$$\boxed{\textbf{Introduction to Nonlinear Equations}}$$

In this section we will cover two main algorithms to solve non-linear equations,

- Gradient Descent
- Newton's Method

Both of these algorithms are greatly used in daily tasks and both depend on gradients. We will start with the least square problem to introduce gradient descent.

1 Gradient Descent for Linear Regression

In the previous lecture, we studied the least square problem for linear regression, where by setting the gradient of the mean square error with respect to the parameters-vector to zero, we can solve for the optimal value,

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) \;=\; \frac{1}{n}\left(-\mathbf{X}^\top\mathbf{Y}+\mathbf{X}^\top\mathbf{X}\,\mathbf{w}\right)=\mathbf{0}. \tag{1}$$

where the optimal $\mathbf{w}^*$ is given by,

$$\mathbf{w}^* \;=\; (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}\,. \tag{2}$$

We didn't discussed the computational cost of this approach. If we consider a linear model with a large number of parameters,

$$f(x,\mathbf{w})=\mathbf{w}^\top\Phi(x)=\begin{bmatrix}w_0, & w_1, & \cdots, & w_p\end{bmatrix}\underbrace{\begin{bmatrix}x^0\\ x^1\\ \vdots\\ x^p\end{bmatrix}}_{\Phi(x)}=\sum_{i=1}^{p}w_i x^i. \tag{3}$$

meaning, for example $p\sim 10^4$, this approach is unfeasible as the matrix inversion of $(\mathbf{X}^\top\mathbf{X})^{-1}$ scales cubically with the size of the matrix.

An alternative approach is to use a **search algorithm** where each step should guide us closer to the solution of our problem. One of the most used strategies is the **gradient descent**,

$$\mathbf{w}_{\text{new}}=\mathbf{w}_{\text{old}}-\eta\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w})\big|_{\mathbf{w}_{\text{old}}} \tag{4}$$

where

- η is a scale parameter, commonly with value lower than 1.
- $\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w})\big|_{\mathbf{w}_{\text{old}}}$ is the Jacobian (gradient) of the function $\mathcal{L}(\mathbf{w})$.

In gradient descent, each step (Eq. 4), "should" move us closer to the minima of the function. We must recall that when we reach a minima/maxima, $\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w})\approx\mathbf{0}$, meaning $\mathbf{w}_{\text{new}}\approx\mathbf{w}_{\text{old}}$; meaning we stop updating the current value of the parameters. Fig. 1, illustrates the **trajectory** to find the optimal solution of linear regression problem, $f(x)=a\,x+b$. Each step is carried using Eq. 4, where for linear models $\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w})$ is given by Eq. 1.

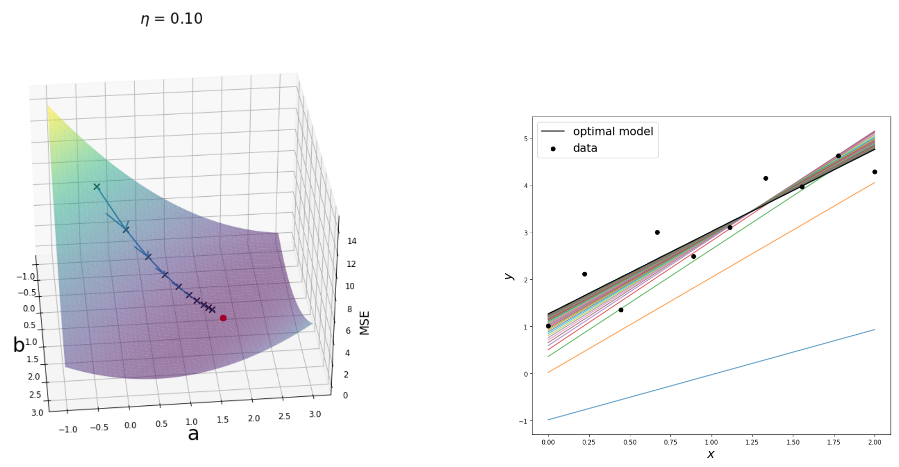

Figure 1: (left panel) Gradient descent trajectory to minimize the mean square error for a linear model, $f(x) = a\,x + b$. (right panel) The prediction of the models sampled through gradient descent. The ● and the black solid line are the optimal parameters found by Eq. 2.

1.1 Why $-\eta \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w})$?

Gradient descent is driven by the local knowledge of the function multiply by -1. Let's consider the minimization of $f(x) = x^2$, see Fig. 2. If we sample a point where x is positive, $x > 0$, $f(x)$ is also positive, the same can be said if $x < 0$. However, the derivatives at that specific locations $\frac{d\,f(x)}{d\,x}\Big|_{x_0}$, is not the same. For $x_0 = 1$, $\frac{d\,f(x)}{d\,x}\Big|_{x_0=1} = 2$, while for $\frac{d\,f(x)}{d\,x}\Big|_{x_0=-1} = -2$. The derivative tells us the change of the function,

- for $x_0 = 1$,
  - if $d\,x > 0$ (positive), meaning $x_0 + h > x_0$, the value of $f(x_0 + h) > f(x_0)$
    • the value of $f(x)$ **increases** if we increase x_0.
  - if $d\,x < 0$ (negative), meaning $x_0 + h < x_0$, the value of $f(x_0 + h) < f(x_0)$
    • the value of $f(x)$ **decreases** if we decrease x_0.
- for $x_0 = -1$,
  - if $d\,x > 0$ (positive), meaning $x_0 + h > x_0$, the value of $f(x_0 + h) < f(x_0)$
    • the value of $f(x)$ **decreases** if we decrease x_0.
  - if $d\,x < 0$ (negative), meaning $x_0 + h < x_0$, the value of $f(x_0 + h) > f(x_0)$
    • the value of $f(x)$ **increases** if we increase x_0.

From these simple analysis, we can tell that we if we multiple the gradient by -1, we will always guarantee that we will point to a lower value of the function $f(x)$.

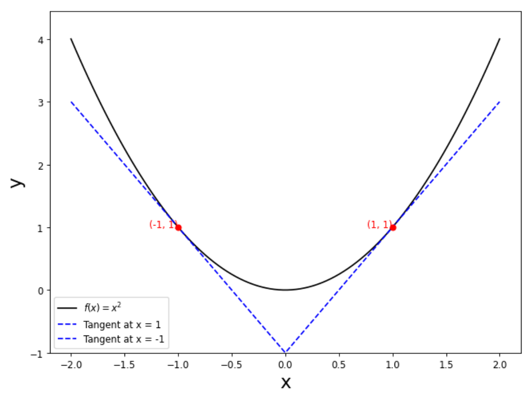

Figure 2: Gradient of $f(x) = x^2$, and its derivative $\frac{d\,f(x)}{d\,x} = 2x$ at two different points.

2 Gradient Descent for NonLinear Equations

Gradient Descent algorithms are not only used for linear regression, they are some of the most common frameworks to find the minima of nonlinear functions. As an example first, let's consider the use of gradient descent in physical chemistry. We start with the following chemical reaction,

$$\mathrm{N_2(g) + 3\,H_2(g) \longrightarrow 2\,NH_3(g)}$$

where the equilibrium constant is $K = 4.34\times10^{-3}$. If we follow the ICE tables methodologies where an x amount is formed, we get the following equilibrium equation,

$$Q = \frac{(2x)^2}{(1-x)(1-3x)^3} \tag{5}$$

The **goal** is to find the value of x at equilibrium, $f(x) = \ln Q$ is equal to $\ln K$,

$$\ln K = \underbrace{2\ln(2x)^2 - \ln(1-x) - 3\ln(1-3x)}_{f(x)} \tag{6}$$

From Fig. 3a), we can observe that there is only one value of x where $f(x) = \ln K$; meaning there is only one point where $f(x)$ intercepts $\ln K$. Finding this point is not trivial, however, we can **construct** an error function that will allow us to search for it. One way is to construct an error function using Eq. 6,

$$\mathcal{L}(x) = \left(f(x) - \ln K\right)^2. \tag{7}$$

We know that this equation is minima when $f(x) = \ln K$, and positive else where $\mathcal{L}(x) > 0$. This "cooked" error function "looks" quadratic, but is not, as we can see from Fig. 3. Here, we will use gradient descent to search for the minima of $\mathcal{L}(x)$. To do so, we require to compute the gradient,

$$\begin{aligned}\frac{d\;\mathcal{L}(x)}{d\,x} &= 2\left(f(x) - \ln K\right)\left(\frac{d\,f(x)}{d\,x}\right) = 2\left(f(x) - \ln K\right)\left(\frac{d\,f(x)}{d\,x}\right) \\ &= 2\left(f(x) - \ln K\right)\left(\frac{2}{x} + \frac{1}{(1-x)} + \frac{9}{(1-3x)}\right).\end{aligned} \tag{8}$$

Combining this gradient of $\mathcal{L}(x)$ and the gradient descent step, Eq. 4, we can iteratively search for the value of x where $f(x) = \ln K$. The gradient descent trajectory is displayed in Fig. 3(b), where after approximately 10 steps, we reach $x = 0.03092$, the value where $\mathcal{L}(x)$ is minimum.

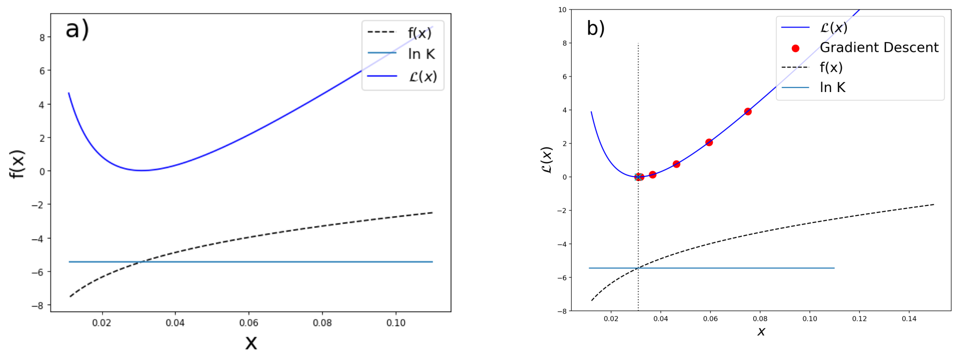

Figure 3: $f(x)$ is given by Eq. 6, and $K = 4.34 \times 10^{-3}$. $\mathcal{L}(x)$ is defined in Eq. 9. The The ● represent each gradient descent step towards finding the minimum of $\mathcal{L}(x)$.

3 Gradient Descent for NonLinear Multivariate Equations

In the previous sections we saw how gradient descent can be used to search for the lowest possible value of a function, these cases were simple ones, so in this section we will revise it for nonlinear multivariate-equations. Again, we will consider a multi-set of chemical reactions, and using the equilibrium condition seek the value of the concentrations when we reach chemical equilibrium.

$$\mathrm{H_2C_2O_4(aq) + H_2O} \rightarrow \mathrm{H_3O^+(aq) + HC_2O_4{}^-(aq)},\ K_{a1} = 5.6 \times 10^{-2}$$
$$\mathrm{HC_2O_4{}^-(aq) + H_2O} \rightarrow \mathrm{H_3O^+(aq) + C_2O_{42}{}^-(aq)},\ K_{a2} = 1.5 \times 10^{-4}$$
$$2\,\mathrm{H_2O} \rightarrow \mathrm{H_3O^+(aq) + OH^-(aq)},\ K_w = 10^{-14}$$

Each equilibrium can be rewritten in terms of the following variables,

$$m_{\mathrm{H_2C_2O_4}} = m^0 - x_{a1},\ \ m_{\mathrm{HC_2O_4{}^-}} = x_{a1} - x_{a2},\ \ m_{C2O42-} = x_{a2},\ \ m_{H3O+} = x_{a1} + x_{a2} + x_w,\ \ m_{\mathrm{OH^-}} = x_w$$

where m^0 is the initial concentration of $\mathrm{H_2C_2O_4}$. Using these variables we can derive the following chemical equilibrium conditions,

$$K_{a1} = \frac{(x_{a1} - x_{a2})(x_{a1} + x_{a2} + x_w)}{m^0 - x_{a1}} \tag{9}$$
$$K_{a2} = \frac{x_{a2}(x_{a1} + x_{a2} + x_w)}{x_{a1} + x_{a2}} \tag{10}$$
$$K_w = x_w(x_{a1} + x_{a2} + x_w) \tag{11}$$

For each of these equations, we can equate the right hand side with the left hand side and reach the following residual equations ($g_i(x_{a1}, x_{a2}, x_w)$), which are a function of the three varaibles,

$$g_{a1}(x_{a1}, x_{a2}, x_w) = K_{a1}(m^0 - x_{a1}) - (x_{a1} - x_{a2})(x_{a1} + x_{a2} + x_w) \tag{12}$$
$$g_{a2}(x_{a1}, x_{a2}, x_w) = K_{a2}(x_{a1} + x_{a2}) - x_{a2}(x_{a1} + x_{a2} + x_w) \tag{13}$$
$$g_w(x_{a1}, x_{a2}, x_w) = K_w - x_w(x_{a1} + x_{a2} + x_w) \tag{14}$$

Using the vector notation, we will create a vector $\mathbf{G}$ that contains these three residual equations,

$$\mathbf{G}(x_{a1}, x_{a2}, x_w) = \begin{bmatrix} g_{a1}(x_{a1}, x_{a2}, x_w) \\ g_{a2}(x_{a1}, x_{a2}, x_w) \\ g_w(x_{a1}, x_{a2}, x_w) \end{bmatrix} = \begin{bmatrix} K_{a1}(m^0 - x_{a1}) - (x_{a1} - x_{a2})(x_{a1} + x_{a2} + x_w) \\ K_{a2}(x_{a1} + x_{a2}) - x_{a2}(x_{a1} + x_{a2} + x_w) \\ K_w - x_w(x_{a1} + x_{a2} + x_w) \end{bmatrix}, \tag{15}$$

when each element of $\mathbf{G}(x_{a1}, x_{a2}, x_w)$ is equal to 0, we found the concentration at equilibrium.

Similarly to the previous section, we can "create" an error function that when it is 0 in all three residual functions, it means we found the value of the concentrations at equilibrium. One easy error function is,

$$\begin{aligned} \mathcal{L}(x_{a1}, x_{a2}, x_w) &= \frac{1}{2}\mathbf{G}(x_{a1}, x_{a2}, x_w)^\top \mathbf{G}(x_{a1}, x_{a2}, x_w) \\ &= \frac{1}{2}\left(g_{a1}^2(x_{a1}, x_{a2}, x_w) + g_{a2}^2(x_{a1}, x_{a2}, x_w) + g_w^2(x_{a1}, x_{a2}, x_w)\right). \end{aligned} \tag{16}$$

The next step is to construct the Jacobian of $\mathcal{L}(x_{a1}, x_{a2}, x_w)$,

$$\nabla \mathcal{L}(x_{a1}, x_{a2}, x_w) = \begin{bmatrix} \frac{\partial \mathcal{L}(x_{a1}, x_{a2}, x_w)}{\partial x_{a1}} \\ \frac{\partial \mathcal{L}(x_{a1}, x_{a2}, x_w)}{\partial x_{a2}} \\ \frac{\partial \mathcal{L}(x_{a1}, x_{a2}, x_w)}{\partial x_w} \end{bmatrix}. \tag{17}$$

We can compute these individual gradients by using the chain rule but instead, we are going to take advantage of the vector notation in Eq. 17. As we you can see $\mathbf{G}(x_{a1}, x_{a2}, x_w)$ is a vector function that depends on the three variables, we can compute, let us define $\mathbf{x}^\top = [x_{a1}, x_{a2}, x_w]$, to ease the notation. To run gradient descent, we require the following gradient,

$$\nabla \mathcal{L}(x_{a1}, x_{a2}, x_w) = \frac{\partial \mathcal{L}(x_{a1}, x_{a2}, x_w)}{\partial \mathbf{x}} = \frac{1}{2}\frac{\partial}{\partial \mathbf{x}}\mathbf{G}(x_{a1}, x_{a2}, x_w)^\top \mathbf{G}(x_{a1}, x_{a2}, x_w), \tag{18}$$

we can map the above vector derivative to $\frac{\partial g_1(x) g_2(x)}{\partial x} = \left(\frac{\partial g_1(x)}{\partial x}\right) g_2(x) + g_1(x)\left(\frac{\partial g_2(x)}{\partial x}\right)$, giving us[1],

$$\frac{\partial \mathcal{L}(x_{a1}, x_{a2}, x_w)}{\partial \mathbf{x}} = \mathbf{J}_{\mathbf{G}}(x_{a1}, x_{a2}, x_w)^\top \mathbf{G}(x_{a1}, x_{a2}, x_w), \tag{19}$$

where $\mathbf{J}_{\mathbf{G}}(x_{a1}, x_{a2}, x_w)$ is the Jacobian matrix of the vector $\mathbf{G}(x_{a1}, x_{a2}, x_w)$,

$$\begin{aligned} \mathbf{J}_{\mathbf{G}}(x_{a1}, x_{a2}, x_w) &= \begin{bmatrix} \frac{\partial g_1}{\partial x_{a1}} & \frac{\partial g_1}{\partial x_{a2}} & \frac{\partial g_1}{\partial x_w} \\ \frac{\partial g_2}{\partial x_{a1}} & \frac{\partial g_2}{\partial x_{a2}} & \frac{\partial g_2}{\partial x_w} \\ \frac{\partial g_w}{\partial x_{a1}} & \frac{\partial g_w}{\partial x_{a2}} & \frac{\partial g_w}{\partial x_w} \end{bmatrix} \\ &= \begin{bmatrix} -K_{a1} - 2x_{a1} - x_w & 2x_{a2} + x_w & -x_{a1} + x_{a2} \\ K_{a2} - x_{a2} & -K_{a2} + 2x_{a2} - x_{a1} - x_w & -x_{a2} \\ -x_w & -x_w & -x_{a1} - x_{a2} - 2x_w \end{bmatrix}. \end{aligned} \tag{20}$$

Finally we can combine Eqs. 18 and 20, to fully construct the Jacobian of $\mathcal{L}(x_{a1}, x_{a2}, x_w)$ (Eq. 19) required for gradient descent,

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} - \eta \nabla \mathcal{L}(\mathbf{x})\big|_{\mathbf{x}_{\text{old}}} \tag{21}$$

Fig. 4 displays the gradient descent trajectory, and in the right panel we show the value of $\mathcal{L}(x_{a1}, x_{a2}, x_w)$ at each iteration of the search, and the norm of the $\nabla \mathcal{L}(x_{a1}, x_{a2}, x_w)$. As we can see, at the end of the search the norm of this gradient is approximately zero ($\|\nabla \mathcal{L}(x_{a1}, x_{a2}, x_w)\| \approx 10^{-14}$).

[1]Eq. (93) from the Matrix CookBook

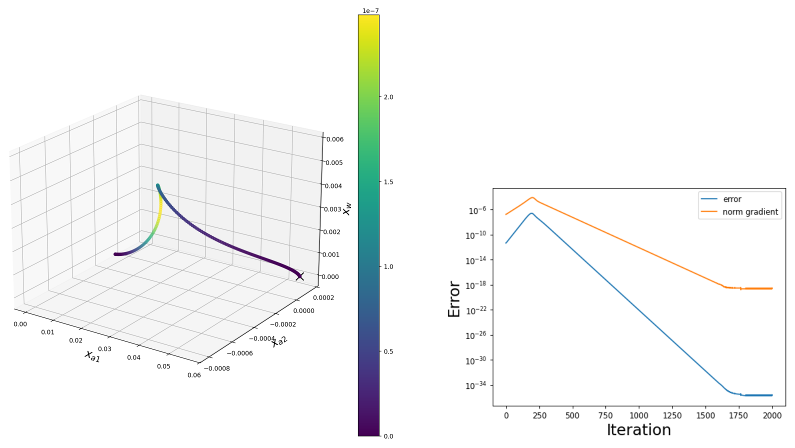

Figure 4: (left panel) Gradient descent trajectory to minimize $\mathcal{L}(x_{a1}, x_{a2}, x_w)$ (Eq. 17), the color indicates the value of $\mathcal{L}$ and $\times$ the final value fond by gradient descent. (right panel) The value of $\mathcal{L}(x_{a1}, x_{a2}, x_w)$ and the norm of $\nabla\,\mathcal{L}(x_{a1}, x_{a2}, x_w)$ at each iteration.

4 Newton's Method for Nonlinear Equations

In the previous section we saw that gradient descent can be used to solve non-linear equations if we combine it with an error function. This strategy is not as common, as there are some other methods that could find the minima of a function with less iterations. Here, we will also study **Newton's Method**, which also uses gradients to solve non-linear equations. First we will start with a one-dimensional system to present this new method and then extended it to multi-variate problems

4.1 One-dimensional systems

In Newton's method, we rely on a local expansion of the function $f(x)$ around a points x_0 using the Taylor expansion,

$$y(x) = \underbrace{f(x_0)}_{\text{zero-order term}} + \underbrace{\left.\frac{d\,f(x)}{d\,x}\right|_{x_0}(x - x_0)}_{\text{first-order term}} + \underbrace{\cdots}_{\text{higher-order derivative terms}} \tag{22}$$

where $y(x)$ is the approximation of the function $f(x)$ at x_0.

To make this discussion more chemistry guided let's think about the ammonia reaction from the prev. section, Eq. 5. We aim to solve the same equation, $f(x) = \ln K$. Using the Taylor expansion notation at first order we get,

$$\ln K = f(x_0) + \left.\frac{d\,f(x)}{d\,x}\right|_{x_0}(x - x_0), \tag{23}$$

where $f(x)$ is given in Eq. 6. The only unknown from this equation is the value of x that satisfies this expansion. If we solve for x, we get,

$$x = x_0 - \left(\left.\frac{d\,f(x)}{d\,x}\right|_{x_0}\right)^{-1}(f(x) - \ln K), \tag{24}$$

x represents the point at where the $\left.\frac{d\,f(x)}{d\,x}\right|_{x_0} = 0$.

Similar to gradient descent, Newton's method step function, Eq. 24, will move us each iteration closer to the value where $f(x) = \ln K$. I am not good at doing animations but the Wikipedia's page on Newton's method has an animation that clearly explains each iteration of Newton's method, Fig. 5.

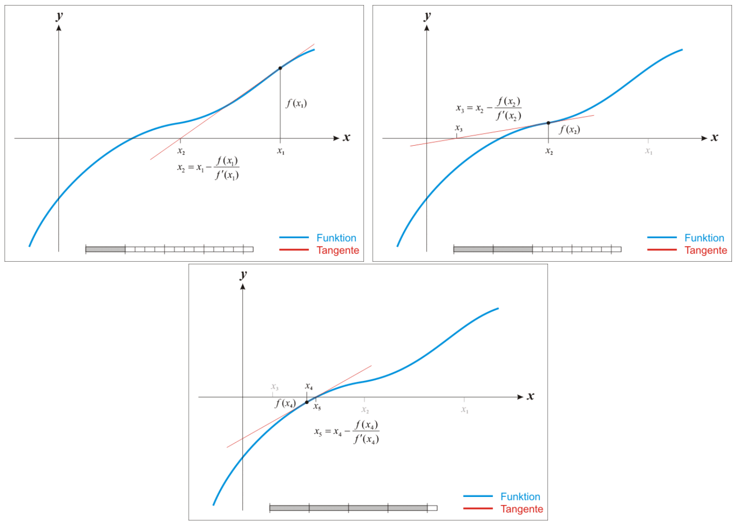
y
x
$f(x_1)$
$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$
$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$
$f(x_2)$
$f(x_4)$
$x_5 = x_4 - \frac{f(x_4)}{f'(x_4)}$
Funktion
Tangente

Figure 5: Newton's method to search for $f(x) = 0$. Figures were taken from Wikipedia's page.

For this chemical example, the trajectory search using Newton's method is illustrated in Fig. 6. As we can observe, this trajectory search is significantly different compare to the one using gradient descent, Fig. 3. Gradient descent does local searches, "rolling down the hill", while Newton's method looks more "random" or with "bigger steps".

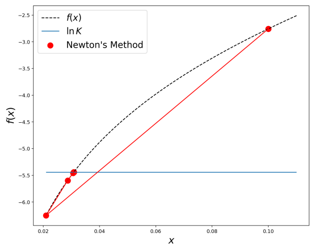

Figure 6: Newton's method to search for $f(x) = \ln K$.

4.2 Multi-dimensional nonlinear equations

Here, we will extend Newton's method for vectorial functions,

$$\boldsymbol{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_3(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \ln K_1 \\ \ln K_2 \\ \vdots \\ \ln K_n \end{bmatrix}, \tag{25}$$

where each equation for example is an individual chemical equilibrium.

To derive Eq. 24 for vectorial functions, we need to do the Taylor expansion for $\boldsymbol{f}(\mathbf{x})$ around a point $\mathbf{x}_0$,

$$\boldsymbol{y}(\mathbf{x}) = \underbrace{\boldsymbol{f}(\mathbf{x}_0)}_{\text{zero-order term}} + \underbrace{(\mathbf{x}-\mathbf{x}_0)^\top \nabla \boldsymbol{f}(\mathbf{x})\Big|_{\mathbf{x}_0}}_{\text{first-order term}} + \underbrace{\cdots}_{\text{higher-order terms}} \tag{26}$$

Following the same procedure we did for the one-dimensional system, we need to solve for the value of $\mathbf{x}$. The solution for this multi-dimensional system gives us the following equation for each iteration for the Newton's method,

$$\mathbf{x} = \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}_{\text{old}}} - J^{-1}_{\mathbf{f}(\mathbf{x})} \begin{bmatrix} f_1(\mathbf{x}_n) - \ln K_1 \\ f_2(\mathbf{x}_n) - \ln K_2 \\ \vdots \\ f_n(\mathbf{x}_n) - \ln K_n \end{bmatrix}, \tag{27}$$

where $J_{\mathbf{f}(\mathbf{x})}$ is the Jacobian of Eq. 25, given by,

$$J_{\mathbf{f}(\mathbf{x})} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_m} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_m} \\ \vdots & & \cdots & \vdots \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_m} \end{bmatrix} \tag{28}$$

$J_{\mathbf{f}(\mathbf{x})}$ is a matrix with $n-$rows, as $\boldsymbol{f}(\mathbf{x})$ has n functions, and $m-$columns, assuming $\mathbf{x}^\top = [x_1, x_2, \cdots, x_m]$.

Compared to gradient descent where we only require the computation of $J_{\mathbf{f}(\mathbf{x})}$, Newton's method requires the inversion of this Jacobina, adding an additional cost to the over all procedure. For small systems, $J^{-1}_{\mathbf{f}(\mathbf{x})}$ is feasible, but as we consider larger problems the inversion of the Jacobian becomes unfeasible.

For the three equilibrium systems that we previously presented, we can defined the following functions,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} K_{a1}(m_0 - x_{a1}) - (x_{a1} - x_{a2})(x_{a1} + x_{a2} + x_w) \\ K_{a2}(x_{a1} - x_{a2}) - (x_{a2})(x_{a1} + x_{a2} + x_w) \\ K_w - (x_w)(x_{a1} + x_{a2} + x_w) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{29}$$

and the Jacobian of $\mathbf{f}$ is given in Eq. 21. Fig. 7 illustrates the search trajectory using Newton's method, and as we can observe, the number of evaluations are fewer compared to gradient descent.

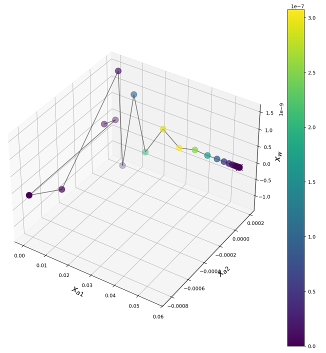

Figure 7: Newton's method search trajectory.