positive definite, we merely try to compute its Cholesky factorization using any of the methods given above.

The situation in the context of roundoff error is more interesting. The numerical stability of the Cholesky algorithm roughly follows from the inequality

$$g_{ij}^2 \;\le\; \sum_{k=1}^{i} g_{ik}^2 \;=\; a_{ii}.$$

This shows that the entries in the Cholesky triangle are nicely bounded. The same conclusion can be reached from the equation $\| G \|_2^2 = \| A \|_2$.

The roundoff errors associated with the Cholesky factorization have been extensively studied in a classical paper by Wilkinson (1968). Using the results in this paper, it can be shown that if $\hat{x}$ is the computed solution to $Ax = b$, obtained via the Cholesky process, then $\hat{x}$ solves the perturbed system

$$(A+E)\hat{x} = b \qquad \| E \|_2 \le c_n \mathbf{u} \| A \|_2,$$

where c_n is a small constant that depends upon n. Moreover, Wilkinson shows that if $q_n \mathbf{u}\, \kappa_2(A) \le 1$ where q_n is another small constant, then the Cholesky process runs to completion, i.e., no square roots of negative numbers arise.

It is important to remember that symmetric positive definite linear systems can be ill-conditioned. Indeed, the eigenvalues and singular values of a symmetric positive definite matrix are the same. This follows from (2.4.1) and Theorem 4.2.3. Thus,

$$\kappa_2(A) \;=\; \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

The eigenvalue $\lambda_{\min}(A)$ is the "distance to trouble" in the Cholesky setting. This prompts us to consider a permutation strategy that steers us away from using small diagonal elements that jeopardize the factorization process.

4.2.7 The LDL^T Factorization with Symmetric Pivoting

With an eye towards handling ill-conditioned symmetric positive definite systems, we return to the LDL^T factorization and develop an outer product implementation with pivoting. We first observe that if A is symmetric and P_1 is a permutation, then $P_1 A$ is *not* symmetric. On the other hand, $P_1 A P_1^T$ is symmetric suggesting that we consider the following factorization:

$$P_1 A P_1^T \;=\; \begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ v/\alpha & I_{n-1} \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & \tilde{A} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ v/\alpha & I_{n-1} \end{bmatrix}^T$$

where

$$\tilde{A} \;=\; B - \frac{1}{\alpha} v v^T.$$

Note that with this kind of *symmetric pivoting*, the new (1,1) entry α is some diagonal entry a_{ii}. Our plan is to choose P_1 so that α is the largest of A's diagonal entries. If we apply the same strategy recursively to $\tilde{A}$ and compute

$$\tilde{P}\tilde{A}\tilde{P}^T \;=\; \tilde{L}\tilde{D}\tilde{L}^T,$$

then we emerge with the factorization

$$PAP^T = LDL^T \tag{4.2.10}$$

where

$$P = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P} \end{bmatrix} P_1, \qquad L = \begin{bmatrix} 1 & 0 \\ v/\alpha & \tilde{L} \end{bmatrix}, \qquad D = \begin{bmatrix} \alpha & 0 \\ 0 & \tilde{D} \end{bmatrix}.$$

By virtue of this pivot strategy,

$$d_1 \ge d_2 \ge \cdots \ge d_n > 0.$$

Here is a nonrecursive implementation of the overall algorithm:

Algorithm 4.2.2 (Outer Product LDL^T with Pivoting) Given a symmetric positive semidefinite $A \in \mathbb{R}^{n\times n}$, the following algorithm computes a permutation P, a unit lower triangular L, and a diagonal matrix $D = \text{diag}(d_1, \ldots, d_n)$ so $PAP^T = LDL^T$ with $d_1 \ge d_2 \ge \cdots \ge d_n > 0$. The matrix element a_{ij} is overwritten by d_i if $i = j$ and by ℓ_{ij} if $i > j$. $P = P_1 \cdots P_n$ where P_k is the identity with rows k and $piv(k)$ interchanged.

for $k = 1{:}n$
    $piv(k) = j$ where $a_{jj} = \max\{a_{kk}, \ldots, a_{nn}\}$
    $A(k,:) \leftrightarrow A(j,:)$
    $A(:,k) \leftrightarrow A(:,j)$
    $\alpha = A(k,k)$
    $v = A(k+1{:}n, k)$
    $A(k+1{:}n, k) = v/\alpha$
    $A(k+1{:}n, k+1{:}n) = A(k+1{:}n, k+1{:}n) - vv^T/\alpha$
end

If symmetry is exploited in the outer product update, then $n^3/3$ flops are required. To solve $Ax = b$ given $PAP^T = LDL^T$, we proceed as follows:

$$Lw = Pb, \qquad Dy = w, \qquad L^T z = y, \qquad x = P^T z.$$

We mention that Algorithm 4.2.2 can be implemented in a way that only references the lower trianglar part of A.

It is reasonable to ask why we even bother with the LDL^T factorization given that it appears to offer no real advantage over the Cholesky factorization. There are two reasons. First, it is more efficient in narrow band situations because it avoids square roots; see §4.3.6. Second, it is a graceful way to introduce factorizations of the form

$$PAP^T = \left(\begin{array}{c}\text{lower}\\ \text{triangular}\end{array}\right) \times \left(\begin{array}{c}\text{simple}\\ \text{matrix}\end{array}\right) \times \left(\begin{array}{c}\text{lower}\\ \text{triangular}\end{array}\right)^T,$$

where P is a permutation arising from a symmetry-exploiting pivot strategy. The symmetric indefinite factorizations that we develop in §4.4 fall under this heading as does the "rank revealing" factorization that we are about to discuss for semidefinite problems.