We are trying to optimise the objective function

$$X^T \log(\Lambda) - 1^T \Lambda - \frac{1}{2}(f(\Lambda) - \mu - P)^T \Sigma^{-1} (f(\Lambda) - \mu - P)$$

where μ and Σ are assumed known, P is the projection of $f(\Lambda)$ onto the first k principal components of Σ. To simplify the algebra, we will let $f(\Lambda) = \mu + Va$ for some vector a. This makes the objective function

$$X^T \log(\Lambda) - 1^T \Lambda - \sum_{i=k+1}^{p} \frac{{a_i}^2}{2d_i}$$

The derivative of this objective function is

$$\sum_j \frac{X_j}{\Lambda_j} \frac{d\Lambda_j}{da_i} - \sum_j \frac{d\Lambda_j}{da_i} - I_{>k}(i) \frac{a_i}{d_i}$$

Suppose we let $g = f^{-1}$ and g' be the derivative of g. Then we have

$$\Lambda_j = g\left((\mu + Va)_j\right) = g\left(\mu_j + \sum_i V_{ji} a_i\right)$$

and therefore

$$\frac{d\Lambda_j}{da_i} = g'\left(f\left(\Lambda_j\right)\right) V_{ji}$$

Substituting this into the derivative of the objective function gives:

$$\sum_j X_j V_{ji} \frac{g'\left(f\left(\Lambda_j\right)\right)}{\Lambda_j} - \sum_j V_{ji} g'\left(f\left(\Lambda_j\right)\right) - I_{>k}(i) \frac{a_i}{d_i}$$

or equivalently

$$\sum_j \frac{(X_j - \Lambda_j) V_{ji}}{\Lambda_j f'\left(\Lambda_j\right)} - \frac{a_i}{d_i}$$

In the particular case where $f(\lambda) = \log(\lambda)$, this simplifies — the derivative is

$$V^T X - V^T \Lambda - D^{-1} a_{k^+}$$

The solution is therefore the solution to

$$V^T (X - \Lambda) = D^{-1} a_{k^+}$$

This suggests an iterative approach to solving this problem

1. $\Lambda = X + VD^{-1}a$
2. $a = V^T(\log(\Lambda) - \mu)$

This approach can get stuck in cycles. To prevent this, we can take partial steps. Instead of following this full step, we calculate the new value of a using this method and take a step that goes only 50% of the way towards it. That is

$$a_{\text{new}} = 0.5\hat{a} + 0.5a_{\text{old}}$$

Empirically, this seems fairly robust, but it may be possible to adjust the weights to get faster convergence.

This approach can be unstable if D has small values. An alternative approach is the reversed approach. Given Λ, it is straightforward to find the X for which this value of Λ maximises the objective function. That is, we get

$$X = \Lambda + VD^*V^T(\log(\Lambda) - \mu)$$

We can therefore solve this by interpolation. Given a value of Λ, we can let $s = VD^*V^T\log(\Lambda - \mu)$ and $t = X - \Lambda$. Then we can let $\Lambda^* = \exp((VD^*V^T)^{-1}t + \mu)$, and $X^+ = \Lambda + s$, $X^- = \Lambda^* + t$. Now we find the weighted average of X^+ and X^- that is closest to X, and take the corresponding weighted average of Λ and Λ^* as our new value of Λ.

We need to start with $\Lambda > 0$.

and the second derivative with respect to a_l is

$$\sum_j V_{ji}\frac{d\Lambda_j}{da_l} - 2{d_l}^{-1}\delta_{il}1_{l>k}$$

Recalling that for $f(\lambda) = \log(\lambda)$, we have $\frac{d\Lambda_j}{da_l} = V_{jl}\Lambda_j$, so the second derivative with respect to a_i and a_l is

$$-\sum_j V_{ji}V_{jl}\Lambda_j - 2{d_l}^{-1}\delta_{il}1_{l>k}$$

If we let L be a diagonal matrix with values Λ, then we can write this in matrix form as

$$-\left(V^TLV + 2D_{k^+}^{-1}\right)$$

Now, letting $S = V^TD_{k^+}^{-1}V$ the Newton-Raphson method gives

$$\begin{aligned}
a_{new} &= a - \left(VLV^T - 2D^{-1}\right)^{-1}\left(VX + V\Lambda - 2D^{-1}a\right) \\
&= a - V\left(L - 2S\right)^{-1}V^T\left(VX + V\Lambda - 2D^{-1}a\right) \\
&= a - V\left(L - 2S\right)^{-1}\left(X + \Lambda - 2V^TD^{-1}a\right) \\
&= a - V\left(I - 2L^{-1}S\right)^{-1}L^{-1}\left(X + \Lambda - 2V^TD^{-1}a\right) \\
&= a - V\left(I - 2L^{-1}S\right)^{-1}\left(L^{-1}X + 1 - 2L^{-1}V^TD^{-1}a\right)
\end{aligned}$$