Introduction to Electric Power Systems
Lecture 11
Power Flow

Keith Moffat

Contents

1 Power Flow Equations

1.1 Complex-Valued Power Flow for a Single Branch

The vector of complex-valued voltages V is defined as the "state" of the system. V and Y together define the current and power on every branch of the system according to the equations:

$$I_{ik} = y_{ik}(V_i - V_k) \tag{1}$$
$$S_{ik} = (V_i - V_k)I_{ik}^* = |V_i - V_k|^2 y_{ik} \tag{2}$$

Note that the linear circuit elements (resistors, inductors, capacitors) that are used to model the grid and constitute the entries of Y are linear in the relationship between voltage and current (Eqn. 1), *not* in the relationship between voltage and power (Eqn. 2). However, recall from the first week's discussion that in power systems we generally work with power injections/extractions from the network because unlike current, power is conserved across physical domains. So a current injection vector I is not available.

Once you know the voltages on the network, it is straightforward to compute the power flows and injections on the network. What is challenging is solving the power flow equations the *other* way—determining the voltages if you know the power injections.

1.2 Complex-Valued Power Flow for the Full Network

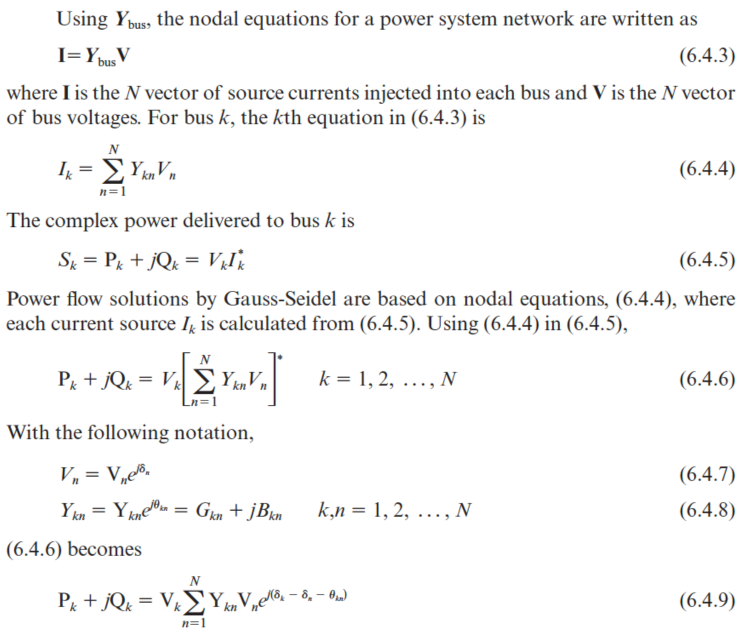
Equations (1) and (2) describe the current and power flowing on a single network branch. For an entire network, the power flow equations can be written in a single (vector-valued) equation using the bus admittance matrix Y. Defining $\mathrm{diag}(V)$ as a matrix with vector V in the diagonal entries and zeros elsewhere, the nodal power injections S are given by:

$$\begin{aligned} S &= \mathrm{diag}(V) I^* \\ &= \mathrm{diag}(V)(YV)^* \end{aligned} \tag{3}$$

Once again, these equations are nonlinear because of the VV^* multiplication.

1.3 Real-Valued Power Flow (for a Single Node Attached to Many Branches)

Other forms of the power flow equations exist. Often, we prefer real-valued equations. One version of the real-valued power flow equations are derived on page 350 of GOS:



Using $\boldsymbol{Y}_{\text{bus}}$, the nodal equations for a power system network are written as

$$\mathbf{I} = \boldsymbol{Y}_{\text{bus}}\mathbf{V} \tag{6.4.3}$$

where $\mathbf{I}$ is the N vector of source currents injected into each bus and $\mathbf{V}$ is the N vector of bus voltages. For bus k, the kth equation in (6.4.3) is

$$I_k = \sum_{n=1}^{N} Y_{kn} V_n \tag{6.4.4}$$

The complex power delivered to bus k is

$$S_k = \mathrm{P}_k + j\mathrm{Q}_k = V_k I_k^* \tag{6.4.5}$$

Power flow solutions by Gauss-Seidel are based on nodal equations, (6.4.4), where each current source I_k is calculated from (6.4.5). Using (6.4.4) in (6.4.5),

$$\mathrm{P}_k + j\mathrm{Q}_k = V_k\left[\sum_{n=1}^{N} Y_{kn} V_n\right]^* \qquad k = 1, 2, \ldots, N \tag{6.4.6}$$

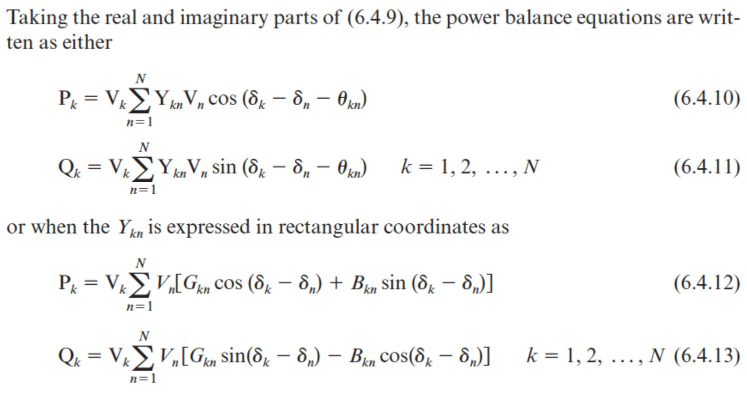
With the following notation,

$$V_n = \mathrm{V}_n e^{j\delta_n} \tag{6.4.7}$$

$$Y_{kn} = \mathrm{Y}_{kn} e^{j\theta_{kn}} = G_{kn} + jB_{kn} \qquad k,n = 1, 2, \ldots, N \tag{6.4.8}$$

(6.4.6) becomes

$$\mathrm{P}_k + j\mathrm{Q}_k = \mathrm{V}_k \sum_{n=1}^{N} \mathrm{Y}_{kn} \mathrm{V}_n e^{j(\delta_k - \delta_n - \theta_{kn})} \tag{6.4.9}$$



Taking the real and imaginary parts of (6.4.9), the power balance equations are written as either

$$\mathrm{P}_k = \mathrm{V}_k \sum_{n=1}^{N} \mathrm{Y}_{kn} \mathrm{V}_n \cos(\delta_k - \delta_n - \theta_{kn}) \tag{6.4.10}$$

$$\mathrm{Q}_k = \mathrm{V}_k \sum_{n=1}^{N} \mathrm{Y}_{kn} \mathrm{V}_n \sin(\delta_k - \delta_n - \theta_{kn}) \qquad k = 1, 2, \ldots, N \tag{6.4.11}$$

or when the Y_{kn} is expressed in rectangular coordinates as

$$\mathrm{P}_k = \mathrm{V}_k \sum_{n=1}^{N} V_n [G_{kn}\cos(\delta_k - \delta_n) + B_{kn}\sin(\delta_k - \delta_n)] \tag{6.4.12}$$

$$\mathrm{Q}_k = \mathrm{V}_k \sum_{n=1}^{N} V_n [G_{kn}\sin(\delta_k - \delta_n) - B_{kn}\cos(\delta_k - \delta_n)] \qquad k = 1, 2, \ldots, N \tag{6.4.13}$$

This version of the real valued power flow equations relates the voltages on the network in polar coordinates to the power injections at each bus in rectangular coordinates:

$$\begin{aligned} P_i = P_{Gi} - P_{Di} &= \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik}) \\ Q_i = Q_{Gi} - Q_{Di} &= \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik}) \end{aligned} \tag{4}$$

Both (3) and (4) encode the same information about the relationship between the power injections and the voltages on the network, they just do so in different coordinates.
Note: Equation (4) has an important detail that can be overlooked: the G_{ik}s and B_{ik}s in Eqn. (4) are the real and imaginary portions of the entries of Y, respectively. Thus, G_{ik} and B_{ik} come from the Y matrix: When $i \neq k$, $G_{ik} = -\text{Real}(y_{ik})$ and $B_{ik} = -\text{Imag}(y_{ik})$. When $i = k$, $G_{ii} = \text{Real}(y_i + \sum_{l \neq i} y_{il})$ and $B_{ii} = \text{Imag}(y_i + \sum_{l \neq i} y_{il})$, *not* just the real and imaginary parts of the shunt admittance of node i.

Q. *Why would we want rectangular coordinates for power, and polar coordinates for voltage?*

---

2 Power Flow Problem Setup

At a high level, the goal of the power flow problem (sometimes also referred to as "load flow") is to determine the voltages on the network, given nodal power injections (positive for generation, negative for consumption). Power flow is ubiquitous in power system planning and control.
The voltages on the network define the state of the network. The voltages are important for two reasons.

1. Each bus has max and min magnitude limits determined by operating standards.
2. Once the voltages for each bus are known, determining any power flow on any line in the network is straightforward.

More specifically, the power flow problem is to determine the voltages on the network, given the following information for each bus:

- Loads: P and Q
- Generators: P and Q, or P and $|V|$
- Slack bus: $|V| = 1$ and $\delta = 0$

The slack bus serves two purposes in power flow:

1. It injects the necessary real and reactive power so that the power balance is met for the network, including network losses.
2. It serves as the phasor angle reference.

Determining which bus serves as the slack bus in power flow analysis is often a design decision because rarely does a single node satisfy the network power imbalance in reality.[1]

---

[1] One exception to this paradigm is the substation node in distribution networks, which behaves like an infinite bus for the distribution network if the transmission line impedances are negligible compared with the distribution line impedances.

3 Slack Bus

The slack bus adjusts its generation so that the real and reactive power on the network are balanced, taking into account the losses on the network. It is necessary to select one node as the slack bus to get power flow to converge.

Q. *Why do we need a slack bus for power flow to converge?*

For transmission networks, often, all of the generators participate in balancing real and reactive power. So selecting one node as the slack bus is an approximation of the real problem. That approximation is less egregious if you select the generator that does most of the power-balancing (presumably a large generator). For distribution networks, which are fed power from the transmission network through the substation transformer, the substation is a good selection for the slack bus. Details on the slack bus are given in Appendix A.

4 Linear vs. Nonlinear Systems of Equations

A system of linear equations is a special type of a system of (nonlinear) equations that can be expressed in the following form:

$$b = Ax \tag{5}$$

Systems of linear equations have a number of desirable properties:

- It is easy to determine when there is a solution.
- It is easy to find a solution.
- It is easy to determine when the solution is unique. If it is not unique, it is easy to determine where the other solutions are.

Systems of nonlinear equations, on the other hand, in general do not share these properties. For systems of nonlinear equations:

- It is not easy to determine when there is a solution.
- It is not easy to find a solution.
- It is not easy to determine when the solution is unique or where the other solutions might be.

Power flow is a system of nonlinear equations.

5 Newton's Method

Today, we are used to having computers solve things for us. But some problems, due to their size (dimension) or complexity, are hard or intractable even for computers. Because of it's nonlinear structure and size, power flow is one such problem. The power flow problem is a system of nonlinear equations. That system can be compactly represented as $b = g(x)$, where b and x are vectors, and g is a set of equations. Any system of equations can be formulated as trying to find the zero crossings, i.e. by subtracting b from both sides:

$$f(x) = g(x) - b = 0 \tag{6}$$

where 0 is a vector of zeros. The system of nonlinear equations can have zero, one, or many solutions.

The iteration method is the primary (and sometimes only) tool used to solve hard nonlinear problems such as power flow. In the iteration method, we take a guess at the solution at iteration i, $x^{(i)}$, and observe how close $f(x^{(i)})$ is to 0. Then, using some information from $f(x)$, we determine the guess for the next iteration. Two commonly used iterative methods are Newton's method, and Gauss-Seidel.

Isaac Newton originally proposed Newton's method (or a precursor for Netwon's Method, which is also called Newton-Rhapson) as a technique for finding the roots of a polynomial. Newton's method applies to non-polynomial equations as well, such as the power flow problem. In Newton's method, each iteration's $x^{(i)}$ is determined by approximating $f(x)$ by the first order Taylor expansion. For a one-dimensional x, the first order Taylor expansion of $f(x)$ at $x^{(i)}$ is:

$$f(x^{(i+1)}) \approx f(x^{(i)}) + (x^{(i+1)} - x^{(i)}) \frac{\partial f}{\partial x}|_{x^{(i)}} \tag{7}$$

where $\frac{\partial f}{\partial x}|_{x^{(i)}}$ is the derivative of f with respect to x, evaluated at $x^{(i)}$. We want to find an $x^{(i)}$ such that $f(x^{(i+1)}) = 0$. Setting $f(x^{(i)}) + (x^{(i+1)} - x^{(i)}) \frac{\partial f}{\partial x}|_{x^{(i)}} = 0$ and solving for $x^{(i+1)}$, we get the Newton iteration:

$$x^{(i+1)} = x^{(i)} - \left(\frac{\partial f}{\partial x}|_{x^{(i)}}\right)^{-1} f(x^{(i)}) \tag{8}$$

Because f is not exactly represented by (7), the $x^{(i+1)}$ from (8) will likely not make $f(x^{(i+1)}) = 0$ exactly. However it will likely be a step in the right direction. The hope, which is often the case, is that if we take many steps in the right direction, then eventually $f(x^{(i+1)})$ will be (close to) 0.

For multiple dimension Newton's method, in which x is a vector and f is a set of equations, the derivative $\frac{\partial f}{\partial x}|_{x^{(i)}}$ is replaced by the Jacobian of f with respect to x, evaluated at $x^{(i)}$: $J(f(x))|_{x^{(i)}}$. This process continues until $f(x^{(i+1)})$ is sufficiently small, or the iteration counter reaches a maximum number of iterations.

Q. *Draw a one-dimensional nonlinear equation that has two zeros. What will Newton's method do for this equation?*

Q. *Draw a one-dimensional nonlinear equation that has one zero. What will Newton's method do for this equation?*

Q. *Draw a one-dimensional nonlinear equation that has no zeros. What will Newton's method do for this equation?*

Q. *How many iterations does Newton's method require for linear systems?*

---

6 Newton's Method For Power Flow

For the power flow problem, we have the system of nonlinear equations which include the power injections at every bus, described by (4). For these equations, x is the stacked vector of voltage angles and magnitudes: $x = [\delta; |V|]$. We apply Newton's Method to the nonlinear system of equations f, which is the difference between $g(x)$, the stacked vector of the real and reactive power injections determined by (4), and the assigned power injections:

$$f(x) = g(x) - [P; Q]. \tag{9}$$

We have found a solution to $f(x)$ when (9) is equal to zero.

Q. *In an N bus system with m P, $|V|$ generator buses (as opposed to P, Q generator buses), how many variables are there to solve for in the power flow problem?*

Recall the power flow problem for just two buses. This problem can have zero, one or two voltage solutions. With three buses there can be between zero and four solutions. In general the number of solutions is upper bounded exponentially by the number of nodes N: (number of solutions) $< 2^{N-1}$. Finding the high voltage solution(s) can be challenging, and is dependent on the initial guess. If a solution is found, it is possible that the solution is a low voltage solution (recall the two bus example), and not actually useful for the network. If this occurs, another initial guess must be tried.

For a large network with thousands of nodes, it is easy to see how this problem becomes intimidating. But the same intuitions from the two node example applies: when there are multiple solutions, the initial guess can impact the solution produced by Newton's method.

It is also possible that there are no solutions, as we saw for general nonlinear equations in Section 5. When there are no solutions, Newton's Method will not converge no matter what initial guess is used.

Q. *What does it mean when the Newton method for power flow doesn't converge?*

Hint: In what scenarios might power flow not converge?
Without additional information, is it possible to distinguish between these scenarios?

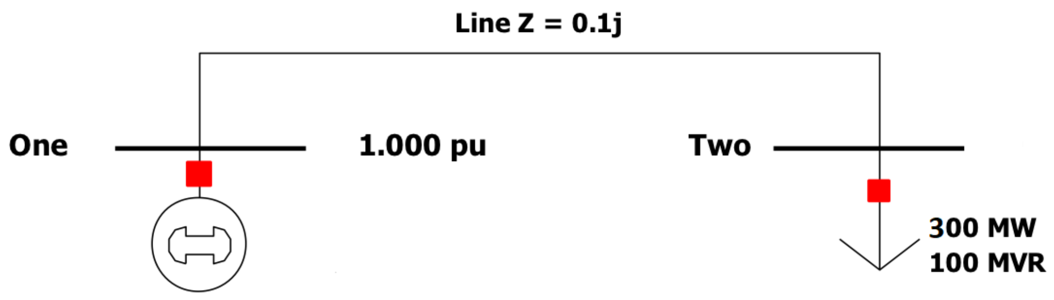
Near common operating states, grid operators are familiar with how power flow behave, and have good initial guesses that tend to converge to useful operating states. Operating the power system in an unfamiliar operating state, on the other hand, can be very challenging. The operators may not have good initial guesses for the power flow problem. This can be major challenge after a blackout or during a contingency because system operators have to/should demonstrate that a given control action will not crash the system. That is, system operators have to/should demonstrate that power flow converges before they make the decision to change the power balance (e.g. turn a generator on or off). Finding an initial state that results in power flow converging in an unfamiliar operating regime can be "maddeningly hard" [Stott, DC Power Flow Revisited, 2009].

When learning power flow, it is important to run Newton's method (and/or Gauss-Seidel) by hand at least once to get intuition for how the algorithm works. In practice, though, a power system engineer would never execute iterative power flow by hand, and probably not have to code up an iterative algorithm either because commercial power flow solvers exist. It is, however, important to understand what is going on under the hood so you can know what to expect. Specifically, it is important to understand:

- Why iterative solution methods are used
- Why a slack bus is needed
- That power flow can have multiple solutions (i.e. high and low voltage solutions), or no solution
- That the initial guess affects power flow convergence
  - The initial guess will determine which power flow solution the iterative method will converge to
  - For some initial guesses iterative power flow will not converge, even though the power injections are feasible. It is not easy to differentiate this circumstance from the circumstance in which the power injections are not feasible

7 2x2 Newton Example

Consider the two-bus example illustrated in the diagram. Bus 1 is the slack bus. Given the impedance of the line, and given P and Q at Bus 2, use Newton's Method to find the voltage magnitude and angle at Bus 2. Just go through the first iteration to find the guess at the start of the second iteration. Assume a flat start.



Answer:

$$x = \begin{bmatrix} \delta_2 \\ |V_2| \end{bmatrix}$$

$$Y_{bus} = jB_{bus} = \begin{bmatrix} -j10 & j10 \\ j10 & -j10 \end{bmatrix}$$

First, we simplify the (4) equations by plugging in $|V_1| = 1$, $\delta_1 = 0$, $B_{ik} = 10$, $B_{ii} = -10$, and $G_{ik} = G_{ii} = 0$. This gives us a simplified equation for the real and reactive power injections at bus 2, in terms of our estimated values of $|V_2|$ and δ_2:

$$P_{2calc} = |V_2| 10 \sin \delta_2$$
$$Q_{2calc} = |V_2|(-10) \cos \delta_2 + 10|V_2|^2$$

Define the function:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} P_{2calc}(x) - P_2 \\ Q_{2calc}(x) - Q_2 \end{bmatrix} = \begin{bmatrix} 10|V_2|\sin\delta_2 + 3 \\ -10|V_2|\cos\delta_2 + 10|V_2|^2 + 1 \end{bmatrix}$$

to be the mismatch between the real and reactive powers we've calculated with our estimated $|V_2|$, δ_2 values, and the known true values of those real and reactive powers. Note the injections are negative, so 3 and 1 are added. Our goal is to find the roots of $f(x)$. To find the roots using the Newton method, we define the iteration:

$$x^{(i+1)} = x^{(i)} - J^{-1}(f(x))|_{x^{(i)}} f(x^{(i)}) \tag{10}$$

The Jacobian for this two bus system is defined as:

$$J(f(x))|_{x^{(i)}} = \begin{bmatrix} \frac{f_1(x)}{d_{\delta_2}} & \frac{f_1(x)}{d_{|V_2|}} \\ \frac{f_2(x)}{d_{\delta_2}} & \frac{f_2(x)}{d_{|V_2|}} \end{bmatrix}_{x^{(i)}} = \begin{bmatrix} 10|V_2|\cos\delta_2 & 10\sin\delta_2 \\ 10|V_2|\sin\delta_2 & -10\cos\delta_2 + 20|V_2| \end{bmatrix}_{x^{(i)}}$$

First Iteration:

We have our initial guess, $x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and we can calculate $f(x(0)) = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$. To get the inverse Jacobian we evaluate the Jacobian of $f(x)$ at $x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and then take the inverse:

$$J(f(x))^{-1}|_{x(0)} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Plugging $x(0)$, $J(f(x))^{-1}|_{x(0)}$ and $f(x(0))$ into (10):

$$x(1) = x(0) - J^{-1}(f(x))|_{x(0)} f(x(0)) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$$

Second Iteration (Bonus):

From the previous iteration we have $x(1) = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$, and we can calculate $f(x(1)) = \begin{bmatrix} -2.7 + 3 \\ -.5 + 1 \end{bmatrix} = \begin{bmatrix} .3 \\ .5 \end{bmatrix}$. Using the Jacobian formula from above we get

$$J(f(x))|_{x(1)} = \begin{bmatrix} 8.6 & -2.96 \\ -2.66 & 8.45 \end{bmatrix}$$

and taking the inverse:

$$J(f(x))^{-1}|_{x(1)} = \begin{bmatrix} 0.13 & 0.046 \\ 0.041 & 0.13 \end{bmatrix}$$

Plugging $x(1)$, $J(f(x))^{-1}|_{x(1)}$ and $f(x(1))$ into (10):

$$x(2) = x(1) - J^{-1}(f(x))|_{x(1)} f(x(1)) = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix} - \begin{bmatrix} 0.13 & 0.046 \\ 0.041 & 0.13 \end{bmatrix} \begin{bmatrix} .34 \\ .5 \end{bmatrix} = \begin{bmatrix} -0.37 \\ 0.82 \end{bmatrix}$$

Plugging $x(2)$ into f gives: $f(x(2)) = \begin{bmatrix} -2.94 + 3 \\ -0.93 + 1 \end{bmatrix} = \begin{bmatrix} 0.058 \\ 0.067 \end{bmatrix}$.

This process continues until $f(x) < \epsilon$, or the algorithm gives up (reaches its max iteration count). For this problem, it appears to have done pretty well after just two iterations.

A Slack Bus Details

Technically, the power-balancing and voltage-reference roles can be split up between two different buses, though this is uncommon. When the slack bus performs/satisfies both of the aforementioned roles, then the slack bus can be thought of as an infinite bus.

The slack bus is a theoretical construct that ensures that the power flow equations have a solution. In practice, grids do not generally have a single node that serves the two slack bus roles: 1) balancing the power on the network, and 2) maintaining a constant voltage. Transmission networks have constant voltage magnitude nodes, but these nodes usually have a constant real power output as well. Thus, they do not satisfy the (real) power-balancing role. Usually, the real power balance for transmission networks is maintained in a distributed manner using droop control laws.

On the other hand, the substation node in distribution network analysis often does approximately satisfy the slack bus criteria. In order for the substation node to act as a slack bus, a number of conditions must be met: 1) there can be only one substation, 2) the transmission network upstream of the substation must be sufficiently high admittance so that the substation voltage does not depend on the power flowing from the transmission network through the substation node to the distribution network.

A *Bonus Material*: Newton's Method for Optimization

For those of you who have seen Newton's method in optimization, we are talking about the same Newton's method, but applied to solving systems of nonlinear equations rather than optimizing over a cost function. Newton's method in optimization actually finds (or tries to find) a zero of the gradient field, not the value field. So an optimization Newton step is a linear step in the gradient domain. When the adjusted gradient is applied to the value field, the step in the value field is a scaled and rotated gradient step. The Jacobian of the gradient field is called the Hessian, which is analogous to the second order derivative of a univariate function.