Numerical Analysis: Linear Algebra Final

Truman Ellis

December 9, 2010

Problem 1

a) The following results provide the underpinnings of a proof that the algorithm completes and computes the Cholesky factorization.

i Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & \star \\ \hline A_{BL} & A_{BR} \end{array}\right)$ and $L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)$, where the $\star$ indicates the symmetric part of A that is not stored, A_{00} and L_{00} are square and L is lower triangular.

ii Assume that A_{00} has a Cholesky factorization $A_{00} = L_{00}L_{00}^T$.

iii Argue that $l_{10}^T := a_{10}^T L_{00}^{-T}$ is well-defined.

Proof

With the assumption that A_{00} has a Cholesky factorization, we have $L_0 0$, and since L_{00} is the Cholesky factorization of a symmetric positive definite matrix, it is invertible. Multiplying both sides by L_{00}^{-T} on the right, we can rewite this as

$$l_{10}^T L_{00}^T = a_{10}^T$$

or

$$L_{00} l_{10} = a_{10}\,.$$

Since L_{00} is lower triangular, we can trivially solve this for l_{10} at each iteration. Therefore this operation is well-defined.

iv Prove that $\alpha_{11} - l10^T l_{10}$ is positive.

Proof

First let's expand our expression: $l_{10}^T = a_{10}^T L_{00}^{-T}$. So

$$\alpha_{11} - l_{10}^T l_{10} = \alpha_{11} - a_{10}^T L_{00}^{-T} L_{00}^{-1} a_{10} = \alpha_{11} - a_{10}^T A_{00}^{-1} a_{10}\,.$$

To the contrary, assume $\alpha_{11} - a_{10}^T A_{00}^{-1} a_{10} < 0$, then there exists a positive constant C such that, $\alpha_{11} = a_{10}^T A_{00}^{-1} a_{10} - C$. Then we can substitute this equality into A and compute $x^T A x$ with

$$x = \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline 0 \end{array}\right)\,.$$

Then we get

$$
\begin{aligned}
x^T Ax &= \left(\begin{array}{c|c|c} x_0^T & \chi_1 & 0 \end{array}\right) \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & a_{10}^T A_{00}^{-1} a_{10} - C & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right) \left(\begin{array}{c} x_0 \\ \hline \chi_1 \\ \hline 0 \end{array}\right) \\
&= x_0^T A_{00} x_0 + \chi_1 a_{10}^T x_0 + x_0^T a_{10} \chi_1 + \chi_1 (a_{10}^T A_{00}^{-1} a_{10} - C) \chi_1 \\
&= x_0^T A_{00} x_0 + \chi_1 a_{10}^T (x_0 + A_{00}^{-1} a_{10} \chi_1) + x_0^T a_{10} \chi_1 - \chi_1 C \chi_1 \,.
\end{aligned}
$$

Now let $x_0 = -A_{00}^{-1} a_{10} \chi_1$, then

$$
\begin{aligned}
x^T Ax &= \cancel{\chi_1^2 a_{10}^T A_{00}^{-T} a_{10}} - \cancel{\chi_1^2 a_{10}^T A_{00}^{-T} a_{10}} - C\chi_1^2 \\
&= -C\chi_1^2 \,.
\end{aligned}
$$

Since $C > 0$ by definition, the result is negative for $\chi_1 \neq 1$. This is a contradiction since A is symmetric positive definite, thus we must have $\alpha_1 1 - l_{10}^T l_{10}$ positive.

b) Use the above insights to provide and inductive proof of the Cholesky factorization theorem.

Proof

Proof by induction.

Base case: $n = 1$. Clearly the result is true for a 1×1 matrix $A = \alpha_{11}$: Since A is symmetric positive definite, α_{11} is real and positive and the Cholesky factorization is simply given by $\lambda_{11} = \sqrt{\alpha_{11}}$, and is unique if we insist that λ_{11} be positive.

Inductive step: Assume the result is true for symmetric positive definite matrix $A \in \mathbb{C}^{n \times n}$. We need to show that it holds for $A \in \mathbb{C}^{(n+1)\times(n+1)}$. Let $A \in \mathbb{C}^{(n+1)\times(n+1)}$ be symmetric positive definite. Partition A and L,

$$
\left(\begin{array}{c|c} A_{TL} & \star \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & \star & \star \\ \hline a_{10}^T & \alpha_{11} & \star \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)
$$

$$
\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right) .
$$

Let $l_{10}^T = a_{10}^T L_{00}^{-T}$, which we just showed is well-defined and $A_{00} = L_{00} L_{00}^T$ by the inductive hypothesis. Then let $\lambda_{11} = \sqrt{\alpha_{11} - l_{10}^T l_{10}}$, where $\alpha_{11} - l_{10}^T l_{10}$ is positive. Finally, $L_{22} = \textsc{Chol}(A_{22})$, whic exists by the inductive hypothesis. Then L is the desired Cholesky factor of A.
By the principle of mathematical induction, the theorem holds.

Problem 2

Given a matrix $A \in \mathbb{R}^{m \times n}$. A *communication avoiding* algorithm for the QR factorization follows:

- View matrix A as two submatrices $B\,, C \in \mathbb{R}^{\frac{m}{2} \times n}$ so that $A = \left(\begin{array}{c} B \\ \hline C \end{array}\right)$.

- Compute the QR factorizations of B and C: $B = Q_B R_B$ and $C = Q_C R_C$.
- Compute the QR factorization of $\left(\begin{array}{c} R_B \\ \hline R_C \end{array}\right) = Q_{BC} R_{BC}$.

a) Prove that, under mild conditions, the R_{BC} that results from the above procedure is exactly the R that would come out of a QR factorization of A. What are those mild conditions?

Proof

Expanding according to the process described above,

$$\begin{aligned} A = \left(\begin{array}{c} B \\ \hline C \end{array}\right) &= \left(\begin{array}{c} Q_B R_B \\ \hline Q_C R_C \end{array}\right) \\ &= \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right) \left(\begin{array}{c} R_B \\ \hline R_C \end{array}\right) \\ &= \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right) Q_{BC} R_{BC} \,, \end{aligned}$$

where Q_B and Q_C are $\frac{m}{2} \times n$, R_B, R_C, and R_{BC} are $n \times n$, and Q_{BC} is $2n \times n$.

The matrix $\left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right)$ is unitary, since Q_B and Q_C are unitary and

$$\begin{aligned} &\left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right)^H \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right) = \left(\begin{array}{c|c} Q_B^H Q_B & 0 \\ \hline 0 & Q_C^H Q_C \end{array}\right) \\ &= I = \left(\begin{array}{c|c} Q_B Q_B^H & 0 \\ \hline 0 & Q_C Q_C^H \end{array}\right) = \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right) \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right)^H . \end{aligned}$$

Since a unitary matrix represents a length-preserving transformation, the product of two unitary matrices is also unitary, and the final R_{BC} preserves the length information of the original matrix. Provided A has linearly independent columns, it has a unique QR factorization and

$$Q = \left(\begin{array}{c|c} Q_B & 0 \\ \hline 0 & Q_C \end{array}\right) Q_{BC}$$

and

$$R = R_{BC} \,,$$

where $A = QR$.

b) Propose an algorithm for computing the QR factorization of $\left(\begin{array}{c} R_B \\ \hline R_C \end{array}\right)$ via Householder transformations that takes advantage of the fact that R_B and R_C are upper triangular.

c) Compute the approximate operation count of your algorithm and compare it to the operation count had you used a standard Householder transformation based algorithm on $\left(\begin{array}{c} R_B \\ \hline R_C \end{array}\right)$ without taking any advantage of the special structure of R_B and R_C.

Algorithm: $[R] := \mathrm{QR}(R, U)$

Partition $\left(\begin{array}{c} R \\ \hline\hline U \end{array}\right) \rightarrow \left(\begin{array}{c} \left(\begin{array}{c|c} R_{TL} & R_{TR} \\ \hline 0 & R_{BR} \end{array}\right) \\ \hline\hline \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) \end{array}\right)$

where R_{TL} and U_{TL} are 0×0,

while $m(R_{TL}) < m(R)$ **do**

Repartition

$$\left(\begin{array}{c} \left(\begin{array}{c|c} R_{TL} & R_{TR} \\ \hline 0 & R_{BR} \end{array}\right) \\ \hline\hline \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) \end{array}\right) \rightarrow \left(\begin{array}{c} \left(\begin{array}{c|c|c} R_{00} & r_{01} & R_{02} \\ \hline 0 & \rho_{11} & r_{12}^T \\ \hline 0 & 0 & R_{22} \end{array}\right) \\ \hline\hline \left(\begin{array}{c|c|c} U_{00} & u_{01} & U_{02} \\ \hline 0 & \upsilon_{11} & u_{12}^T \\ \hline 0 & 0 & U_{22} \end{array}\right) \end{array}\right)$$

where ρ_{11} and υ_{11} are 1×1

$$\left[\left(\begin{array}{c} \rho_{11} \\ \hline x_{21} \end{array}\right), \tau_1\right] := \textsc{Housev}\left(\begin{array}{c} \rho_{11} \\ \hline\hline u_{01} \\ \hline \upsilon_{11} \end{array}\right)$$

$$\text{Update}\left(\begin{array}{c} r_{12}^T \\ \hline\hline U_{02} \\ \hline u_{12}^T \end{array}\right) := \left(I - \frac{1}{\tau}\left(\begin{array}{c} 1 \\ \hline x_{21} \end{array}\right)\left(\begin{array}{c|c} 1 & x_{21} \end{array}\right)\right)\left(\begin{array}{c} r_{12}^T \\ \hline\hline U_{02} \\ \hline u_{12}^T \end{array}\right)$$

via the steps

$$w_{12}^T := \left(r_{12}^T + x_{21}^H\left(\begin{array}{c} U_{02} \\ \hline u_{12}^T \end{array}\right)\right)/\tau_1$$

$$\left(\begin{array}{c} r_{12}^T \\ \hline\hline U_{02} \\ \hline u_{12}^T \end{array}\right) := \left(\begin{array}{c} r_{12}^T - w_{12}^T \\ \hline\hline \left(\begin{array}{c} U_{02} \\ \hline u_{12}^T \end{array}\right) - x_{21} w_{12}^T \end{array}\right)$$

Continue with

$$\left(\begin{array}{c} \left(\begin{array}{c|c} R_{TL} & R_{TR} \\ \hline 0 & R_{BR} \end{array}\right) \\ \hline\hline \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) \end{array}\right) \leftarrow \left(\begin{array}{c} \left(\begin{array}{c|c|c} R_{00} & r_{01} & R_{02} \\ \hline 0 & \rho_{11} & r_{12}^T \\ \hline 0 & 0 & R_{22} \end{array}\right) \\ \hline\hline \left(\begin{array}{c|c|c} U_{00} & u_{01} & U_{02} \\ \hline 0 & \upsilon_{11} & u_{12}^T \\ \hline 0 & 0 & U_{22} \end{array}\right) \end{array}\right)$$

endwhile

Figure 1: Algorithm for computing $[R] := \mathrm{QR}(R, U)$.

Solution

The majority of the computation goes into

$$w_{12}^T := \left(r_{12}^T + x_{21}^H \left(\frac{U_{02}}{u_{12}^T} \right) \right) / \tau_1$$

and

$$\left(\frac{U_{02}}{u_{12}^T} \right) - x_{21} w_{12}^T .$$

During the kth iteration this means a matrix-vector multiplication $x_{21}^H \left(\frac{U_{02}}{u_{12}^T} \right)$ and a rank-1 update with matrix $\left(\frac{U_{02}}{u_{12}^T} \right)$ which is of size $(k+1) \times (n-k-1)$, for a cost of $4(k+1)(n-k)$ flops. Thus the total cost is approximately

$$\begin{aligned} \sum_{k=0}^{n-1} 4(k+1)(n-k-1) &\approx 4 \int_0^n (k+1)(n-k-1) dk \\ &= \frac{2}{3} n^3 - 4n \approx \frac{2}{3} n^3 . \end{aligned}$$

The standard Householder QR algorithm would have cost

$$2(2n)n^2 - \frac{2}{3} n^3 = \frac{10}{3} n^3 ,$$

which is five times slower.

Problem 3

a) Prove that $\hat{V}^{(k)}$ and $\hat{A}^{(k)}$ from the subspace iteration are the same as $V^{(k)}$ and $A^{(k)}$ from the QR algorithm.

Proof

For reference, let us reprint the two algorithms side by side.

Subspace iteration

$\hat{A}^{(0)} := A$
$\hat{V}^{(0)} := I$
for $k = 0$ to convergence **do**
  $A\hat{V}^{(k)} \rightarrow \hat{V}^{(k+1)} \hat{R}^{(k+1)}$ (QR factorization)
  $\hat{A}^{(k+1)} := \hat{V}^{(k+1)T} A \hat{V}^{(k+1)}$
end for

QR algorithm

$A^{(0)} := A$
$V^{(0)} := I$
for $k = 0$ to convergence **do**
  $A^{(k)} \rightarrow Q^{(k+1)} R^{(k+1)}$ (QR factorization)
  $A^{(k+1)} := R^{(k+1)} Q^{(k+1)}$
  $V^{(k+1)} := V^{(k)} Q^{(k+1)}$
end for

Figure 2: Basic subspace iteration and basic QR algorithm.

Base case: $k = 0$. Both algorithms start identically with $\hat{A}^{(0)} = A^{(0)} = A$ and $\hat{V}^{(0)} = V^{(0)} = I$, so both algorithms enter the for loop identically. Our first computation in the subspace iteration is to compute

$$[\hat{V}^{(1)}, \hat{R}^{(1)}] = \mathrm{QR}(A\hat{V}^{(0)}) = \mathrm{QR}(A)\,,$$

since $\hat{V}^{(0)} = I$. The first computation of the QR algorithm is to obtain

$$[\hat{Q}^{(1)}, \hat{R}^{(1)}] = \mathrm{QR}(A^{(0)}) = \mathrm{QR}(A)\,,$$

since $A^{(0)} = A$. So at this point we have $\hat{V}^{(1)} = Q^{(1)}$ and $\hat{R}^{(1)} = R^{(1)}$. In the next step of the subspace iteration we compute

$$\begin{aligned}\hat{A}^{(1)} &:= \hat{V}^{(1)T} A \hat{V}^{(1)} = \cancel{\hat{V}^{(1)T}\hat{V}^{(1)}}\hat{R}^{(1)}\hat{V}^{(1)} = \hat{R}^{(1)}\hat{V}^{(1)} \\ &= R^{(1)}Q^{(1)} = A^{(1)}\,,\end{aligned}$$

since we just factored $A = \hat{V}^{(1)}\hat{R}^{(1)}$, $\hat{V}^{(1)} = Q^{(1)}$, and $\hat{R}^{(1)} = R^{(1)}$, which is exactly the second step of the QR algorithm. Finally, the QR algorithm computes

$$V^{(1)} := V^{(0)}Q^{(1)} = Q^{(1)}\,,$$

since $V^{(0)} = I$, which is exactly the result we got from the subspace iteration. So after one iteration, we get the desired result: $\hat{V}^{(1)} = V^{(1)}$ and $\hat{A}^{(1)} = A^{(1)}$.

Inductive step: Assume that $\hat{V}^{(k)} = V^{(k)}$ and $\hat{A}^{(k)} = A^{(k)}$ holds for some k, we need to show that it holds for $k+1$. From the inductive hypothesis, we can assume the following at step $k+1$:

1. $AV^{(k-1)} = V^{(k)}R^{(k)}$
2. $A^{(k)} = V^{(k)T}AV^{(k)}$
3. $A^{(k-1)} = Q^{(k)}R^{(k)}$
4. $A^{(k)} = R^{(k)}Q^{(k)}$
5. $V^{(k)} = V^{(k-1)}Q^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$

From these assumptions, we can derive some helpful identities. Combining the second and last properties yields,

$$A^{(k)} = Q^{(k)T}\cdots Q^{(1)T}AQ^{(1)}\cdots Q^{(k)}\,. \tag{1}$$

Moving on to our $k+1$ step, anywhere we see a (k) term, we can apply our assumptions, thus step 1 of the QR algorithm becomes (via Eq. 1)

$$Q^{(k)T}\cdots Q^{(1)T}AQ^{(1)}\cdots Q^{(k)} = Q^{(k+1)}R^{(k+1)}\,. \tag{2}$$

From assumption 5, step 1 of the subspace iteration becomes

$$AQ^{(1)}\cdots Q^{(k)} = \hat{V}^{(k+1)}\hat{R}^{(k+1)}\,. \tag{3}$$

There is a strong similarity in structure between these two equations, and in fact, if we multiply Eq. 3 on the left by $V^{(k)T}$, we get

$$Q^{(k)T}\cdots Q^{(1)T}AQ^{(1)}\cdots Q^{(k)} = A^{(k)} = \hat{V}^{(k+1)}\hat{R}^{(k+1)}\,, \tag{4}$$

which is just the QR factorization of $A^{(k)}$. Assuming $A^{(k)}$ has linearly independent columns, the QR factorization is unique and we must have

$$Q^{(k)T}\cdots Q^{(1)T}\hat{V}^{(k+1)}\hat{R}^{(k+1)} = Q^{(k+1)}R^{(k+1)}\,.$$

Multiplying both sides on the left by $V^{(k)}$,

$$\hat{V}^{(k+1)}\hat{R}^{(k+1)} = Q^{(1)}\cdots Q^{(k+1)}R^{(k+1)} = V^{(k+1)}R^{(k+1)}\,.$$

Since the QR factorization is unique, we must have

$$\hat{V}^{(k+1)} = V^{(k+1)}\,,$$

and

$$\hat{R}^{(k+1)} = R^{(k+1)}\,.$$

Finally, we need to prove that $\hat{A}^{(k+1)} = A^{(k+1)}$. Recall that $A^{(k)} = Q^{(k+1)}R^{(k+1)}$. Looking at step 2 of the subspace iteration,

$$\begin{aligned}
\hat{A}^{(k+1)} &= Q^{(k+1)T}\cdots Q^{(1)T}AQ^{(1)}\cdots Q^{(k+1)} \\
&= Q^{(k+1)T}\cdots \cancel{Q^{(1)T}Q^{(1)}}\ \underbrace{R^{(1)}Q^{(1)}}_{A^{(1)}=Q^{(2)}R^{(2)}}\ \cdots Q^{(k+1)} \\
&= Q^{(k+1)T}\cdots \cancel{Q^{(2)T}Q^{(2)}}\ \underbrace{R^{(2)}Q^{(2)}}_{A^{(2)}=Q^{(3)}R^{(3)}}\ \cdots Q^{(k+1)} \\
&\ \ \vdots \\
&= \cancel{Q^{(k+1)T}Q^{(k+1)}}R^{(k+1)}Q^{(k+1)} \\
&= R^{(k+1)}Q^{(k+1)} = A^{(k+1)}\,.
\end{aligned}$$

So by the principle of mathematical induction, $\hat{R}^{(k)} = R^{(k)}$, $\hat{V}^{(k)} = V^{(k)}$, and $\hat{A}^{(k)} = A^{(k)}$ for all k.

Extending to shifted QR algorithm We would follow a very similar process to extend the results to a shifted QR algorithm, printed below for reference.

Subspace iteration

$\hat{A}^{(0)} := A$
$\hat{V}^{(0)} := I$
for $k = 0$ to convergence **do**
  $\hat{\mu}_k := v_{n-1}^{(k)T} A_{n-1}^{(k)}$
  $(A - \hat{\mu}_k I)\hat{V}^{(k)} \to \hat{V}^{(k+1)}\hat{R}^{(k+1)}$ (QR factorization)
  $\hat{A}^{(k+1)} := \hat{V}^{(k+1)T} A \hat{V}^{(k+1)}$
end for

QR algorithm

$A^{(0)} := A$
$V^{(0)} := I$
for $k = 0$ to convergence **do**
  $\mu_k := \alpha_{n-1,n-1}^{(k)}$
  $A^{(k)} - \mu_k I \to Q^{(k+1)}R^{(k+1)}$ (QR factorization)
  $A^{(k+1)} := R^{(k+1)}Q^{(k+1)} + \mu_k I$
  $V^{(k+1)} := V^{(k)}Q^{(k+1)}$
end for

Figure 3: Basic shifted subspace iteration and basic shifted QR algorithm.

The first step would be to follow the algorithms through for $k = 0$ which would show that

$$\hat{\mu}_0 = \mu_0 = \alpha_{n-1,n-1}^{(0)}\,,$$

$$\hat{V}^{(1)} = V^{(1)} = Q^{(1)}\,,$$

and

$$\hat{A}^{(1)} = A^{(1)} = R^{(1)}Q^{(1)} + \mu_0 I\,.$$

Then we would need to propose the inductive hypothesis and compose a list of assumptions as before:

1. $\mu_{k-1} = v_{n-1}^{(k-1)T} A_{n-1}^{(k-1)}$

2. $(A - \mu_{k-1}I)V^{(k-1)} = V^{(k)}R^{(k)}$
3. $A^{(k)} = V^{(k)T}AV^{(k)}$
4. $\mu_{k-1} = \alpha^{(k-1)}_{n-1,n-1}$
5. $A^{(k-1)} - \mu_{k-1}I = Q^{(k)}R^{(k)}$
6. $A^{(k)} = R^{(k)}Q^{(k)} + \mu_{k-1}I$
7. $V^{(k)} = V^{(k-1)}Q^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}$

Again, we could combine terms to get

$$A^{(k)} = Q^{(k)T}\cdots Q^{(1)T}AQ^{(1)}\cdots Q^{(k)}\,. \tag{5}$$

We could substitute this into step 2 of the shifted QR algorithm and compare the results with step 2 of the shifted subspace iteration. We could multiply step two of the subspace iteration on the left by $V^{(k)T}$ and use the uniqueness of QR factorization to argue that

$$\hat{V}^{(k+1)} = V^{(k+1)}\,,$$

and

$$\hat{R}^{(k+1)} = R^{(k+1)}\,.$$

This would also allow us to see that $\hat{\mu}_k = \mu_k = \alpha^{(k)}_{n-1,n-1}$. Finally, we could collapse the third step of the shifted subspace iteration to the third step of the shifted QR algorithm, proving that

$$\hat{A}^{(k+1)} = A^{(k+1)}\,.$$

Thus, the theorem would hold by the principle of mathematical induction, and we will have proved our point.

1 Problem 4

With $\lambda_{11} = 1$, the main source of error in our calculations is from

$$\chi_1 := \beta_1 - l_{10}^T x_0\,. \tag{6}$$

According to Theorem 4.1, we wish to push all of the error into the L matrix for easy comparison between results. Theorem 3.12 however, does not enumerate exactly the correct result to allow us to do this. Equation 6 fits the pattern of "$\mu := \alpha - (x^T y)$." R1-B puts would make it possible to put the error into both L and the b vector, while R1-F and R2-F could put the error into either x or b depending on how they were considered. The remaining results fit the pattern "$\nu := (\alpha - (x^T y))/\lambda$." Of course, in our case, $\lambda = 1$, so we don't need that final divide. Omitting the division operation, we can modify R4-B to fit our purposes, in this case we get

$$\check{\nu} = \frac{\alpha - (x + \delta x)^T y}{\lambda + \delta\lambda}, \text{ where } |\delta\lambda| \leq \gamma_1 |\lambda| \text{ and } |\delta x| \leq \gamma_n |x|\,.$$

Note, the division operation is not committed, it is only shown here to show the error accumulation in the "ones" on the diagonal in lambda. The other alternative was to throw all of the error on x, the part of L below the diagonal, but this method makes for a more direct comparison of results.

With this new error pattern established, we can modify our results to compute a new error bound. Clearly the first change we can make is in Fig. 4.1 where we can replace $\chi_1 := (\beta_1 - l_{10}^T x_0)\lambda_{11}$ with simply $\chi_1 := (\beta_1 - l_{10}^T x_0)$. This correction should be made anywhere this result is computed. In step 1 of Section 4.2, our backward error analysis becomes

$$(L + \Delta L)\check{x} = b \text{ with } |\Delta L| \leq \gamma_{n-1} |L|\,,$$

where the $\max(\gamma_2, \gamma_{n-1})$ term was replaced with simply γ_{n-1}. Also, the error postcondition becomes

$$(L + \Delta L)\check{x} = b \wedge |\Delta L| \leq \gamma_{n-1} |L| \wedge m(x) = n \,.$$

Similarly, the error invariant becomes,

$$(L_{TL} + \Delta L_{TL})\check{x} = b_T \wedge |\Delta L_{TL}| \leq \gamma_{n-1} |L_{TL}| \wedge m(x_T) = k \,.$$

The bottom half of Equation (4.2) becomes

$$\left| \left(\begin{array}{c|c} \Delta L_{00} & 0 \\ \hline \delta l_{10}^T & \delta\lambda_{11} \end{array} \right) \right| \leq \gamma_k \left| \left(\begin{array}{c|c} L_{00} & 0 \\ \hline l_{10}^T & 1 \end{array} \right) \right| .$$

Now, our error variables must satisfy

$$\begin{array}{ll} b_0 = (L_{00} + \Delta L_{00})\check{x}_0 & \wedge |\Delta L_{00}| \leq \gamma_k |L_{00}| \\ \beta_1 = (l_{10}^T + \delta l_{10}^T)\check{x}_0 + (1 + \delta\lambda_{11})\check{\chi}_1 & \wedge \left|(\delta l_{10}^T | \delta\lambda_{11})\right| \leq \gamma_k \left|(l_{10}^T | 1)\right| \end{array}$$

Looking now at Figure 4.2, we need to make the following changes. The error postcondition in line 1b should be replaced with the one described above. The error invariants in 2a, 2b, 2c, and 2d should be replaced with the one described above. Lines 5a, 6, 7, and 8 should be modified to reflect the fact that $\lambda_{11} = 1$. In lines 6, 7, and 8 replace $\max(\gamma_2, \gamma_{k-1})$ with γ_{k-1} and $\max(\gamma_2, \gamma_k)$ with γ_k.