tool from derivatives is that it allows us to calculate the value of a function at nearby points. Given the value of a function at a point, $f(x)$, and its derivative, df/dx,

we can estimate the value of the function at a point near x, i.e.:

$$\begin{aligned} \frac{\Delta f}{\Delta x} &\approx \frac{\mathrm{d}f}{\mathrm{d}x} \\ \Delta f &\approx \Delta x \frac{\mathrm{d}f}{\mathrm{d}x} \\ f\left(x+\Delta x\right) - f\left(x\right) &\approx \Delta x \frac{\mathrm{d}f}{\mathrm{d}x} \\ f\left(x+\Delta x\right) &\approx f\left(x\right) + \Delta x \frac{\mathrm{d}f}{\mathrm{d}x}. \end{aligned} \tag{7}$$

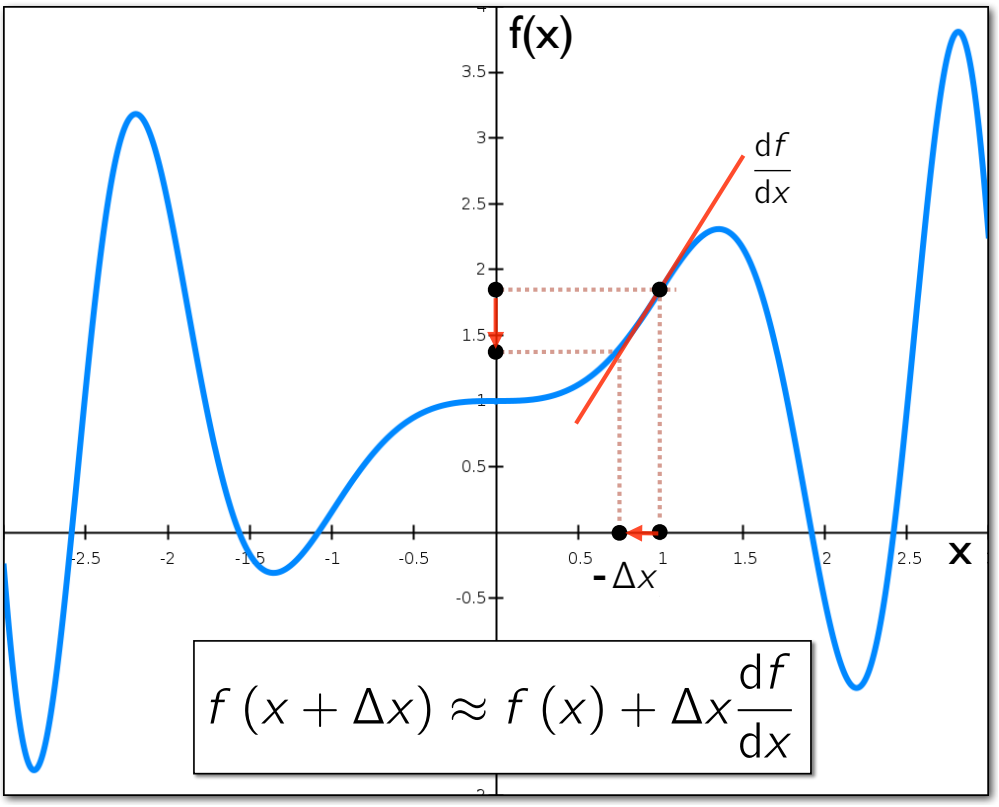


Figure 5: Derivatives allow us to calculate the value of a function at nearby points. Given the value of a function at a point, $f(x)$, and its derivative, df/dx, we can estimate the value of the function at a point near x

1.4 The gradient-descent method

Many times, we want to find the value for which a function is zero, i.e., we want to find solutions to the equation $f\left(x\right) = 0$. This equation can be solved analytically or numerically. One of the many numerical methods to solving this equation is the gradient-descent method.

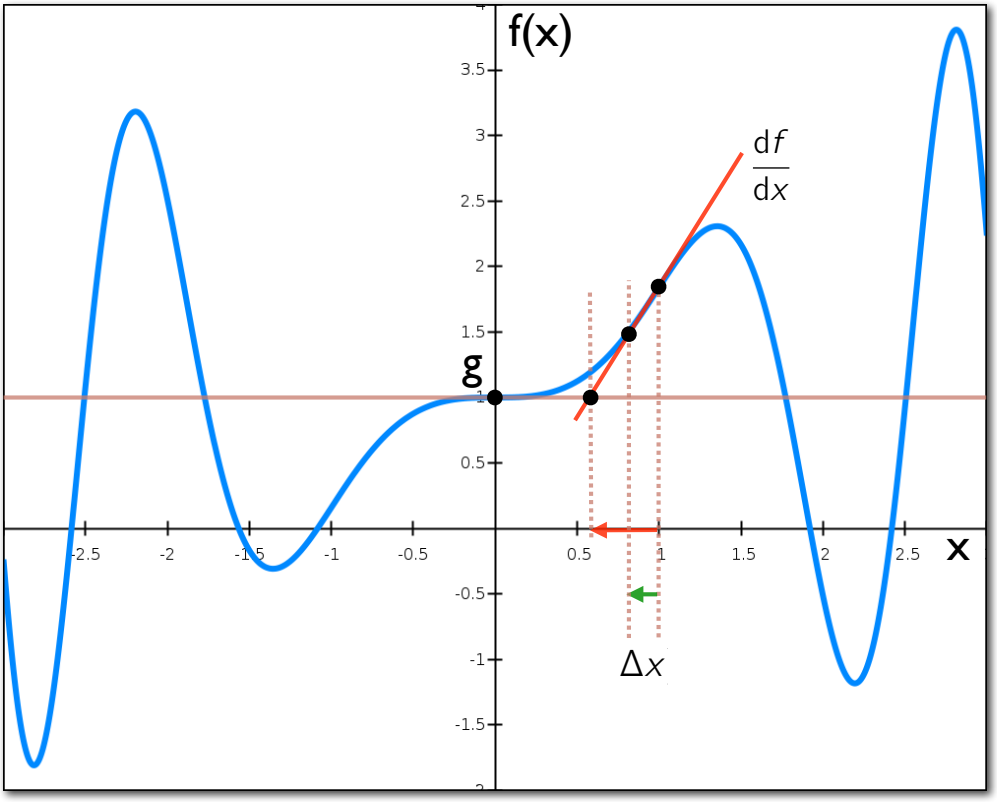


Figure 6: Computing the gradient descent's Δx step. The red arrow shows a step that was computed using $\Delta x = (g - f(x_i)) \left(\frac{df}{dx}\right)^{-1}$, which may be too large (and over optimistic) given the function's linear approximation df/dx. In contrast, the green arrow shows a (smaller) step that was computed using $\Delta x = \beta (g - f(x_i)) \left(\frac{df}{dx}\right)^{-1}$, with $0 < \beta \leq 1$. For instance, we can set $\beta = 0.1$.

If we can evaluate $f(x)$ and df/dx for any value of x, we can always follow the slope (i.e., gradient) in the direction towards 0. Starting at some initial value x_0, take small steps until we find a value x_n for which $f(x_n) = 0$.

$$x_{i+1} = x_i + \Delta x. \tag{8}$$

For each step, we choose a value of Δx that brings us closer to our goal. We can try to

choose Δx to bring us closer to where the slope passes through 0, i.e.:

$$\begin{aligned}
\frac{\Delta f}{\Delta x} &\approx \frac{\mathrm{d}f}{\mathrm{d}x} \\
\Delta f &\approx \Delta x \frac{\mathrm{d}f}{\mathrm{d}x} \\
\cancelto{0}{f\left(x_i + \Delta x\right)} - f\left(x_i\right) &\approx \Delta x \frac{\mathrm{d}f}{\mathrm{d}x} \\
-f\left(x_i\right) &\approx \Delta x \frac{\mathrm{d}f}{\mathrm{d}x} \\
\Delta x &= -f\left(x_i\right) \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^{-1}. \qquad (9)
\end{aligned}$$

If we want to find where a function equals some value g instead of 0, we can think of the problem as minimizing $f\left(x\right) - g$ and just step towards g, i.e.:

$$\Delta x = \left(g - f\left(x_i\right)\right) \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^{-1}. \qquad (10)$$

However, Equation 10 assumes that our linear approximation of the function given by its derivative is reliable for large values of Δx. However, this is not the case for non-smooth functions with varying derivatives. A safer way to choosing Δx is to multiply it by a parameter $\beta \in (0, 1]$ to scale the step. With the inclusion of the β scale factor, Equation 10 can be re-written as:

$$\Delta x = \beta \left(g - f\left(x_i\right)\right) \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^{-1}. \qquad (11)$$

Figure 6 shows the Δx steps computed by using Equations 10 and 11.

Algorithm 1 Gradient descent (scalar function of a single scalar variable)

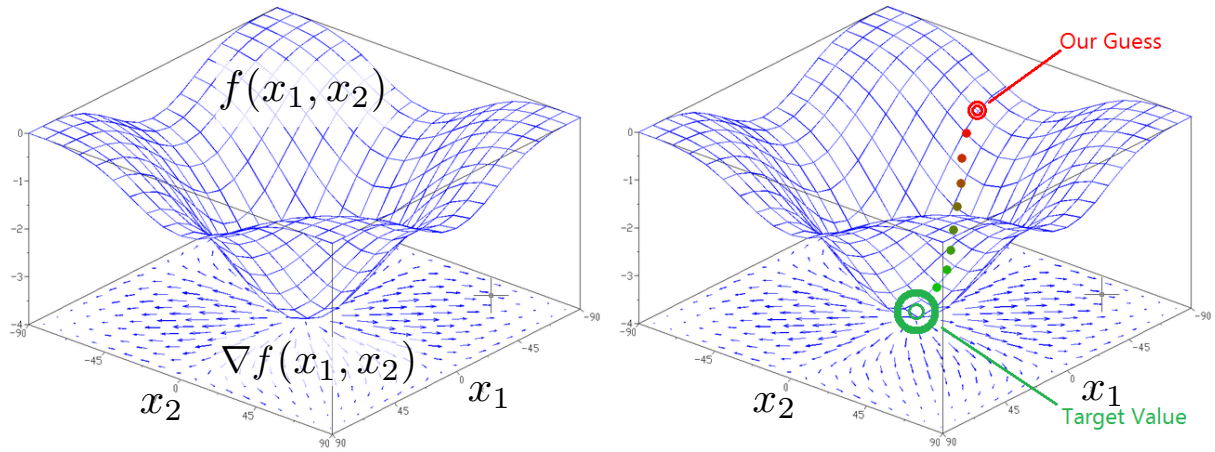
1: $x_0 \leftarrow$ starting value
2: $f_0 \leftarrow f\left(x_0\right)$ ▷ Evaluate f at x_0
3: **while** $f_n \neq g$ **do**
4: $\quad s_i \leftarrow \frac{\mathrm{d}f}{\mathrm{d}x}\left(x_i\right)$ ▷ Compute slope
5: $\quad x_{i+1} \leftarrow x_i + \beta \left(g - f_i\right) \frac{1}{s_i}$ ▷ Take a step along Δx
6: $\quad f_{i+1} \leftarrow f\left(x_{i+1}\right)$ ▷ Evaluate f at new x_{i+1}
7: **end while**

1.5 Derivative of a scalar function of multiple scalar variables (i.e., vector variable)

Let f be a scalar function of a vector variable. The vector variable is $\mathbf{x} = (x_1, x_2, \ldots, x_N)^\mathsf{T}$. This type of function is also called a scalar function of multiple variables or multi-variate function. The value of the function at a point $\mathbf{x}$ is given by $f(\mathbf{x})$ or $f(x_1, x_2, \ldots, x_N)$, and its derivative w.r.t. x is:

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} = \nabla f = \frac{\partial f}{\partial (x_1, x_2, \ldots, x_N)} = \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2} \; \cdots \; \frac{\partial f}{\partial x_N}\right]^\mathsf{T}, \tag{12}$$

which is called the *Gradient* of f at x, and denoted by ∇f. An example of a gradient vector field of a function $f(x_1, x_2)$ is shown in Figure 7.



(a) (b)

Figure 7: (a) Gradient vector field of two-variate function. (b) Gradient-descent method. In the gradient-descent method, we want to take the direction inverse to the gradient as the gradient (slope) points "uphill". Plots adapted from `https://goo.gl/zejgBw`.

References

[1] David E. Breen. Cost minimization for animated geometric models in computer graphics. *The Journal of Visualization and Computer Animation*, 8(4):201–220, 1997.