CHAPTER 8

CONSTRAINED OPTIMIZATION 2: SEQUENTIAL QUADRATIC PROGRAMMING, INTERIOR POINT AND GENERALIZED REDUCED GRADIENT METHODS

8.1 The Interior Point (IP) Algorithm

8.1.1 Problem Definition

In general, for the *primal-dual* method given here, there are two approaches to developing the equations for the IP method: putting slack variables into their own vector, and incorporating slack variables as part of the design variables **x**. Both methods can be found in the literature. We have opted for the former, because the development is somewhat more straightforward. The development of the latter is given in a later optional section.

For simplicity, we will start with a problem that only has inequality constraints.

$$\text{Min} \quad f(\mathbf{x}) \tag{8.1}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) \ge 0 \quad i = 1, \ldots, m \tag{8.2}$$

We will add in *slack* variables to turn the inequalities into equalities:

$$\text{Min} \quad f(\mathbf{x}) \tag{8.3}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) - s_i = 0 \quad i = 1, \ldots, m \tag{8.4}$$

$$s_i \ge 0 \quad i = 1, \ldots, m \tag{8.5}$$

Slack variables are so named because they take up the "slack" between the constraint value and the right hand side. For an inequality constraint to be feasible, s_i must be ≥ 0. If (8.4) is satisfied and $s_i = 0$, then the original inequality is binding.

The IP algorithm eliminates the lower bounds on **s** by incorporating a *barrier function* as part of the objective:

$$\text{Min} \quad f_\mu = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(s_i) \tag{8.6}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) - s_i = 0 \quad i = 1, \ldots, m \tag{8.7}$$

We note that as s_i approaches zero (from a positive value, i.e. from feasible space), the negative barrier term goes to infinity. This obviously penalizes the objective and forces the algorithm to keep s positive. The IP algorithm solves a sequence of these problems for a decreasing set of barrier parameters μ. As μ approaches zero, the barrier becomes steeper and sharper. This is illustrated in Fig. 8.9.

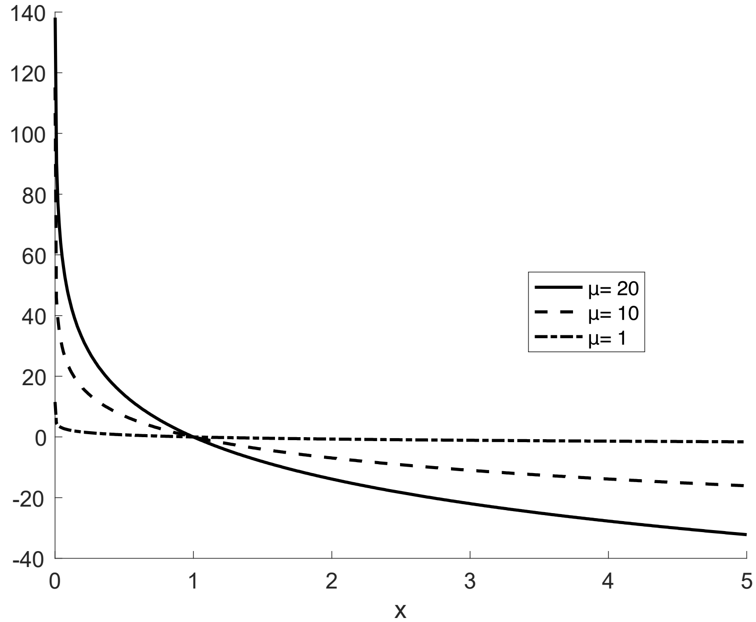

Fig. 8.9 Barrier function $-\mu \ln(x)$ for μ=20, 10 and 1.

We can define the Lagrangian function for this problem as,

$$L(\mathbf{x},\mathbf{s},\lambda) = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(s_i) - \sum_{i=1}^{m} \lambda_i (g_i(\mathbf{x}) - s_i) \tag{8.8}$$

Taking the gradient of this function with respect to $\mathbf{x}, \mathbf{s}, \lambda$, and setting it equal to zero gives us the KKT conditions for (8.6)–(8.7),

$$\nabla_x L = \nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) = \mathbf{0} \tag{8.9}$$

$$\nabla_s L = s_i \lambda_i - \mu = 0 \qquad i = 1, \ldots, m \tag{8.10}$$

$$\nabla_\lambda L = -(g_i(\mathbf{x}) - s_i) = 0 \qquad i = 1, \ldots, m \tag{8.11}$$

If we define **e** as the vector of 1's of *m* dimension and,

$$\mathbf{S} = \begin{pmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_m \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix}$$

we can replace (8.10) above with (8.13) below,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) = \mathbf{0} \tag{8.12}$$

$$\mathbf{S}\Lambda\mathbf{e} - \mu\mathbf{e} = \mathbf{0} \tag{8.13}$$

$$g_i(\mathbf{x}) - s_i = 0 \quad i = 1, \ldots, m \tag{8.14}$$

It is worth noting that as $\mu \to 0$, the above equations (along with $\lambda, \mathbf{s} \geq 0$) represent the KKT conditions for the original problem (8.1)-(8.2). Notice that with $\mu = 0$, (8.13) (perhaps more easily seen in (8.10)) expresses complementary slackness for the slack variable bounds, i.e. either $\lambda_i = 0$ or $s_i = 0$.

8.1.2 Problem Solution

We will use the NR method to solve the set of equations represented by (8.12)-(8.14). The coefficient matrix will be the first derivatives of these equations. We note that we have n equations from (8.12), m equations from (8.13) and m equations from (8.14). Similarly, these equations are functions of n variables x, m variables λ, and m variables s.

The coefficient matrix for NR can be represented as,

$$\begin{bmatrix} \left(\nabla_x f_{1NR}\right)^T & \left(\nabla_s f_{1NR}\right)^T & \left(\nabla_\lambda f_{1NR}\right)^T \\ \vdots & \vdots & \vdots \\ \left(\nabla_x f_{(n+2m)NR}\right)^T & \left(\nabla_s f_{(n+2m)NR}\right)^T & \left(\nabla_\lambda f_{(n+2m)NR}\right)^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \\ \Delta\lambda \end{bmatrix} = -\begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} \tag{8.15}$$

where f_{1NR} (the first NR equation) is given by,

$$f_{1NR} = \frac{\partial f}{\partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_1} \tag{8.16}$$

and *r1*, *r2* and *r3* represent the residuals for (8.12)–(8.14). If we substitute (8.16) into the first row of (8.15), the first row of the coefficient matrix becomes,

$$\underbrace{\left[\frac{\partial^2 f}{\partial x_1^2} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_1^2}\right], \ldots, \left[\frac{\partial^2 f}{\partial x_n \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_n \partial x_1}\right]}_{\nabla_x f^T}, \underbrace{[0], \ldots, [0]}_{\nabla_s f^T}, \underbrace{\left[-\frac{\partial g_1}{\partial x_1}\right], \ldots, \left[-\frac{\partial g_m}{\partial x_1}\right]}_{\nabla_\lambda f^T}$$

Recalling,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i \tag{8.17}$$

and

$$\mathbf{J}^k = \begin{bmatrix} \left(\nabla g_1^k\right)^T \\ \vdots \\ \left(\nabla g_m^k\right)^T \end{bmatrix} \qquad \left(\mathbf{J}^k\right)^T = \begin{bmatrix} \nabla g_1^k, & \cdots & \nabla g_m^k \end{bmatrix}$$

we can write the NR iteration equations at step k as,

$$\begin{bmatrix} \nabla_x^2 L^k & 0 & \left(-\mathbf{J}^k\right)^T \\ 0 & \Lambda^k & \mathbf{S}^k \\ \mathbf{J}^k & -\mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{s}^k \\ \Delta \lambda^k \end{bmatrix} = - \begin{bmatrix} \nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) \\ \mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} \\ \mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k \end{bmatrix} \tag{8.18}$$

At this point, we could solve this system of equations to obtain $\Delta\mathbf{x}^k$, $\Delta\mathbf{s}^k$, and $\Delta\boldsymbol{\lambda}^k$. However, for large problems we can gain efficiency by simplifying this expression. If we look at row 2 above,

$$\Lambda^k \Delta \mathbf{s}^k + \mathbf{S}^k \Delta \lambda^k = -\mathbf{S}^k \Lambda^k \mathbf{e} + \mu \mathbf{e} \tag{8.19}$$

We would like to solve (8.19) for $\Delta\mathbf{s}^k$. Rearranging terms and pre-multiplying both sides by $\left(\Lambda^k\right)^{-1}$ gives,

$$\Delta \mathbf{s}^k = -\left(\Lambda^k\right)^{-1} \mathbf{S}^k \Lambda \mathbf{e} + \mu \Lambda^{-1} \mathbf{e} - \Lambda^{-1} \mathbf{S}^k \Delta \mathbf{x}^k \tag{8.20}$$

For diagonal matrices, the order of matrix multiplication does not matter, so the first term on the right hand side simplifies to give,

$$\Delta \mathbf{s}^k = -\mathbf{S}^k \mathbf{e} + \mu \Lambda^{-1} \mathbf{e} - \Lambda^{-1} \mathbf{S}^k \Delta \mathbf{x}^k \tag{8.21}$$

Examining the first and second terms on the right hand side, we note that,

$$-\mathbf{S}^k \mathbf{e} = -\begin{pmatrix} s_1^k & & 0 \\ & \ddots & \\ 0 & & s_n^k \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -\mathbf{s}^k \qquad \mu(\Lambda^k)^{-1} \mathbf{e} = \begin{pmatrix} \frac{\mu}{\lambda_1^k} & & 0 \\ & \ddots & \\ 0 & & \frac{\mu}{\lambda_m^k} \end{pmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{s}^k$$

Equation (8.21) becomes,

$$\begin{aligned}\Delta\mathbf{s}^k &= -\mathbf{s}^k + \mathbf{s}^k - (\Lambda^k)^{-1}\mathbf{S}^k\Delta\mathbf{x}^k \\ &= -(\Lambda^k)^{-1}\mathbf{S}^k\Delta\mathbf{x}^k\end{aligned} \tag{8.22}$$

We define Ω to be,

$$\Omega^k = (\Lambda^k)^{-1}\mathbf{S}^k \tag{8.23}$$

so that we have,

$$\Delta\mathbf{s}^k = -\Omega^k\Delta\lambda^k \tag{8.24}$$

We can then write (8.18) as a reduced set of equations:

$$\begin{bmatrix} -\nabla_x^2 L^k & \left(\mathbf{J}^k\right)^T \\ \mathbf{J}^k & \Omega^k \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\lambda^k \end{bmatrix} = -\begin{bmatrix} -\nabla f(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i\nabla g_i(\mathbf{x}) \\ \mathbf{g}_i(\mathbf{x}) - \mathbf{s}_i \end{bmatrix} \tag{8.25}$$

This matrix represents a linear symmetric system of equations (symmetric because the off-diagonal elements are the transpose of one another), of lower dimension than (8.18) and is more efficiently solved. Once we have solved for $\Delta\lambda^k$, we can use (8.24) to get $\Delta\mathbf{s}^k$.

8.1.3 The Line Search

It is sometimes said, "the devil is in the details." That is certainly true for the line search for the IP method. Modern algorithms employ a number of line search techniques, including merit functions, trust regions and filter methods [XX, XX, XX]. They include techniques to handle non-positive-definite hessians or otherwise poorly conditioned problems, or to regain feasibility. Sometimes a different step length is used for the primal variables (**x, s**) and the dual variables (λ). We will adopt that strategy here as well.

We can consider $\Delta\mathbf{x}, \Delta\mathbf{s}, \Delta\lambda,$ as the search directions for **x, s**, and $\boldsymbol{\lambda}$. We then need to determine the step size (between 0-1) in these directions,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k\Delta\mathbf{x}^k \tag{8.26}$$

$$\lambda^{k+1} = \lambda^k + \alpha^k\Delta\lambda^k \tag{8.27}$$

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha^k\Delta\mathbf{s}^k \tag{8.28}$$

Similar to SQP, a straightforward method is to accept α if it results in a decrease in a merit function that combines the objective with a sum of the violated constraints, i.e.,

$$P = f^k + \nu\sum_{i=1}^{viol}\left|g_i\right|$$

where ν is a constant. We will also reduce the step length if necessary to keep a slack variable or a Lagrange multiplier positive.

8.1.4 Example 1: Two Variables, One Constraint

We will apply the IP method to the same example given for SQP in Section 8.3.8, namely,

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5 \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 \ge 0 \end{aligned}$$

We reformulate the problem using a slack variable, s_1, for the constraint:

$$\begin{aligned} \text{Min} \quad & f(\mathbf{x}) = x_1^4 - 2x_2x_1^2 + x_2^2 + x_1^2 - 2x_1 + 5 \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 - s_1 = 0 \\ & s_1 \ge 0 \end{aligned}$$

We eliminate the lower bound for s_1 by adding a barrier term,

$$\begin{aligned} \text{Min} \quad & f_\mu(\mathbf{x}) = f(\mathbf{x}) - \mu\ln(s_1) \\ \text{s.t.} \quad & g(\mathbf{x}) = -(x_1 + 0.25)^2 + 0.75x_2 - s_1 = 0 \end{aligned}$$

At the starting point we have,

$$(\mathbf{x}^0)^{\mathrm{T}} = [-1,4],\ f^0 = 17,\ (\nabla f^0)^T = [8,6],\ \ g^0 = 2.4375,\ (\nabla g^0)^T = [1.5,\ 0.75]$$

We will assume we do not have second derivatives available, so we will begin with the Hessian set to $\nabla^2 \mathrm{L}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. $\mathbf{J}^0 = [1.5,\ 0.75]$. We will also set $\mu^0 = 5,\ s_1^0 = 2.4375,\ \lambda_1^0 = 2$.

This value of λ was picked to approximately satisfy (8.13), i.e. $\lambda^0 s^0 - \mu^0 = 0$.

If the merit function increases for a proposed step, we will cut the step in half and continue doing so several times.

8.1.4.1 First Iteration

Based on the data above, the coefficient matrix (refer back to (8.18)) for the first step is,

$$\begin{bmatrix} 1 & 0 & 0 & -1.5 \\ 0 & 1 & 0 & -0.75 \\ 0 & 0 & 2 & 2.4375 \\ 1.5 & 0.75 & -1 & 0 \end{bmatrix}$$

For the residual vector, we have,

$$\nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) = \begin{bmatrix} 8 \\ 6 \end{bmatrix} - 2\begin{bmatrix} 1.5 \\ 0.75 \end{bmatrix} = \begin{bmatrix} 5 \\ 4.5 \end{bmatrix}$$

$$\mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} = \begin{bmatrix} 2.4375 \end{bmatrix}\begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} 5 \end{bmatrix} = \begin{bmatrix} -0.125 \end{bmatrix}$$

$$\mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k = \begin{bmatrix} 2.4375 \end{bmatrix} - \begin{bmatrix} 2.4375 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

Thus the set of equations for our first step becomes,

$$\begin{bmatrix} 1 & 0 & 0 & -1.5 \\ 0 & 1 & 0 & -0.75 \\ 0 & 0 & 2 & 2.4375 \\ 1.5 & 0.75 & -1 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^0 \\ \Delta s^0 \\ \Delta \lambda^0 \end{bmatrix} = -\begin{bmatrix} 5 \\ 4.5 \\ -0.125 \\ 0 \end{bmatrix}$$

The solution is $\Delta \mathbf{x}^T = [-0.930,\ -2.465],\quad \Delta s = -3.244,\quad \Delta \lambda = 2.713$
Because the full step would make the slack negative, we set the step length for **x** and **s** to be 0.748 to keep the slack slightly positive. We accept the full step for λ. At this trial point the objective has decreased to 11.78 but the constraint is violated at –0.471. However the merit function has decreased from 17 to 14.00 and we accept the step. Our new point is
$\mathbf{x}^T = [-1.695, 2.157],\quad s = 0.012,\quad \lambda = 4.713.$

8.1.4.2 Second Iteration

At $\left(\mathbf{x}^1\right)^T = [-1.695,\ 2.157],\quad f^1 = 11.78,\quad \left(\nabla f^1\right)^T = [-10.256 \quad -1.435]$

$g^1 = -0.4714,\quad \left(\nabla g^1\right)^T = [2.891 \quad 0.75]$

We start this step by updating the Hessian of the Lagrangian. To do so, we evaluate the Lagrangian gradient at $\mathbf{x}^0$ and $\mathbf{x}^1$. We then calculate the γ and Δ**x** vectors. (As we did for SQP, we use the same Lagrange multiplier, λ^1, for both gradients.)

$$\nabla \mathrm{L}\left(\mathbf{x}^0,\lambda^1\right)=\begin{bmatrix} 8.0 \\ 6.0 \end{bmatrix}-\left(4.713\right)\begin{bmatrix} 1.5 \\ 0.75 \end{bmatrix}=\begin{bmatrix} 0.930 \\ 2.465 \end{bmatrix}$$

$$\nabla \mathrm{L}\left(\mathbf{x}^1,\lambda^1\right)=\begin{bmatrix} -10.256 \\ -1.435 \end{bmatrix}-\left(4.713\right)\begin{bmatrix} 2.891 \\ 0.75 \end{bmatrix}=\begin{bmatrix} -23.881 \\ -4.967 \end{bmatrix}$$

$$\gamma^0=\nabla \mathrm{L}\left(\mathbf{x}^1,\lambda^1\right)-\nabla \mathrm{L}\left(\mathbf{x}^0,\lambda^1\right)=\begin{bmatrix} -24.811 \\ -7.435 \end{bmatrix}$$

$$\Delta\mathbf{x}^0=\begin{bmatrix} -1.695 \\ 2.157 \end{bmatrix}-\begin{bmatrix} -1.0 \\ 4.0 \end{bmatrix}=\begin{bmatrix} -0.695 \\ -1.843 \end{bmatrix}$$

We use this data to obtain the new estimate of the Lagrangian Hessian using BFGS Hessian update, as we did for SQP (see the SQP example for details). The new Hessian is,

$$\nabla^2 \mathrm{L}^1=\begin{bmatrix} 20.762 & 5.629 \\ 5.629 & 1.910 \end{bmatrix}$$

After evaluating the residuals, our next set of equations becomes,

$$\begin{bmatrix} 20.762 & 5.629 & 0 & -2.891 \\ 5.629 & 1.910 & 0 & -0.75 \\ 0 & 0 & 4.713 & 0.012 \\ 2.891 & 0.75 & -1 & 0 \end{bmatrix}\begin{bmatrix} \Delta x^1 \\ \Delta s^1 \\ \Delta\lambda^1 \end{bmatrix}=-\begin{bmatrix} -23.881 \\ -4.970 \\ -0.943 \\ -0.484 \end{bmatrix}$$

The solution gives: $\Delta\mathbf{x}^T=[1.104,\ -3.319]$, $\Delta s=0.218$, $\Delta\lambda=-6.797$. For λ we only take 0.693 of the step to keep $\lambda \geq 0$. With the full step for **x** and **s**, our merit function decreases from 14.0 to 10.8. Our new point is,

$$\left(\mathbf{x}^2\right)^T=[-0.592,\ -1.162],\quad s=0.230,\quad \lambda=0,\quad f=8.820,\quad g=-0.988$$

Subsequent steps proceed in a similar manner. The progress of the algorithm to the optimum, with our relatively unsophisticated line search, is shown in Fig. 8.10 below. The algorithm reaches the optimum in about nine steps. At every iteration we reduce μ according to the equation $\mu^{k+1}=\frac{\mu^k}{5}$. For comparison purposes, the progress of the Interior Point method in `fmincon` on this problem is shown in Fig. 8.11.

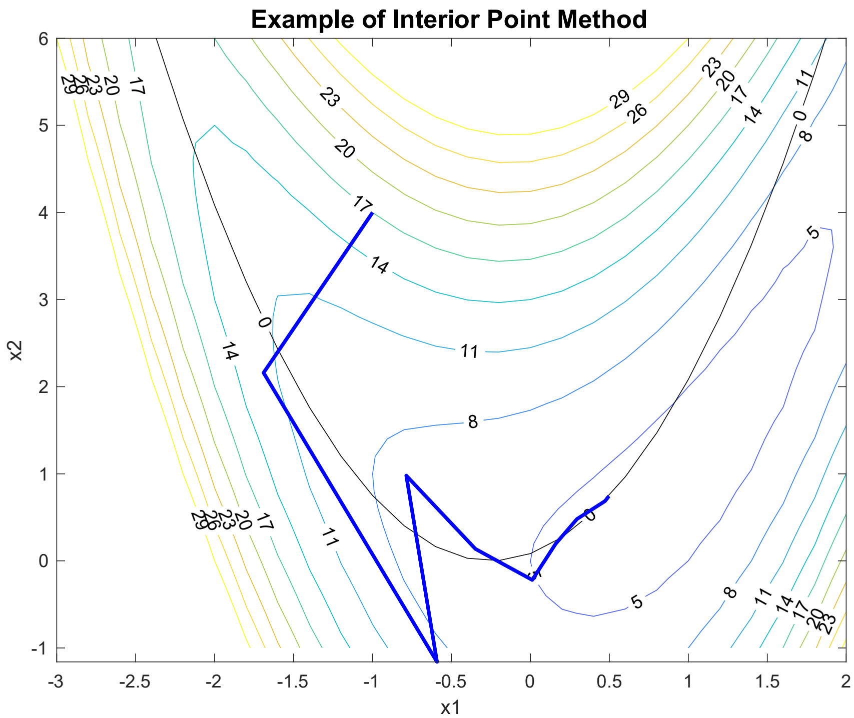

Fig 8.10. The progress of the IP algorithm on Example 1.

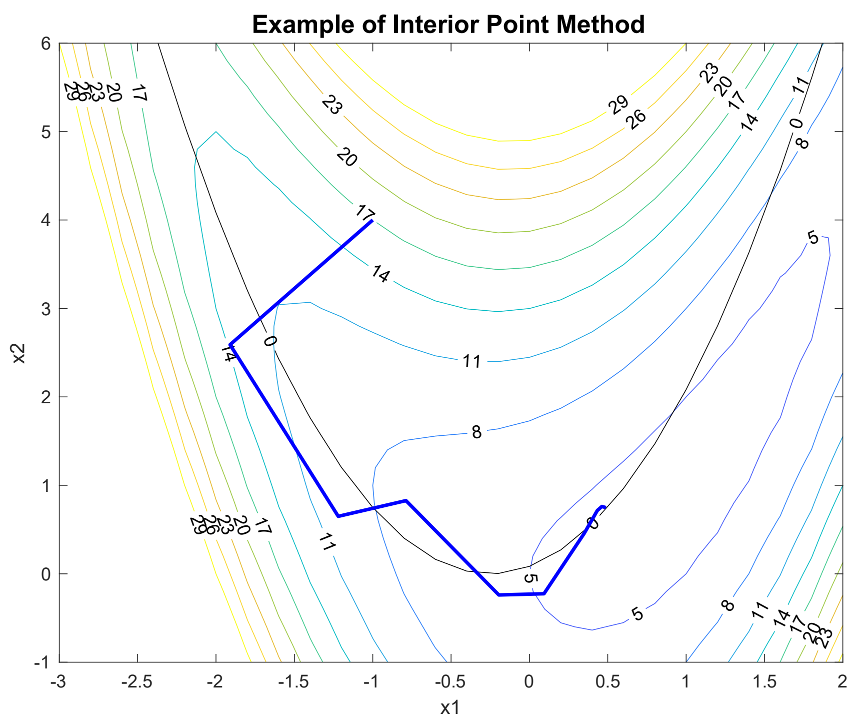

Fig. 8.11 Path of `fmincon` IP algorithm on same problem.

8.1.5 Example 2: One Variable, Two Constraints

In this example problem the functions are very simple, but the setup for the constraints is more involved than the previous example. We will show how the problem changes when we have two slack variables.

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 \ge 0 \\ & x_1 \ge 1 \end{aligned}$$

We will change the inequality and the bound to equality constraints by means of slack variables so that we have,

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 - s_1 = 0 \\ & x_1 - 1 - s_2 = 0 \\ & s_1, s_2 \ge 0 \end{aligned}$$

We remove the bounds on the slacks by adding barrier terms,

$$\begin{aligned} \text{Min} \quad & f_\mu = x_1^2 - \mu \sum_{i=1}^{2} \ln(s_i) \\ \text{s.t.} \quad & -2x_1 + 9 - s_1 = 0 \\ & x_1 - 1 - s_2 = 0 \end{aligned}$$

By inspection, the solution to the problem is $x_1 = 1, \quad s_1 = 7, \quad s_2 = 0; \quad f = 1$.

8.1.5.1 First Iteration

At our starting point $x_1 = 3, \quad \left(\mathbf{s}^0\right)^T = [3,2], \quad f^0 = 9, \quad g_1^0 = 0, \quad g_2^0 = 0$

We also have, $\nabla f^0 = [6], \quad \nabla^2 f = [2], \quad \nabla g_1 = [-2], \quad \nabla g_2 = [1], \quad \nabla^2 g_1 = [0], \quad \nabla^2 g_2 = [0]$

We will use $\mu^0 = 2, \quad \alpha^0 = 0.5, \quad (\lambda^0)^T = [1,1]$. Thus,

$$\mathbf{S}^0 = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \quad \Lambda^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{J}^0 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Because we only have one variable, the Hessian of the Lagrangian is just a 1x1 matrix (we use the actual second derivative here for simplicity):

$$\nabla_x^2 L = \nabla^2 f - \lambda_1 \nabla^2 g_1 - \lambda_2 \nabla^2 g_2 = [2] - (1)[0] - (1)[0] = [2]$$

With this information we can then build our coefficient matrix,

$$\begin{bmatrix} \nabla_x^2 L^k & 0 & \left(-\mathbf{J}^k\right)^T \\ 0 & \Lambda^k & \mathbf{S}^k \\ \mathbf{J}^k & -\mathbf{I} & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 2 \\ -2 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

The vector of residuals is,

$$\nabla f^k(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i^k \nabla g_i^k(\mathbf{x}) = \quad [6]-(1)[-2]-(1)[1] = 7$$

$$\mathbf{S}^k \Lambda^k \mathbf{e} - \mu^k \mathbf{e} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 2.0 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{g}_i^k(\mathbf{x}) - \mathbf{s}_i^k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We can then solve for our new point as (recall alpha = 0.5),
$x_1^1 = 2.174$, $\left(\mathbf{s}^1\right)^T = [4.652, 1.174]$, $\left(\lambda^1\right)^T = [0.283, 1.413]$ at which point $f = 4.73$, and $g_1^1 = 0$, $g_2^1 = 0$.

8.1.6 *An Alternate Development of the Newton Iteration Equations

*This section is optional.

As mentioned at the start of the section on IP algorithms, there are two approaches to developing the NR equations: separating out the slack variables in their own vector, and including the slacks as part of the **x** vector. Previously we kept the slacks separate. Now we will combine them with **x.** This development follows the work by Wachter and Biegler [XX, XX].

For simplicity, we will start with a problem which only has equality constraints. We will, however, include a lower bound on the variables:

$$\text{Min} \quad f\left(\mathbf{x}\right) \tag{8.29}$$

$$\text{s.t.} \quad g_i\left(\mathbf{x}\right) = 0 \quad i = 1, \ldots, m \tag{8.30}$$

$$\mathbf{x} \geq 0 \tag{8.31}$$

As before, we eliminate the lower bounds by including them in the objective function with a barrier function:

$$\text{Min} \quad f_\mu = f(\mathbf{x}) - \mu \sum_{i=1}^{n} \ln(x_i) \tag{8.32}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) = 0 \quad i = 1, \ldots, m \tag{8.33}$$

We now consider the necessary conditions for a solution to the barrier problem represented by (8.32)-(8.33). We first define,

$$z_i = \frac{\mu}{x_i} \tag{8.34}$$

We can write the KKT conditions as,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) - \mathbf{z} = \mathbf{0} \tag{8.35}$$

$$g_i(\mathbf{x}) = 0 \qquad i = 1, \ldots, m \tag{8.36}$$

$$x_i z_i - \mu = 0 \qquad i = 1, \ldots, n \tag{8.37}$$

If we define **e** as the vector of 1's of *n* dimension and,

$$\mathbf{X} = \begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{pmatrix} \qquad \mathbf{Z} = \begin{pmatrix} z_1 & & 0 \\ & \ddots & \\ 0 & & z_n \end{pmatrix}$$

we can replace (8.37) above with (8.40) below,

$$\nabla f(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}) - \mathbf{z} = \mathbf{0} \tag{8.38}$$

$$g_i(\mathbf{x}) = 0 \quad i = 1, \ldots, m \tag{8.39}$$

$$\mathbf{XZe} - \mu \mathbf{e} = 0 \tag{8.40}$$

As $\mu \to 0$, the above equations (along with $\mathbf{z} \geq 0$) represent the KKT conditions for the original problem. The variables **z** can be viewed as the Lagrange multipliers for the bound constraints. Notice that with $\mu = 0$, (8.40) expresses complementary slackness for the bound constraints, i.e. either $x_i = 0$ or $z_i = 0$.

8.1.7 Problem Solution

We will now construct the coefficient matrix for the Newton iteration. We note that we have n equations from (8.35), m equations from (8.39) and n equations from (8.37). Similarly, these equations are functions of n variables x, m variables λ, and n variables z.

For example, the first row of the coefficient matrix for NR could be represented as,

$$\begin{bmatrix} \left(\nabla_x f_{1NR}\right)^T & \left(\nabla_\lambda f_{1NR}\right)^T & \left(\nabla_z f_{1NR}\right)^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\lambda \\ \Delta\mathbf{z} \end{bmatrix} = -\begin{bmatrix} r1 \end{bmatrix} \tag{8.41}$$

where f_{1NR} is given by,

$$f_{1NR} = \frac{\partial f}{\partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_1} - z_1 \tag{8.42}$$

and $r1$ represents the residuals for (8.38). If we substitute (8.42) into matrix (8.41), the first row of the coefficient matrix is,

$$\underbrace{\left[\frac{\partial^2 f}{\partial x_1^2} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_1^2} - \frac{\partial z_1}{\partial x_1}\right], \ldots, \left[\frac{\partial^2 f}{\partial x_n \partial x_1} - \sum_{i=1}^{m} \lambda_i \frac{\partial^2 g_i}{\partial x_n \partial x_1} - \frac{\partial z_1}{\partial x_n}\right]}_{\nabla_x f^T}, \underbrace{\left[-\frac{\partial g_1}{\partial x_1}\right], \ldots, \left[-\frac{\partial g_m}{\partial x_1}\right]}_{\nabla_\lambda f^T}, \underbrace{[-1], \ldots, [0]}_{\nabla_z f^T}$$

Defining,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i + \mu \mathbf{X}^{-2} \tag{8.43}$$

we can write the NR iteration equations at step k as,

$$\begin{bmatrix} \nabla_x^2 \mathrm{L}^k & \left(-\mathbf{J}^k\right)^T & -\mathbf{I} \\ \mathbf{J}^k & 0 & 0 \\ \mathbf{Z}^k & 0 & \mathbf{X}^k \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\lambda^k \\ \Delta\mathbf{z}^k \end{bmatrix} = -\begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} \tag{8.44}$$

The reader might wish to compare this with (8.18). As before, we could solve this system of equations to obtain $\Delta\mathbf{x}^k$, $\Delta\boldsymbol{\lambda}^k$, and $\Delta\mathbf{z}^k$. However, we can simplify this expression. We will start by looking at the third set of equations in (8.18) above,

$$\mathbf{Z}^k \Delta\mathbf{x}^k + 0 + \mathbf{X}^k \Delta\mathbf{z}^k = -\mathbf{X}^k \mathbf{Z}^k \mathbf{e} + \mu\mathbf{e} \tag{8.45}$$

where we have substituted in the actual residual value for $r3$, i.e.,

$$-r3 = -\mathbf{X}^k\mathbf{Z}^k\mathbf{e} + \mu\mathbf{e}$$

We would like to solve (8.45) for $\Delta\mathbf{z}^k$. Rearranging terms gives,

$$\mathbf{X}^k\Delta\mathbf{z}^k = -\mathbf{X}^k\mathbf{Z}^k\mathbf{e} + \mu\mathbf{e} - \mathbf{Z}^k\Delta\mathbf{x}^k \tag{8.46}$$

If we pre-multiply both sides by $(\mathbf{X}^k)^{-1}$ we have,

$$\Delta\mathbf{z}^k = -\mathbf{Z}^k\mathbf{e} + \mu(\mathbf{X}^k)^{-1}\mathbf{e} - (\mathbf{X}^k)^{-1}\mathbf{Z}^k\Delta\mathbf{x}^k \tag{8.47}$$

Examining the first and second terms on the right hand side, we note that,

$$-\mathbf{Z}^k\mathbf{e} = -\begin{pmatrix} z_1^k & & 0 \\ & \ddots & \\ 0 & & z_n^k \end{pmatrix}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -\mathbf{z}^k \qquad \mu(\mathbf{X}^k)^{-1}\mathbf{e} = \begin{pmatrix} \frac{\mu}{x_1^k} & & 0 \\ & \ddots & \\ 0 & & \frac{\mu}{x_n^k} \end{pmatrix}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{z}^k$$

Equation (8.47) becomes,

$$\begin{aligned}\Delta\mathbf{z}^k &= -\mathbf{z}^k + \mathbf{z}^k - (\mathbf{X}^k)^{-1}\mathbf{Z}^k\Delta\mathbf{x}^k \\ &= -(\mathbf{X}^k)^{-1}\mathbf{Z}^k\Delta\mathbf{x}^k\end{aligned} \tag{8.48}$$

We define **S** to be,

$$\mathbf{S}^k = (\mathbf{X}^k)^{-1}\mathbf{Z}^k \tag{8.49}$$

so that we have,

$$\Delta\mathbf{z}^k = -\mathbf{S}^k\Delta\mathbf{x}^k \tag{8.50}$$

Now we will examine the first two rows of (8.44). These represent equations,

$$\nabla_x^2 L^k\Delta\mathbf{x}^k + \left(-\mathbf{J}^k\right)^T\Delta\lambda^k + -\mathbf{I}\Delta\mathbf{z}^k = -r1 \tag{8.51}$$

$$\mathbf{J}^k\Delta\mathbf{x}^k = -r2 \tag{8.52}$$

We see that $\Delta\mathbf{z}^k$ only appears in (8.51). If we substitute (8.50) for $\Delta\mathbf{z}^k$ and gather terms, (8.51) becomes,

$$\left[\nabla_x^2 L^k + \mathbf{S}^k\right]\Delta\mathbf{x}^k + \left(-\mathbf{J}^k\right)^T \Delta\lambda^k = -r1 \quad (8.53)$$

or in matrix form,

$$\begin{bmatrix} \nabla_x^2 L^k + \mathbf{S} & \left(-\mathbf{J}^k\right)^T \\ \mathbf{J}^k & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^k \\ \Delta\lambda^k \end{bmatrix} = -\begin{bmatrix} r1 \\ r2 \end{bmatrix} \quad (8.54)$$

Once we have solved for $\Delta\mathbf{x}^k$, we can use (8.50) to get $\Delta\mathbf{z}^k$.

8.1.8 Example 1: Solving the IP Equations

We will illustrate this approach on a similar example problem used earlier:

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 \ge 0 \\ & x_1 \ge 0 \end{aligned}$$

We will change the inequality constraint to an equality constraint by means of a slack variable, x_2, so that we have,

$$\begin{aligned} \text{Min} \quad & f = x_1^2 \\ \text{s.t.} \quad & -2x_1 + 9 - x_2 = 0 \\ & x_1, x_2 \ge 0 \end{aligned}$$

We will eliminate the lower bounds by using a barrier formulation:

$$\begin{aligned} \text{Min} \quad & f_\mu = x_1^2 - \mu \sum_{i=1}^{2} \ln(x_i) \\ \text{s.t.} \quad & -2x_1 - x_2 + 9 = 0 \end{aligned}$$

starting from $(\mathbf{x}^0)^T = [3,3]$. At this point, $f^0 = 9, (\nabla f^0)^T = [6,0];\ g^0 = 0, (\nabla g^0)^T = [-2,-1]$. We will set $\mu = 10$ and $\lambda = 1$. This then gives, $z_1 = \frac{\mu}{x_1} = 3.333$ and $z_2 = \frac{\mu}{x_2} = 3.333$ Noting,

$$\nabla_x^2 L = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i + \mu \mathbf{X}^{-2} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} - (1)\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1.111 & 0 \\ 0 & 1.111 \end{bmatrix}$$

$$\mathbf{J}^k = [-2, -1]$$

We can write the coefficient matrix,

$$\begin{bmatrix} \nabla_x^2 L^k & \left(-\mathbf{J}^k\right)^T & -\mathbf{I} \\ \mathbf{J}^k & 0 & 0 \\ \mathbf{Z}^k & 0 & \mathbf{X}^k \end{bmatrix} = \begin{bmatrix} 3.111 & 0 & 2 & -1 & 0 \\ 0 & 1.111 & 1 & 0 & -1 \\ -2 & -1 & 0 & 0 & 0 \\ 3.333 & 0 & 0 & 3 & 0 \\ 0 & 3.333 & 0 & 0 & 3 \end{bmatrix}$$

By evaluating (8.38) through (8.40) we find the vector of residuals to be,

$$-\begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} = -\begin{bmatrix} 4.667 \\ -2.333 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Solving this set of equations, gives $\Delta\mathbf{x} = \begin{bmatrix} -0.712 \\ 1.424 \end{bmatrix}, \quad \Delta\lambda = -0.831, \quad \Delta\mathbf{z} = \begin{bmatrix} 0.791 \\ -1.582 \end{bmatrix}$

If we take the full step by adding these delta values to our beginning values, we have

$$\mathbf{x}^1 = \begin{bmatrix} 2.288 \\ 4.424 \end{bmatrix}, \quad \lambda^1 = 0.169, \quad \mathbf{z}^1 = \begin{bmatrix} 4.124 \\ 1.805 \end{bmatrix}$$

At this new point the constraint is satisfied ($g^1 = 0$) and the objective has decreased from 9 to 5.235.

8.2 References

Nocedal J. and S. Wright, *Numerical Optimization*, 2nd Edition, Springer, 2006

Wachter A. and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming,* 2005.

Powell, M. J. D., "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," *Mathematical Programming,* 14, (1978) 224-248.

Powell, M.J.D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Numerical Analysis, Dundee 1977,* Lecture Notes in Mathematics no. 630, Springer-Verlag, New York, 1978.

L.S. Lasdon et. al, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 4 No. 1 March 1978

Gabriele G.A., and K.M. Ragsdell, "The Generalized Reduced Gradient Method: A Reliable Tool for Optimal Design," *Trans ASME, J. Eng. Industry,* May 1977.

Byrd, R. H., J. C. Gilbert, and J. Nocedal. "A trust region method based on interior point techniques for nonlinear programming." *Mathematical Programming*, Vol 89, No. 1, 2000, pp. 149–185.

Waltz, R. A., J. L. Morales, J. Nocedal, and D. Orban. "An interior algorithm for nonlinear optimization that combines line search and trust region steps." *Mathematical Programming*, Vol 107, No. 3, 2006, pp. 391–408.

Biegler, L. T., *Nonlinear Programming, Concepts, Algorithms, and Applications to Chemical Processes*, SIAM, 2010

Goldfarb, D., and A. Idnani, "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs," *Math. Programming,* v. 27, 1983, p.1-33.

The IP algorithm uses "barrier functions" to enforce feasibility of the constraints. It can be used on very large problems. Continue….

The GRG algorithm works by computing search directions which improve the objective and satisfy the constraints, and then conducting line searches in a very similar fashion to the algorithms we studied in Chapter 3. GRG requires more function evaluations than SQP, but it has the desirable property that it stays feasible once a feasible point is found. If the optimization process is halted before the optimum is reached, the designer is guaranteed to have in hand a better design than the starting design.

Other more sophisticated approaches apply trust regions and filter methods. [XX, XX]. For example we might accept α if it leads to "sufficient" progress towards reducing the barrier function or the constraint violation. Part of this approach is to maintain a filter which contains combinations of constraint violation values and barrier function values that are prohibited for a successful step length. Initially, the filter is set so the algorithm will not allow trial points to be accepted that exceed a maximum constraint violation. Later the filter is augmented with additional conditions which help insure the algorithms does not cycle between points that alternate between decreasing the constraint violation and the barrier function.

If we use the more efficient method to solve these equations represented by, we first compute **S**:

$$\mathbf{S}^k = (\mathbf{X}^k)^{-1}\mathbf{Z}^k = \begin{bmatrix} 0.333 & 0 \\ 0 & 0.333 \end{bmatrix}\begin{bmatrix} 3.333 & 0 \\ 0 & 3.333 \end{bmatrix} = \begin{bmatrix} 1.111 & 0 \\ 0 & 1.111 \end{bmatrix}$$

We can then define the set of equations given by XX,

$$\begin{bmatrix} 4.222 & 0 & 2 \\ 0 & 2.222 & 1 \\ -2 & -1 & 0 \end{bmatrix}\begin{bmatrix} \Delta\mathbf{x} \\ \Delta\lambda \end{bmatrix} = -\begin{bmatrix} 4.667 \\ -2.333 \\ 0 \end{bmatrix}$$

The solution gives us the same $\Delta\mathbf{x}$ and $\Delta\lambda$,

$$\Delta\mathbf{x} = \begin{bmatrix} -0.712 \\ 1.424 \end{bmatrix}, \quad \Delta\lambda = -0.831$$

Substitution of the $\Delta\mathbf{x}$ values into (8.24) gives us $\Delta\mathbf{z}$:

$$\Delta\mathbf{z}^k = -\mathbf{S}^k\Delta\mathbf{x}^k = -\begin{bmatrix} 1.111 & 0 \\ 0 & 1.111 \end{bmatrix}\begin{bmatrix} -0.712 \\ 1.424 \end{bmatrix} = \begin{bmatrix} 0.791 \\ -1.582 \end{bmatrix}$$

which is what we computed previously.