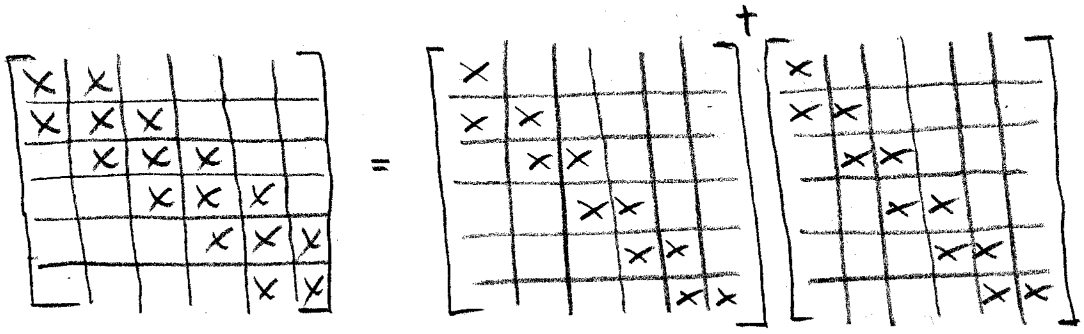
5.4 Implementing Whitening Decision Feedback

- The use of Cholesky factorization does not sound promising, but it's reasonably straightforward.

- Brute force application of Cholesky to a block of N symbols requires about $N^3 K^3$ operations [Proa95].

- The block tridiagonal nature of $C^{+}RC$ can be exploited to reduce computation.



$$C^{+}RC = F^{+} \quad F$$

- [Wei96] shows how the blocks of F can be computed recursively, in a series of K×K smaller Cholesky decompositions.

  Effort proportional to NK^3 — a major reduction!

  Per detected bit (NK of them), it's proportional to K^2 — so K^2 times the effort of the conventional single-user detector

- [Alex98] provides a more understandable proof that the recursion converges.

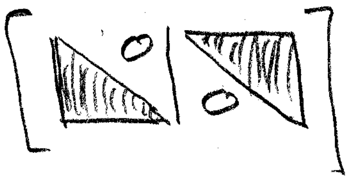
- The formalism of Cholesky decomposition can be skirted if we use Gauss elimination (Appendix O) instead. For some $H = C^{+}RC$, factor by:

① $$\left[\begin{array}{ccccc|ccccc} H_{11} & H_{12} & \cdots & H_{1,K-1} & H_{1K} & 1 & 0 & \cdots & 0 & 0 \\ H_{21} & H_{22} & \cdots & H_{2,K-1} & H_{2K} & 0 & 1 & & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ H_{K-1,1} & H_{K-1,2} & \cdots & H_{K-1,K-1} & H_{K-1,K} & 0 & 0 & \cdots & 1 & 0 \\ H_{K1} & H_{K2} & & H_{K,K-1} & H_{KK} & 0 & 0 & \cdots & 0 & 1 \end{array}\right]$$

② Clear out last col of H by subtracting from row i a multiple $H_{i,K}/H_{KK}$ times last row:

$$\left[\begin{array}{ccccc|cccc} \times & \times & & \times & 0 & 1 & 0 & 0 & -H_{1K}/H_{KK} \\ \times & \times & & \times & 0 & 0 & 1 & 0 & -H_{2K}/H_{KK} \\ & & & & 0 & & & & \\ & & & & 0 & & & & \\ \times & \times & & \times & 0 & 0 & 0 & 1 & -H_{K-1,K}/H_{KK} \\ H_{K1} & H_{K2} & \cdots & H_{K,K-1} & H_{KK} & 0 & 0 & 0 & H_{KK} \end{array}\right]$$

③ Clear out other colums of left side similarly until



L U

(see interactive demo)

So

$$UH = L \qquad H = U^{-1}L = \triangledown \cdot \triangle$$

This is just a disguised form of Cholesky, so computation is the same. Decisions and BER are identical.

- To use it, form

$$\underline{z} = U\underline{\xi} = UC^{+}RCA\underline{b} + U\underline{\eta}$$

$$= LA\underline{b} + \underline{\alpha}$$

where components of $\underline{\alpha}$ are uncorrelated:

$$R_{\alpha} = N_0 \, U C^{+} R C U^{+} = N_0 \, L U^{+}$$

and LU^{+} is diagonal, since Hermitian and triangular.

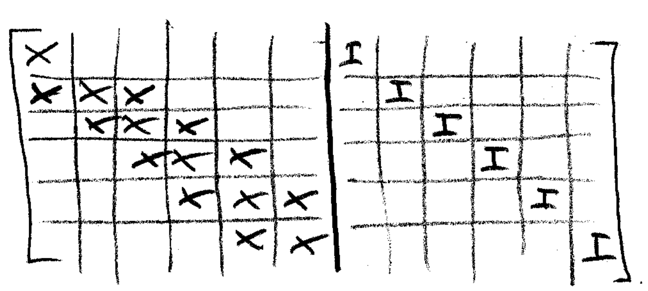
so we can do DF, starting with $b_1(0)$.

→ In fact, we don't have to form U and multiply $\underline{\xi}$ by it. We solve

$$C^{+}RCA\underline{b} \text{ "=" } \underline{\xi}$$

so just reduce $C^{+}RCA|\underline{\xi}$ to lower triangular and back substitute the decisions $\tilde{b}_i$.

- Computation for a block of N symbols is proportional to N^3K^3 if done without thought. Again, block tridiagonal helps:



Computation proportional to NK^3; proportional to K^2 per detected bit.

- Accommodates long codes and time varying C.

- For continuous operation, N unbounded, we would allow a window P symbols into the future and 2K into the past (for DF). In this respect, it resembles DFE, with linear forward section.
  - If short codes, static channels, then the pattern of such a receiver is fixed (and cyclic over K symbols).
  - If long codes, the coeffs appear to need recalculation at every symbol.
  - If short codes, slowly varying channels, is there a way to update, instead of recalculate, the coefficients?

- In asynchronous operation, there is no "last user" of the K, the one that enjoys single user performance with (ideally) no MAI, since more MAI from future rolls in. Ranking by power loses meaning; does error propagation kill the idea?
  - Since all users are first users, we could adopt MMSE, instead of ZF, from the future. Then its a cyclic form of DFE.