51

Matrix Factorizations and Direct Solution of Linear Systems

Christopher Beattie
Virginia Polytechnic Institute and State University

The need to solve systems of linear equations arises often within diverse disciplines of science, engineering, and finance. The expression "direct solution of linear systems" refers generally to computational strategies that are able to produce solutions to linear systems after a predetermined number of arithmetic operations that depends only on the structure and dimension of the coefficient matrix. The evolution of computers has and continues to influence the development of these strategies and has also fostered particular styles of perturbation analysis suited to illuminating their behavior. Some general themes have become dominant, as a result; others have been pushed aside. For example, Cramer's Rule may be properly thought of as a direct solution strategy for solving linear systems; however as normally manifested, it requires a much larger number of arithmetic operations than Gauss elimination and is generally much more susceptible to the deleterious effects of rounding. Most current approaches for the direct solution of a linear system, $A\mathbf{x} = \mathbf{b}$, are patterned after Gauss elimination and favor an initial phase that partially decouples the system of equations: zeros are introduced systematically into the coefficient matrix, transforming it into triangular form; the resulting triangular system is easily solved. The entire process can be viewed in this way:

1. Find invertible matrices $\{S_i\}_{i=1}^{\rho}$ such that $S_\rho \dots S_2 S_1 A = U$ is triangular; then
2. Calculate a modified right-hand side $\mathbf{y} = S_\rho \dots S_2 S_1 \mathbf{b}$; and then
3. Determine the solution set to the triangular system $U\mathbf{x} = \mathbf{y}$.

The matrices S_1, S_2, $\dots$ S_ρ are typically either row permutations of lower triangular matrices (Gauss transformations) or unitary matrices. In either case, inverses are readily available. Evidently, A can be written as $A = NU$, where $N = (S_\rho \dots S_2 S_1)^{-1}$. A solution framework may be built around the availability of decompositions such as this:

1. Find a decomposition $A = NU$ such that U is triangular and $N\mathbf{y} = \mathbf{b}$ is easily solved;
2. Solve $N\mathbf{y} = \mathbf{b}$; then
3. Determine the solution set to the triangular system $U\mathbf{x} = \mathbf{y}$.

51.1 Perturbations of Linear Systems

In the computational environment afforded by current computers, the finite representation of real numbers creates a small but persistent source of errors that may on occasion severely degrade the overall accuracy of a calculation. This effect is of fundamental concern in assessing strategies for solving linear systems.

Rounding errors can be introduced into the solution process for linear systems often before any calculations are performed — as soon as data are stored within the computer and represented within the internal floating point number system of the computer. Further errors that may be introduced in the course of computation often may be viewed in aggregate effectively as an additional contribution to this initial representation error. Inevitably, the linear system for which a solution is computed will deviate slightly from the "true" linear system and it becomes of critical interest to determine whether such deviations will have a significant effect on the accuracy of the final computed result.

Definitions:

Let $A \in \mathbb{C}^{n\times n}$ be a nonsingular matrix, $\mathbf{b} \in \mathbb{C}^n$, and then denote by $\hat{\mathbf{x}} = A^{-1}\mathbf{b}$ the unique solution of the linear system $A\mathbf{x} = \mathbf{b}$.

Given **data perturbations** $\delta A \in \mathbb{C}^{n\times n}$ and $\delta\mathbf{b} \in \mathbb{C}^n$ to A and $\mathbf{b}$, respectively, the **solution perturbation**, $\delta\mathbf{x} \in \mathbb{C}^n$ satisfies the associated **perturbed linear system** $(A + \delta A)(\hat{\mathbf{x}} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ (presuming then that the perturbed system is consistent).

For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$, the **residual vector** associated with $\tilde{\mathbf{x}}$ as an approximate solution to the linear system $A\mathbf{x} = \mathbf{b}$ is defined as $\mathbf{r}(\tilde{\mathbf{x}}) = \mathbf{b} - A\tilde{\mathbf{x}}$.

For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$, the associated (norm-wise) **relative backward error of the linear system** $A\mathbf{x} = \mathbf{b}$ (with respect to the the p-norm, for $1 \le p \le \infty$) is

$$\eta_p(A, \mathbf{b}; \tilde{\mathbf{x}}) = \min \left\{ \varepsilon \;\middle|\; \begin{array}{c} \text{there exist } \delta A, \delta\mathbf{b} \text{ such that} \\ (A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta\mathbf{b} \text{ with } \begin{array}{l} \|\delta A\|_p \le \varepsilon \|A\|_p \\ \|\delta b\|_p \le \varepsilon \|\mathbf{b}\|_p \end{array} \end{array} \right\}.$$

For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$, the associated **component-wise relative backward error of the linear system** $A\mathbf{x} = \mathbf{b}$ is

$$\omega(A, \mathbf{b}; \tilde{\mathbf{x}}) = \min \left\{ \varepsilon \;\middle|\; \begin{array}{c} \text{there exist } \delta A, \delta\mathbf{b} \text{ such that} \\ (A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta\mathbf{b} \text{ with } \begin{array}{l} |\delta A| \le \varepsilon |A| \\ |\delta\mathbf{b}| \le \varepsilon |\mathbf{b}| \end{array} \end{array} \right\},$$

where the absolute values and inequalities applied to vectors and matrices are interpreted component-wise: for example, $|B| \le |A|$ means $|b_{ij}| \le |a_{ij}|$ for all index pairs i, j.

The (norm-wise) **condition number of the linear system** $A\mathbf{x} = \mathbf{b}$ (with respect to the the p-norm, for $1 \le p \le \infty$) is

$$\kappa_p(A, \hat{\mathbf{x}}) = \|A^{-1}\|_p \frac{\|\mathbf{b}\|_p}{\|\hat{\mathbf{x}}\|_p}.$$

The **matrix condition number** of A (with respect to the the p-norm, for $1 \le p \le \infty$) is

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p.$$

The **Skeel condition number of the linear system** $A\mathbf{x} = \mathbf{b}$ is

$$\mathrm{cond}(A, \hat{\mathbf{x}}) = \frac{\| \, |A^{-1}| \, |A| \, |\hat{\mathbf{x}}| \|_\infty}{\|\hat{\mathbf{x}}\|_\infty}.$$

The **Skeel matrix condition number** is $\mathrm{cond}(A) = \| \, |A^{-1}| \, |A| \, \|_\infty$.

Facts: [Hig02], [SS90]

1. For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$, $\tilde{\mathbf{x}}$ is the exact solution to any one of the following families of perturbed linear systems
$$(A + \delta A_\theta)\tilde{\mathbf{x}} = \mathbf{b} + \delta \mathbf{b}_\theta,$$
where $\theta \in \mathbb{C}$, $\delta \mathbf{b}_\theta = (\theta - 1)\,\mathbf{r}(\tilde{\mathbf{x}})$, $\delta A_\theta = \theta\,\mathbf{r}(\tilde{\mathbf{x}})\tilde{\mathbf{y}}^*$, and $\tilde{\mathbf{y}} \in \mathbb{C}^n$ is any vector such that $\tilde{\mathbf{y}}^*\tilde{\mathbf{x}} = 1$. In particular, for $\theta = 0$, $\delta A = 0$ and $\delta \mathbf{b} = -\mathbf{r}(\tilde{\mathbf{x}})$; for $\theta = 1$, $\delta A = \mathbf{r}(\tilde{\mathbf{x}})\tilde{\mathbf{y}}^*$ and $\delta \mathbf{b} = 0$.
2. (Rigal–Gaches Theorem) For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$,
$$\eta_p(A, \mathbf{b}; \tilde{\mathbf{x}}) = \frac{\|\mathbf{r}(\tilde{\mathbf{x}})\|_p}{\|A\|_p\|\tilde{\mathbf{x}}\|_p + \|\mathbf{b}\|_p}.$$
If $\tilde{\mathbf{y}}$ is the dual vector to $\tilde{\mathbf{x}}$ with respect to the p-norm ($\tilde{\mathbf{y}}^*\tilde{\mathbf{x}} = \|\tilde{\mathbf{y}}\|_q\,\|\tilde{\mathbf{x}}\|_p = 1$ with $\frac{1}{p} + \frac{1}{q} = 1$), then $\tilde{\mathbf{x}}$ is an exact solution to the perturbed linear system $(A + \delta A_{\tilde{\theta}})\tilde{\mathbf{x}} = \mathbf{b} + \delta \mathbf{b}_{\tilde{\theta}}$ with data perturbations as in (1) and $\tilde{\theta} = \frac{\|A\|_p\|\tilde{\mathbf{x}}\|_p}{\|A\|_p\|\tilde{\mathbf{x}}\|_p + \|\mathbf{b}\|_p}$, and as a result
$$\frac{\|\delta A_{\tilde{\theta}}\|_p}{\|A\|_p} = \frac{\|\delta \mathbf{b}_{\tilde{\theta}}\|_p}{\|\mathbf{b}\|_p} = \eta_p(A, \mathbf{b}; \tilde{\mathbf{x}}).$$
3. (Oettli–Prager Theorem) For any $\tilde{\mathbf{x}} \in \mathbb{C}^n$,
$$\omega(A, \mathbf{b}; \tilde{\mathbf{x}}) = \max_i \frac{|r_i|}{(|A|\ |\tilde{\mathbf{x}}|\ +\ |\mathbf{b}|)_i}.$$
If $D_1 = \operatorname{diag}\left(\frac{r_i}{(|A|\ |\tilde{\mathbf{x}}|\ +\ |\mathbf{b}|)_i}\right)$ and $D_2 = \operatorname{diag}(sign(\tilde{\mathbf{x}})_i)$, then $\tilde{\mathbf{x}}$ is an exact solution to the perturbed linear system $(A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta \mathbf{b}$ with $\delta A = D_1\,|A|\,D_2$ and $\delta \mathbf{b} = -D_1\,|\mathbf{b}|$
$$|\delta A| \le \omega(A, \mathbf{b}; \tilde{\mathbf{x}})\,|A| \quad \text{and} \quad |\delta \mathbf{b}| \le \omega(A, \mathbf{b}; \tilde{\mathbf{x}})\,|A|$$
and no smaller constant can be used in place of $\omega(A, \mathbf{b}; \tilde{\mathbf{x}})$.
4. The reciprocal of $\kappa_p(A)$ is the smallest norm-wise relative distance of A to a singular matrix, i.e.,
$$\frac{1}{\kappa_p(A)} = \min\left\{ \frac{\|\delta A\|_p}{\|A\|_p} \;\middle|\; A + \delta A \text{ is singular} \right\}.$$
In particular, the perturbed coefficient matrix $A + \delta A$ is nonsingular if
$$\frac{\|\delta A\|_p}{\|A\|_p} < \frac{1}{\kappa_p(A)}.$$
5. $1 \le \kappa_p(A, \hat{\mathbf{x}}) \le \kappa_p(A)$ and $1 \le \operatorname{cond}(A, \hat{\mathbf{x}}) \le \operatorname{cond}(A) \le \kappa_\infty(A)$.
6. $\operatorname{cond}(A) = \min\{\kappa_\infty(DA) \mid D \text{ diagonal}\}$.
7. If $\delta A = 0$, then
$$\frac{\|\delta \mathbf{x}\|_p}{\|\hat{\mathbf{x}}\|_p} \le \kappa_p(A, \hat{\mathbf{x}})\frac{\|\delta \mathbf{b}\|_p}{\|\mathbf{b}\|_p}.$$
8. If $\delta \mathbf{b} = 0$ and $A + \delta A$ is nonsingular, then
$$\frac{\|\delta \mathbf{x}\|_p}{\|\hat{\mathbf{x}} + \delta \mathbf{x}\|_p} \le \kappa_p(A)\frac{\|\delta A\|_p}{\|A\|_p}.$$
9. If $\|\delta A\|_p \le \epsilon\|A\|_p$, $\|\delta \mathbf{b}\|_p \le \epsilon\|\mathbf{b}\|_p$, and $\epsilon < \frac{1}{\kappa_p(A)}$, then
$$\frac{\|\delta \mathbf{x}\|_p}{\|\hat{\mathbf{x}}\|_p} \le \frac{2\,\epsilon\,\kappa_p(A)}{1 - \epsilon\,\kappa_p(A)}.$$

10. If $|\delta A| \le \epsilon |A|$, $|\delta \mathbf{b}| \le \epsilon\, |\mathbf{b}|$, and $\epsilon < \dfrac{1}{\mathrm{cond}(A)}$, then

$$\frac{\|\delta \mathbf{x}\|_\infty}{\|\hat{\mathbf{x}}\|_\infty} \le \frac{2\,\epsilon\,\mathrm{cond}(A, \hat{\mathbf{x}})}{1 - \epsilon\,\mathrm{cond}(A)}.$$

Examples:

1. Let $A = \begin{bmatrix} 1000 & 999 \\ 999 & 998 \end{bmatrix}$ so $A^{-1} = \begin{bmatrix} -998 & 999 \\ 999 & -1000 \end{bmatrix}$. Then $\|A\|_1 = \|A^{-1}\|_1 = 1999$ so that $\kappa_1(A) \approx 3.996 \times 10^6$. Consider

$$\mathbf{b} = \begin{bmatrix} 1999 \\ 1997 \end{bmatrix} \text{ associated with a solution } \hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

A perturbation of the right-hand side $\delta \mathbf{b} = \begin{bmatrix} -0.01 \\ 0.01 \end{bmatrix}$ constitutes a relative change in the right-hand side of $\dfrac{\|\delta \mathbf{b}\|_1}{\|\mathbf{b}\|_1} \approx 5.005 \times 10^{-6}$ yet it produces a perturbed solution $\hat{\mathbf{x}} + \delta \mathbf{x} = \begin{bmatrix} 20.97 \\ -18.99 \end{bmatrix}$ constituting a relative change $\dfrac{\|\delta \mathbf{x}\|_1}{\|\hat{\mathbf{x}}\|_1} = 19.98 \le 20 = \kappa_1(A) \dfrac{\|\delta \mathbf{b}\|_1}{\|\mathbf{b}\|_1}$. The bound determined by the condition number is very nearly achieved. Note that the same perturbed solution $\hat{\mathbf{x}} + \delta \mathbf{x}$ could be produced by a change in the coefficient matrix

$$\delta A = \tilde{\mathbf{r}} \tilde{\mathbf{y}}^* = -\begin{bmatrix} -0.01 \\ 0.01 \end{bmatrix} \begin{bmatrix} \dfrac{1}{39.96} & -\dfrac{1}{39.96} \end{bmatrix} = (1/3996) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

constituting a relative change $\dfrac{\|\delta A\|_1}{\|A\|_1} \approx 2.5 \times 10^{-7}$. Then $(A + \delta A)(\hat{\mathbf{x}} + \delta \mathbf{x}) = \mathbf{b}$.

2. Let $n = 100$ and A be tridiagonal with diagonal entries equal to -2 and all superdiagonal and subdiagonal entries equal to 1 (associated with a centered difference approximation to the second derivative). Let $\mathbf{b}$ be a vector with a quadratic variation in entries

$$b_k = (k-1)(100-k)/10{,}000.$$

Then

$$\kappa_2(A, \hat{\mathbf{x}}) \approx 1, \quad \text{but} \quad \kappa_2(A) \approx 4.1336 \times 10^3.$$

Since the elements of $\mathbf{b}$ do not have an exact binary representation, the linear system that is presented to any computational algorithm will be $A\mathbf{x} = \mathbf{b} + \delta \mathbf{b}$ with $\|\delta \mathbf{b}\|_2 \le \epsilon \|\mathbf{b}\|_2$, where ϵ is the unit roundoff error. For example, if the linear system data is stored in IEEE single precision format, $\epsilon \approx 6 \times 10^{-8}$. The matrix condition number, $\kappa_2(A)$, would yield a bound of $(6 \times 10^{-8})(4.1336 \times 10^3) \approx 2.5 \times 10^{-4}$ anticipating the loss of more than 4 significant digits in solution components even if all computations were done on the stored data with no further error. However, the condition number of the linear system, $\kappa_2(A, \hat{\mathbf{x}})$, is substantially smaller and the predicted error for the system is roughly the same as the initial representation error $\approx 6 \times 10^{-8}$, indicating that the solution will be fairly insensitive to the consequences of rounding of the right-hand side data—assuming no further errors occur. But, in fact, this conclusion remains true even if further errors occur, if whatever computational algorithm that is used produces small backward error, as might be asserted if, say, a final residual satisfies $\|\mathbf{r}\|_2 \le \mathcal{O}(\epsilon)\, \|\mathbf{b}\|_2$. This situation changes substantially if the right-hand side is changed to

$$b_k = (-1)^k (k-1)(100-k)/10{,}000,$$

which only introduces a sign variation in $\mathbf{b}$. In this case, $\kappa_2(A, \hat{\mathbf{x}}) \approx \kappa_2(A)$, and the components of the computed solution can be expected to lose about 4 significant digits purely on

the basis of errors that are made in the initial representation. Additional errors made in the course of the computation can hardly be expected to improve this situation.

51.2 Triangular Linear Systems

Systems of linear equations for which the unknowns may be solved for one at a time in sequence may be reordered to produce linear systems with triangular coefficient matrices. Such systems can be solved both with remarkable accuracy and remarkable efficiency. Triangular systems are the archetype for easily solvable systems of linear equations. As such, they often constitute an intermediate goal in strategies for solving linear systems.

Definitions:

A linear system of equations $T\mathbf{x} = \mathbf{b}$ with $T \in \mathbb{C}^{n\times n}$ (representing n equations in n unknowns) is a **triangular system** if $T = [t_{ij}]$ is either an **upper triangular matrix** ($t_{ij} = 0$ for $i > j$) or a **lower triangular matrix** ($t_{ij} = 0$ for $i < j$).

Facts: [Hig02], [GV96]

1. [GV96, pp. 88–90]

Algorithm 1: Row-wise forward substitution for solving lower triangular system
Input: $L = [\ell_{ij}] \in \mathbb{R}^{n\times n}$ with $\ell_{kj} = 0$ for $k < j$; $\mathbf{b} \in \mathbb{R}^n$
Output: solution vector $\mathbf{x} \in \mathbb{R}^n$ that satisfies $L\mathbf{x} = \mathbf{b}$

$x_1 \leftarrow b_1/\ell_{1,1}$
for $k = 2$ to n
  $x_k \leftarrow (b_k - L_{k,1:k-1} \cdot x_{1:k-1})/\ell_{k,k}$

Algorithm 2: Column-wise back substitution for solving upper triangular system
Input: $U = [u_{ij}] \in \mathbb{R}^{n\times n}$ with $u_{kj} = 0$ for $k > j$; $\mathbf{b} \in \mathbb{R}^n$
Output: solution vector $\mathbf{x} \in \mathbb{R}^n$ that satisfies $U\mathbf{x} = \mathbf{b}$

for $k = n$ down to 2 in steps of -1,
  $x_k \leftarrow b_k/u_{k,k}$
  $b_{1:k-1} \leftarrow b_{1:k-1} - x_k U_{1:k-1,k}$
$x_1 \leftarrow b_1/u_{1,1}$

2. Algorithm 1 involves as a core calculation dot products of portions of coefficient matrix rows with portions of the emerging solution vector. This can incur a performance penalty for large n from accumulation of dot products using a scalar recurrence. A "column-wise" reformulation may have better performance for large n. Algorithm 2 is such a "column-wise" formulation for upper triangular systems.
3. An efficient and reliable implementation for the solution of triangular systems is offered as part of the standard BLAS software library in `xTRSz` (see Chapter 92), where `x`=`S`, `D`, `C`, or `Z` according to whether data are single or double precision real, or single or double precision complex floating point numbers, respectively, and `z`=`V` or `M` according to whether a single system of equations is to be solved or multiple systems (sharing the same coefficient matrix) are to be solved, respectively.

4. The solution of triangular systems using either Algorithm 1 or 2 is *component-wise backward stable*. In particular the computed result, $\tilde{\mathbf{x}}$, produced either by Algorithm 1 or 2 in solving a triangular system, $T\mathbf{x} = \mathbf{b}$, will be the exact result of a perturbed system $(T + \delta T)\tilde{\mathbf{x}} = \mathbf{b}$, where $|\delta T| \leq \dfrac{n\,\epsilon}{1 - n\,\epsilon}|T|$ and ϵ is the unit roundoff error.
5. The error in the solution of a triangular system, $T\mathbf{x} = \mathbf{b}$, using either Algorithm 1 or 2 satisfies
$$\frac{\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|_\infty}{\|\hat{\mathbf{x}}\|_\infty} \leq \frac{n\,\epsilon\,\mathrm{cond}(T, \hat{\mathbf{x}})}{1 - n\,\epsilon\,(\mathrm{cond}(T) + 1)}.$$
6. If $T = [t_{ij}]$ is a lower triangular matrix satisfying $|t_{ii}| \geq |t_{ij}|$ for $j \leq i$, the computed solution to the linear system $T\mathbf{x} = \mathbf{b}$ produced by either Algorithm 1 or the variant of Algorithm 2 for lower triangular systems satisfies
$$|\hat{x}_i - \tilde{x}_i| \leq \frac{2^i\,n\,\epsilon}{1 - n\,\epsilon} \max_{j \leq i} |\tilde{x}_j|,$$
where $\tilde{x}_i$ are the components of the computed solution, $\tilde{\mathbf{x}}$, and $\hat{x}_i$ are the components of the exact solution, $\hat{\mathbf{x}}$. Although this bound degrades exponentially with i, it shows that early solution components will be computed to high accuracy relative to those components already computed.

Examples:

1. Use Algorithm 2 to solve the triangular system
$$\begin{bmatrix} 1 & 2 & -3 \\ 0 & 2 & -6 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$
$k = 3$ step: Solve for $x_3 = 1/3$. Update right-hand side:
$$\begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - (1/3) \begin{bmatrix} -3 \\ -6 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}.$$
$k = 2$ step: Solve for $x_2 = 3/2$. Update right-hand side:
$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix} - (3/2) \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$
$k = 1$ step: Solve for $x_1 = -1$.
2. [Hig02, p. 156] For $\epsilon > 0$, consider $T = \begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & \varepsilon & 0 \\ 0 & 1 & 1 \end{bmatrix}$. Then $T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & \frac{1}{\varepsilon} & 0 \\ 1 & -\frac{1}{\varepsilon} & 1 \end{bmatrix}$, and so $\mathrm{cond}(T) = 5$, even though
$$\kappa_\infty(T) = 2(2 + \frac{1}{\varepsilon}) \approx \frac{2}{\varepsilon} + \mathcal{O}(1).$$
Thus, linear systems having T as a coefficient matrix will be solved to high relative accuracy, independent of both right-hand side and size of ϵ, despite the poor conditioning of T (as measured by κ_∞) as ϵ becomes small. However, note that
$$\mathrm{cond}(T^T) = 1 + \frac{2}{\varepsilon} \quad \text{and} \quad \kappa_\infty(T^T) = (1 + \varepsilon)\frac{2}{\varepsilon} \approx \frac{2}{\varepsilon} + \mathcal{O}(1).$$

So, linear systems having T^T as a coefficient matrix may have solutions that are sensitive to perturbations and indeed, $\text{cond}(T^T, \hat{\mathbf{x}}) \approx \text{cond}(T^T)$ for any right-hand side $\mathbf{b}$ with $b_3 \neq 0$ yielding solutions that are sensitive to perturbations for small ϵ.

51.3 Gauss Elimination and LU Decomposition

Gauss elimination is an elementary approach to solving systems of linear equations, yet it still constitutes the core of the most sophisticated of solution strategies. In the k^{th} step, a transformation matrix, M_k, (a "Gauss transformation") is designed so as to introduce zeros into A — typically into a portion of the k^{th} column — without harming zeros that have been introduced in earlier steps. Typically, successive applications of Gauss transformations are interleaved with row interchanges. Remarkably, this reduction process can be viewed as producing a decomposition of the coefficient matrix $A = NU$, where U is a triangular matrix and N is a row permutation of a lower triangular matrix.

Definitions:

For each index k, a **Gauss vector** is a vector in $\mathbb{C}^n$ with the leading k entries equal to zero: $\boldsymbol{\ell}_k = [\underbrace{0, \ldots, 0}_{k}, \ell_{k+1}, \ldots, \ell_n]^T$. The entries $\ell_{k+1}, \ldots, \ell_n$ are **Gauss multipliers**. The related matrix

$$M_k = I - \boldsymbol{\ell}_k \mathbf{e}_k^T$$

is called a **Gauss transformation**.

For the pair of indices (i, j), with $i \leq j$ the associated **permutation matrix**, $\Pi_{i,j}$ is an $n \times n$ identity matrix with the i^{th} row and j^{th} row interchanged. Note that $\Pi_{i,i}$ is the identity matrix.

A matrix $U \in \mathbb{C}^{m \times n}$ is in **row-echelon form** if (1) the first nonzero entry of each row has a strictly smaller column index than all nonzero entries with strictly larger row index and (2) zero rows occur at the bottom. The first nonzero entry in each row is called a **pivot**. The determining feature of row echelon form is that pivots occur to the left of all nonzero entries in lower rows.

A matrix $A \in \mathbb{C}^{m \times n}$ has an ***LU* decomposition** if there exists a unit lower triangular matrix $L \in \mathbb{C}^{m \times m}$ ($L_{i,j} = 0$ for $i < j$ and $L_{i,i} = 1$ for all i) and an upper triangular matrix $U \in \mathbb{C}^{m \times n}$ ($U_{i,j} = 0$ for $i > j$) such that $A = LU$.

Facts: [GV96]

1. Let $\mathbf{a} \in \mathbb{C}^n$ be a vector with a nonzero component in the r^{th} entry, $a_r \neq 0$. Define the Gauss vector, $\boldsymbol{\ell}_r = [\underbrace{0, \ldots, 0}_{r}, \frac{a_{r+1}}{a_r}, \ldots, \frac{a_n}{a_r}]^T$. The associated Gauss transformation $M_r = I - \boldsymbol{\ell}_r \mathbf{e}_r^T$ introduces zeros into the last $n - r$ entries of $\mathbf{a}$:
$$M_r \mathbf{a} = [a_1, \ldots, a_r, 0, \ldots, 0]^T.$$
2. If $A \in \mathbb{C}^{m \times n}$ with $\operatorname{rank} A = \rho \geq 1$ has ρ leading principal submatrices nonsingular, $A_{1:r,1:r}$, $r = 1, \ldots, \rho$, then there exist Gauss transformations $M_1, M_2, \ldots, M_\rho$ so that
$$M_\rho M_{\rho-1} \cdots M_1 A = U$$
with U upper triangular. Each Gauss transformation M_r introduces zeros into the r^{th} column.
3. Gauss transformations are unit lower triangular matrices. They are invertible, and for the Gauss transformation, $M_r = I - \boldsymbol{\ell}_r \mathbf{e}_r^T$,
$$M_r^{-1} = I + \boldsymbol{\ell}_r \mathbf{e}_r^T.$$

4. If Gauss vectors $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots, \boldsymbol{\ell}_{n-1}$ are given with
$$\boldsymbol{\ell}_1 = \begin{Bmatrix} 0 \\ \ell_{21} \\ \ell_{31} \\ \vdots \\ \ell_{n1} \end{Bmatrix}, \quad \boldsymbol{\ell}_2 = \begin{Bmatrix} 0 \\ 0 \\ \ell_{32} \\ \vdots \\ \ell_{n2} \end{Bmatrix}, \ \ldots, \quad \boldsymbol{\ell}_{n-1} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \ell_{n,n-1} \end{Bmatrix},$$
then the product of Gauss transformations $M_{n-1}M_{n-2}\cdots M_2M_1$ is invertible and has an explicit inverse
$$(M_{n-1}M_{n-2}\ldots M_2M_1)^{-1} = I + \sum_{k=1}^{n-1} \boldsymbol{\ell}_k \mathbf{e}_k^T = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ \ell_{21} & 1 & & & 0 \\ \ell_{31} & \ell_{32} & \ddots & & 0 \\ \vdots & & & 1 & 0 \\ \ell_{n1} & \ell_{n2} & \ldots & \ell_{n,n-1} & 1 \end{bmatrix}.$$
5. If $A \in \mathbb{C}^{m\times n}$ with $\operatorname{rank} A = \rho$ has ρ leading principal submatrices nonsingular, $A_{1:r,1:r}$, $r = 1, \ldots, \rho$, then A has an LU decomposition: $A = LU$, with L unit lower triangular and U upper triangular. The (i, j) entry of L: $L_{i,j}$, with $i > j$, is the Gauss multiplier that was used to introduce a zero into the corresponding (i, j) entry of A.
6. If $A \in \mathbb{C}^{m\times n}$ with $\operatorname{rank} A = m$ (full), then the LU decomposition is unique.
7. Let $\mathbf{a}$ be an arbitrary vector in $\mathbb{C}^n$. For any index r, there is an index $\mu \geq r$, a permutation matrix $\Pi_{r,\mu}$, and a Gauss transformation M_r so that
$$M_r\Pi_{r,\mu}\mathbf{a} = [a_1, \ldots, a_{r-1}, a_\mu, \underbrace{0, \ldots, 0}_{n-r}]^T.$$
The index μ is chosen so that $a_\mu \neq 0$ out of the set $\{a_r, a_{r+1}, \ldots, a_n\}$. If $a_r \neq 0$, then $\mu = r$ and $\Pi_{r,\mu} = I$ is a possible choice; if each element is zero, $a_r = a_{r+1} = \cdots = a_n = 0$, then $\mu = r$, $\Pi_{r,\mu} = I$, and $M_r = I$ is a possible choice.
8. For every matrix $A \in \mathbb{C}^{m\times n}$ with $\operatorname{rank} A = \rho$, there exists a sequence of ρ indices $\mu_1, \mu_2, \ldots, \mu_\rho$ with $i \leq \mu_i \leq m$ for $i = 1, \ldots, \rho$ and Gauss transformations $M_1, \ldots, M_\rho$ so that $M_\rho\Pi_{\rho,\mu_\rho}M_{\rho-1}\Pi_{\rho-1,\mu_{\rho-1}}\cdots M_1\Pi_{1,\mu_1}A = U$ with U upper triangular and in row echelon form. Each pair of transformations $M_r\Pi_{r,\mu_r}$ introduces zeros below the r^{th} pivot.
9. For $r < i < j$, $\Pi_{i,j}M_r = \widetilde{M}_r\Pi_{i,j}$, where $\widetilde{M}_r = I - \tilde{\boldsymbol{\ell}}_r\mathbf{e}_r^T$ and $\tilde{\boldsymbol{\ell}}_r = \Pi_{i,j}\boldsymbol{\ell}_r$ (i.e., the i and j entries of $\boldsymbol{\ell}_r$ are interchanged to form $\tilde{\boldsymbol{\ell}}_r$).
10. For every matrix $A \in \mathbb{C}^{m\times n}$ with $\operatorname{rank} A = \rho$, there is a row permutation of A that has an LU decomposition: $PA = LU$, with a permutation matrix P, unit lower triangular matrix L, and an upper triangular matrix U that is in row echelon form. P can be chosen as $P = \Pi_{\rho,\mu_\rho}\Pi_{\rho-1,\mu_{\rho-1}}\ldots\Pi_{1,\mu_1}$ from Fact 8, though in general there can be many other possibilities as well.
11. Reduction of A with Gauss transformations (or, equivalently, calculation of an LU factorization) must generally incorporate row interchanges. As a practical matter, these row interchanges commonly are chosen so as to bring the largest magnitude entry within the column being reduced up into the pivot location. This strategy is called "partial pivoting." In particular, if zeros are to be introduced into the k^{th} column below the r^{th} row (with $r \leq k$), then one seeks an index μ_r such that $r \leq \mu_r \leq m$ and $|A_{\mu_r,k}| = \max_{r\leq i\leq m} |A_{i,k}|$. When $\mu_1, \mu_2, \ldots, \mu_\rho$ in Fact 8 are chosen in this way, the reduction process is called "Gaussian Elimination with Partial Pivoting" (GEPP) or, within the context of factorization, the permuted LU factorization (PLU).

12. [GV96, p. 115]

Algorithm 3: GEPP/PLU decomposition of a rectangular matrix (outer product)

Input: $A \in \mathbb{R}^{m\times n}$
Output: $L \in \mathbb{R}^{m\times m}$ (unit lower triangular matrix)
$U \in \mathbb{R}^{m\times n}$ (upper triangular matrix - row echelon form)
$P \in \mathbb{R}^{m\times m}$ (permutation matrix) so that $PA = LU$
(P is represented with an index vector $\mathbf{p}$ such that $\mathbf{y} = P\mathbf{z} \Leftrightarrow y_j = z_{p_j}$)

$L \leftarrow I_m$; $U \leftarrow 0 \in \mathbb{R}^{m\times n}$; $\mathbf{p} = [1, 2, 3, \ldots, m]$; and $r \leftarrow 1$;
for $k = 1$ to n
  Find μ such that $r \le \mu \le m$ and $|A_{\mu,k}| = \max_{r\le i\le m} |A_{i,k}|$
  if $A_{\mu,k} \neq 0$, then
    Exchange $A_{\mu,k:n} \leftrightarrow A_{r,k:n}$, $L_{\mu,1:r-1} \leftrightarrow L_{r,1:r-1}$, and $p_\mu \leftrightarrow p_r$
    $L_{r+1:m,r} \leftarrow A_{r+1:m,k}/A_{r,k}$
    $U_{r,k:n} \leftarrow A_{r,k:n}$
    for $i = r + 1$ to m
      for $j = k + 1$ to n
        $A_{i,j} \leftarrow A_{i,j} - L_{i,r}U_{r,j}$
    $r \leftarrow r + 1$

Algorithm 4: GEPP/PLU decomposition of a rectangular matrix (gaxpy)

Input: $A \in \mathbb{R}^{m\times n}$
Output: $L \in \mathbb{R}^{m\times m}$ (unit lower triangular matrix),
$U \in \mathbb{R}^{m\times n}$ (upper triangular matrix - row echelon form), and
$P \in \mathbb{R}^{m\times m}$ (permutation matrix) so that $PA = LU$
(P is represented with an index vector $\boldsymbol{\pi}$ that records row interchanges
$\pi_r = \mu$ means row r and row $\mu \ge r$ were interchanged in step r)

$L \leftarrow I_m \in \mathbb{R}^{m\times m}$; $U \leftarrow 0 \in \mathbb{R}^{m\times n}$; and $r \leftarrow 1$;
for $j = 1$ to n
  $\mathbf{v} \leftarrow A_{1:m,j}$
  if $r > 1$, then
    for $i = 1$ to $r - 1$,
      Exchange $v_i \leftrightarrow v_{\pi_i}$
    Solve the triangular system, $L_{1:r-1,1:r-1} \cdot \mathbf{z} = \mathbf{v}_{1:r-1}$
    $U_{1:r-1,j} \leftarrow \mathbf{z}$
    Update $\mathbf{v}_{r:m} \leftarrow \mathbf{v}_{r:m} - L_{r:m,1:r-1} \cdot \mathbf{z}$
  Find μ such that $|v_\mu| = \max_{r\le i\le m} |v_i|$
  if $v_\mu \neq 0$, then
    $\pi_r \leftarrow \mu$
    Exchange $v_\mu \leftrightarrow v_r$
    for $i = 1$ to $r - 1$,
      Exchange $L_{\mu,i} \leftrightarrow L_{r,i}$
    $L_{r+1:m,r} \leftarrow \mathbf{v}_{r+1:m}/v_r$
    $U_{r,j} \leftarrow v_r$
    $r \leftarrow r + 1$

13. The condition for skipping reduction steps (that is, when $A_{\mu,k} = 0$ in Algorithm 3 or when $v_\mu = 0$ in Algorithm 4) indicates deficiency of column rank and the potential for an infinite number of solutions. These conditions are sensitive to rounding errors that may occur in the calculation of those columns and as such, GEPP/PLU is applied for the most part in full column rank settings ($\operatorname{rank} A = n$), guaranteeing that no zero pivots are encountered and that no reduction steps are skipped.
14. Both Algorithms 3 and 4 require approximately $\frac{2}{3}\rho^3 + \rho m(n-\rho) + \rho n(m-\rho)$ arithmetic operations (with $\operatorname{rank} A = \rho$). Algorithm 3 involves as a core calculation the updating of a submatrix having ever diminishing size. For large matrix dimension, the contents of this submatrix, $A_{r+1:m,k+1:n}$, may be widely scattered through computer memory and a performance penalty can occur in gathering the data for computation (which can be costly relative to the number of arithmetic operations that are performed with that data). Algorithm 4 is a reorganization that avoids excess data motion by delaying updates to columns until the step within which they have zeros introduced. This forces modifications to the matrix entries to be made just one column at a time and the necessary data motion can be more efficient.
15. The overhead associated with partial pivoting comes from lines beginning "Find μ such that ..." in Algorithms 3 and 4, involving a net $mn - \frac{n^2}{2}$ floating point comparison each of which are comparable in computational effort to a floating point subtraction. Typically this adds negligible overhead relative to the core complexity of $\mathcal{O}(\rho^3)$ arithmetic operations. Other strategies for avoiding the adverse effects of small pivots exist. Some are more aggressive than partial pivoting in producing the largest possible pivot (consequently have higher overhead), others are more restrained (and so, are cheaper).
    (a) *"Complete pivoting"* uses both row and column permutations to bring in the largest possible pivot: If zeros are to be introduced into the k^{th} column in row entries $r+1$ to m, then one seeks indices μ and ν such that $r \le \mu \le m$ and $k < \nu \le n$ such that $|A_{\mu,\nu}| = \max_{\substack{r \le i \le m \\ k < j \le n}} |A_{i,j}|$. Gauss elimination with complete pivoting produces a unit lower triangular matrix $L \in \mathbb{R}^{m\times m}$, an upper triangular matrix $U \in \mathbb{R}^{m\times n}$, and two permutation matrices, P and Q, so that $PAQ = LU$. This strategy can require $\mathcal{O}(\frac{mn^2}{2} - \frac{n^3}{6})$ floating point comparisons, potentially adding now nonnegligible overhead to the core arithmetic requirements. Overhead associated with data motion can become significant as well. The added stability that complete pivoting provides is rarely perceived to be worth the additional overhead.
    (b) *"Threshold pivoting"* identifies pivot candidates in each step that achieve a significant (predetermined) fraction of the magnitude of the pivot that would have been used in that step for partial pivoting: Consider all $\hat{\mu}$ such that $r \le \hat{\mu} \le m$ and $|A_{\hat{\mu},k}| \ge \tau \cdot \max_{r \le i \le m} |A_{i,k}|$, where $\tau \in (0,1)$ is a given threshold. This allows pivots to be chosen on the basis of other criteria such as influence on sparsity while still providing some protection from instability. τ can often be chosen quite small ($\tau = 0.1$ or $\tau = 0.025$ are typical values). See Section 53.5.
    (c) *"Rook pivoting"* searches the unreduced portion of the matrix for a pivot by tracing a path alternately along columns and rows ("rook"-like) following largest magnitude entries until an entry having largest magnitude in both its row and column is discovered. This approach is more aggressive than partial pivoting, yet typically has overhead that is a small multiple of that for partial pivoting. For some matrices overhead can be comparable to complete pivoting.
16. An efficient and reliable implementation of the GEPP/PLU factorization is offered in the LAPACK software library as `xGETRF`; solving associated linear systems may be done with `xGESV` (see Section 93.2).

17. If $\hat{P} \in \mathbb{R}^{m\times m}$, $\hat{L} \in \mathbb{R}^{m\times m}$, and $\hat{U} \in \mathbb{R}^{m\times n}$ are the computed permutation matrix and LU factors from either Algorithm 3 or 4 on $A \in \mathbb{R}^{m\times n}$, then
$$\hat{L}\hat{U} = \hat{P}(A + \delta A) \quad \text{with} \quad |\delta A| \leq \frac{2\,n\,\epsilon}{1 - n\,\epsilon}\,|\hat{L}|\,|\hat{U}|$$
and for the particular case that $m = n$ and A is nonsingular, if an approximate solution, $\hat{\mathbf{x}}$, to $A\mathbf{x} = \mathbf{b}$ is computed by solving the two triangular linear systems, $\hat{L}\mathbf{y} = \hat{P}\mathbf{b}$ and $\hat{U}\hat{\mathbf{x}} = \mathbf{y}$, then $\hat{\mathbf{x}}$ is the exact solution to a perturbed linear system:
$$(A + \delta A)\hat{\mathbf{x}} = \mathbf{b} \quad \text{with} \quad |\delta A| \leq \frac{2\,n\,\epsilon}{1 - n\,\epsilon}\,\hat{P}^T\,|\hat{L}|\,|\hat{U}|.$$
Furthermore, $|L_{i,j}| \leq 1$ and $|U_{i,j}| \leq 2^{i-1} \max_{k\leq i} |A_{k,j}|$, so
$$\|\delta A\|_\infty \leq \frac{2^n\,n^2\,\epsilon}{1 - n\,\epsilon}\,\|A\|_\infty.$$

Examples:

1. Using Algorithm 3, find a permuted LU factorization of
$$A = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 2 & 4 & 6 \\ -1 & -1 & -1 & 1 \\ 1 & 1 & 3 & 1 \end{bmatrix}.$$

Setup:	$\mathbf{p} = [1\ 2\ 3\ 4]$, $r \leftarrow 1$
$k = 1$ *step*:	$\mu \leftarrow 2$, $\mathbf{p} = [2\ 1\ 3\ 4]$ Permuted A: $\begin{bmatrix} 2 & 2 & 4 & 6 \\ 1 & 1 & 2 & 3 \\ -1 & -1 & -2 & 1 \\ 1 & 1 & 3 & 1 \end{bmatrix}$
LU snapshot:	$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 2 & 4 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$. Updated $A_{2:4,2:4}$: $\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 4 \\ 0 & 1 & -2 \end{bmatrix}$ $r \leftarrow 2$
$k = 2$ *step*:	$\mu \leftarrow 2$, $\lvert A_{2,2}\rvert = \max_{2\leq i\leq 4} \lvert A_{i,2}\rvert = 0$ (skip reduction step)
$k = 3$ *step*:	$\mu \leftarrow 3$, $\mathbf{p} = [2\ 3\ 1\ 4]$, $\lvert A_{3,3}\rvert = \max_{2\leq i\leq 4} \lvert A_{i,3}\rvert = 1$ Permuted $A_{2:4,3:4}$: $\begin{bmatrix} 1 & 4 \\ 0 & -1 \\ 1 & -2 \end{bmatrix}$
LU snapshot:	$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ \frac{1}{2} & 1 & 0 & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 2 & 4 & 6 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$. Updated $A_{3:4,4}$: $\begin{bmatrix} -1 \\ -6 \end{bmatrix}$ $r \leftarrow 3$

$k = 4$ *step*: $\mu \leftarrow 4$, $\mathbf{p} = [2\ 3\ 4\ 1]$, $|A_{4,4}| = \max_{3 \le i \le 4} |A_{i,4}| = 6$

Permuted $A_{2:4,3:4}$: $\begin{bmatrix} -6 \\ -1 \end{bmatrix}$

LU snapshot: $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 1 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{6} & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 2 & 4 & 6 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & -6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

$r \leftarrow 4$

The permutation matrix associated with $\mathbf{p} = [2\ 3\ 4\ 1]$ is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } PA = \begin{bmatrix} 2 & 2 & 4 & 6 \\ -1 & -1 & -1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & 1 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 1 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{6} & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 4 & 6 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & -6 \\ 0 & 0 & 0 & 0 \end{bmatrix} = L \cdot U.$$

2. Using Algorithm 4, solve the system of linear equations

$$\begin{bmatrix} 1 & 3 & 1 \\ 2 & 2 & -1 \\ 2 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 3 \end{bmatrix}.$$

Phase 1: Find permuted LU decomposition.

$r \leftarrow 1$

$j = 1$ *step*: $\mathbf{v} \leftarrow \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$. $\pi_1 \leftarrow \mu = 2$. Permuted $\mathbf{v}$: $\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$

LU snapshot: $\boldsymbol{\pi} = [2]$ $L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

$r \leftarrow 2$

$j = 2$ *step*: $\mathbf{v} \leftarrow \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix}$. Permuted $\mathbf{v}$: $\begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$.

Solve $1 \cdot \mathbf{z} = 2$. $U_{1,2} \leftarrow \mathbf{z} = [2]$.

$$\begin{bmatrix} \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} \leftarrow \begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} [2]$$

$\pi_2 \leftarrow \mu = 3$. $L_{3,2} \leftarrow -\frac{2}{3}$ and $U_{2,2} \leftarrow -3$

LU snapshot: $\boldsymbol{\pi} = [2,\ 3]$ $L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 2 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

$r \leftarrow 3$

$j = 3$ *step*: $\mathbf{v} \leftarrow \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$ Permuted $\mathbf{v}$: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$. Solve $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \mathbf{z} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$,

$\begin{bmatrix} U_{1,3} \\ U_{2,3} \end{bmatrix} \leftarrow \mathbf{z} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. $v_3 \leftarrow 2\frac{1}{6} = 1 - [\frac{1}{2},\ -\frac{2}{3}] \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

$\pi_3 \rightarrow 3$ and $U_{3,3} \leftarrow 2\frac{1}{6}$

LU snapshot: $\boldsymbol{\pi} = [2,\ 3,\ 3]$ $L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 2 & -1 \\ 0 & -3 & 1 \\ 0 & 0 & 2\frac{1}{6} \end{bmatrix}$.

The permutation matrix associated with $\boldsymbol{\pi}$ is

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \text{ and } PA = \begin{bmatrix} 2 & 2 & -1 \\ 2 & -1 & 0 \\ 1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & -1 \\ 0 & -3 & 1 \\ 0 & 0 & 2\frac{1}{6} \end{bmatrix} = L \cdot U.$$

Phase 2: Solve the lower triangular system $L\mathbf{y} = P\mathbf{b}$.

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 1 \end{bmatrix} \quad \Rightarrow \quad y_1 = -3,\ y_2 = 6,\ y_3 = 6\frac{1}{2}.$$

Phase 3: Solve the upper triangular system $U\mathbf{x} = \mathbf{y}$.

$$\begin{bmatrix} 2 & 2 & -1 \\ 0 & -3 & 1 \\ 0 & 0 & 2\frac{1}{6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 6 \\ 6\frac{1}{2} \end{bmatrix} \quad \Rightarrow \quad x_1 = 1,\ x_2 = -1,\ x_3 = 3.$$

51.4 Symmetric Factorizations

Real symmetric matrices ($A = A^T$) and their complex analogs, Hermitian matrices (Chapter 9), are specified by roughly half the number of parameters than general $n \times n$ matrices, so one could anticipate benefits that take advantage of this structure.

Definitions:

An $n \times n$ matrix, A, is **Hermitian** if $A = A^* = \bar{A}^T$.

$A \in \mathbb{C}^{n \times n}$ is **positive-definite** if $\mathbf{x}^* A\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{C}^n$ with $\mathbf{x} \neq 0$.

The **Cholesky decomposition** (or **Cholesky factorization**) of a positive-definite matrix A is $A = G\,G^*$ with $G \in \mathbb{C}^{n \times n}$ lower triangular and having positive diagonal entries.

Facts: [Hig02], [GV96]

1. A positive-definite matrix is Hermitian. Note that the similar but weaker assertion for a matrix $A \in \mathbb{R}^{n \times n}$ that "$\mathbf{x}^T A\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \neq 0$" does not imply that $A = A^T$.

2. If $A \in \mathbb{C}^{n\times n}$ is positive-definite, then A has an LU decomposition, $A = LU$, and the diagonal of U, $\{u_{11}, u_{22}, \ldots, u_{nn}\}$, has strictly positive entries.
3. If $A \in \mathbb{C}^{n\times n}$ is positive-definite, then the LU decomposition of A satisfies $A = LU$ with $U = D\,L^*$ and $D = \mathrm{diag}(U)$. Thus, A can be written as $A = LDL^*$ with L unit lower triangular and D diagonal with positive diagonal entries. Furthermore, A has a Cholesky decomposition $A = G\,G^*$ with $G \in \mathbb{C}^{n\times n}$ lower triangular. Indeed, if

$$\widehat{D} = diag(\{\sqrt{u_{11}}, \sqrt{u_{22}}, \ldots, \sqrt{u_{nn}}\})$$

then $\widehat{D}\widehat{D} = D$ and $G = L\,\widehat{D}$.
4. [GV96, p. 144] The Cholesky decomposition of a positive-definite matrix A can be computed directly:

Algorithm 5: Cholesky decomposition of a positive-definite matrix

Input: $A \in \mathbb{C}^{n\times n}$ positive definite
Output: $G \in \mathbb{C}^{n\times n}$ (lower triangular matrix so that $A = G\,G^*$)

```
G ← 0 ∈ C^{n×n};
for j = 1 to n
    v ← A_{j:n,j}
    for k = 1 to j − 1,
        v ← v − conj(G_{j,k}) G_{j:n,k}
    G_{j:n,j} ← (1/√v_1) v    (v_1 is v(1))
```

5. Algorithm 5 requires approximately $n^3/3$ floating point arithmetic operations and n floating point square roots to complete (roughly half of what is required for an LU decomposition).
6. If $A \in \mathbb{R}^{n\times n}$ is symmetric and positive-definite and Algorithm 5 runs to completion producing a computed Cholesky factor $\hat{G} \in \mathbb{R}^{n\times n}$, then

$$\hat{G}\hat{G}^T = A + \delta A \quad \text{with} \quad |\delta A| \leq \frac{(n+1)\,\epsilon}{1-(n+1)\,\epsilon}\,|\hat{G}|\,|\hat{G}^T|.$$

Furthermore, if an approximate solution, $\hat{\mathbf{x}}$, to $A\mathbf{x} = \mathbf{b}$ is computed by solving the two triangular linear systems $\hat{G}\mathbf{y} = \mathbf{b}$ and $\hat{G}^T\hat{\mathbf{x}} = \mathbf{y}$, and a scaling matrix is defined as $\Delta = \mathrm{diag}(\sqrt{a_{ii}})$, then the scaled error $\Delta(\mathbf{x} - \hat{\mathbf{x}})$ satisfies

$$\frac{\|\Delta(\mathbf{x}-\hat{\mathbf{x}})\|_2}{\|\Delta\mathbf{x}\|_2} \leq \frac{\kappa_2(H)\,\epsilon}{1-\kappa_2(H)\,\epsilon},$$

where $A = \Delta\,H\,\Delta$. If $\kappa_2(H) \ll \kappa_2(A)$, then it is quite likely that the entries of $\Delta\hat{\mathbf{x}}$ will have mostly the same magnitude and so the error bound suggests that all entries of the solution will be computed to high relative accuracy.
7. If $A \in \mathbb{C}^{n\times n}$ is Hermitian and has all leading principal submatrices nonsingular, then A has an LU decomposition that can be written as $A = LU = L\,D\,L^*$ with L unit lower triangular and D diagonal with real diagonal entries. Furthermore, the number of positive and negative entries of D is equal to the number of positive and negative eigenvalues of A, respectively (the *Sylvester law of inertia*).
8. It may not be prudent to compute the LU (or LDL^T) decomposition of a Hermitian indefinite matrix A without pivoting, yet the usual pivoting strategies will likely eliminate advantages that symmetry might offer. Alternatives use *symmetric pivoting*

to produce a factorization of the permuted matrix: $PAP^T = LDL^T$ where L is unit lower triangular as before, but D is block diagonal with 1×1 or 2×2 diagonal blocks. The block structure is due to the possibility of using 2×2 principal submatrices as "pivots" during the reduction process. (see [GV96, Hig02, Ste98] for details).

(a) *"Bunch-Parlett pivoting"* uses either the largest magnitude diagonal entry in the unreduced submatrix provided it is not much smaller in magnitude than the entry that would have been chosen with complete pivoting, or failing that, permutes into pivot position a 2×2 principal submatrix that captures the complete pivoting choice. The number of comparisons necessary for this strategy is the same as what is necessary for complete pivoting, so the added overhead is nonnegligible and, as is the case with complete pivoting, this approach is generally viewed as unnecessarily conservative.

(b) *"Bunch-Kaufman pivoting"* uses a similar strategy but akin to partial pivoting, using either a 2×2 principal submatrix that captures the partial pivoting choice or one of the two diagonal entries of this submatrix, the choice being governed by their relative magnitudes. The number of comparisons necessary for this strategy is similar to what is necessary for partial pivoting, so the added overhead is generally insignificant. Although the magnitude of the Gauss multipliers (entries of L) cannot be bounded uniformly with respect to A, the solution of linear systems with this approach is nonetheless backward stable (see [Hig02] for a discussion).

(c) *"Symmetric Rook pivoting"* is a refinement of the Bunch-Kaufman approach using a symmetrized form of rook pivoting. It is more aggressive than Bunch-Kaufman pivoting; generally has similar overhead; may on occasion require overhead comparable to complete pivoting; but, significantly for certain applications, will produce an L factor that may be bounded uniformly with respect to A.

9. An efficient and reliable implementation of Cholesky factorization (Algorithm 5) is in the LAPACK software library as `xPOTRF`; for symmetric indefinite matrices, the LDL^T factorization with Bunch-Kaufman pivoting is available as `xSYTRF`. Solving associated linear systems may be done with `xPOSV` and `xSYSV`, respectively (see Section 93.2).

Examples:

1. Calculate the Cholesky decomposition of the 3×3 Hilbert matrix,

$$A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}.$$

Setup:	$G \leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.
$j = 1$ *step*:	$\mathbf{v} \leftarrow [1, \frac{1}{2}, \frac{1}{3}]^T$
G *snapshot*:	$G = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 0 \end{bmatrix}$

$j = 2$ *step*:	$\mathbf{v} \leftarrow \begin{bmatrix} \frac{1}{3} \\ \frac{1}{4} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{1}{12} \\ \frac{1}{12} \end{bmatrix}$
G *snapshot*:	$G = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2\sqrt{3}} & 0 \\ \frac{1}{3} & \frac{1}{2\sqrt{3}} & 0 \end{bmatrix}$
$j = 3$ *step*:	$\mathbf{v} \leftarrow \frac{1}{5} - \left(\frac{1}{3}\right)^2 - \left(\frac{1}{2\sqrt{3}}\right)^2 = \frac{1}{180} = \left(\frac{1}{6\sqrt{5}}\right)^2$
G *snapshot*:	$G = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2\sqrt{3}} & 0 \\ \frac{1}{3} & \frac{1}{2\sqrt{3}} & \frac{1}{6\sqrt{5}} \end{bmatrix}$

51.5 Orthogonalization and the QR Factorization

The process of transforming an arbitrary linear system into a triangular system may also be approached by systematically introducing zeros into the coefficient matrix with unitary transformations: Given a system $A\mathbf{x} = \mathbf{b}$, (1) find unitary matrices V_1, V_2, $\cdots$ V_ℓ such that $V_\ell \ldots V_2 V_1 A = T$ is triangular; (2) calculate $\mathbf{y} = V_\ell \cdots V_2 V_1 \mathbf{b}$; and (3) solve the triangular system $T\mathbf{x} = \mathbf{y}$. Or equivalently,

1. Factor $A = QR$, where Q is unitary ($Q^{-1} = Q^*$) and R is upper triangular.
2. Calculate $\mathbf{y} = Q^*\mathbf{b}$.
3. Solve the triangular system, $R\mathbf{x} = \mathbf{y}$.

The classical approach of Gram-Schmidt orthogonalization leads spontaneously to the QR factorization of a matrix. This topic is discussed in Section 5.5. Two other types of rudimentary unitary transformations to affect the QR factorization will be described here: Householder transformations and Givens transformations.

Definitions:

A ***QR* factorization** of a matrix $A \in \mathbb{C}^{m\times n}$ is a factorization of $A = QR$ where $Q \in \mathbb{C}^{m\times m}$ is unitary ($Q^{-1} = Q^*$) and $R \in \mathbb{C}^{m\times n}$ is upper triangular ($R = [r_{ij}]$ with $r_{ij} = 0$ when $i > j$). See also Section 5.5.

Let $\mathbf{v} \in \mathbb{C}^n$ be a unit vector: $\|\mathbf{v}\|_2 = 1$. The matrix $H = I - 2\mathbf{v}\mathbf{v}^*$ is called a **Householder transformation** (or **Householder reflector**). In this context, $\mathbf{v}$ is a **Householder vector**.

For θ, $\vartheta \in [0, 2\pi)$, let G_{ij} be an $n \times n$ identity matrix modified so that the (i, i) and (j, j) entries are each replaced by $c = \cos(\theta)$, the (i, j) entry is replaced by $s = e^{\imath\vartheta}\sin(\theta)$, and the (j, i) entry is replaced by $-\bar{s} = -e^{-\imath\vartheta}\sin(\theta)$:

$$G_{ij} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -\bar{s} & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}.$$

G_{ij} is called a **Givens transformation** (or **Givens rotation**).

Facts: [GV96, Hig02, Ste98]

1. The QR factorization plays in important role in the solution of least squares problems: Given $A \in \mathbb{C}^{m\times n}$ and $\mathbf{b} \in \mathbb{C}^m$, find $\hat{\mathbf{x}}$ that solves $\min_{\mathbf{x}\in\mathbb{C}^n} \|A\mathbf{x}-\mathbf{b}\|_2$. (See Chapter 52.)
2. A closely related problem is the solution of *underdetermined linear systems*: Given $A \in \mathbb{R}^{m\times n}$ with $m < n$ and $\mathbf{b} \in \mathbb{R}^m$, find $\hat{\mathbf{x}}$ that solves $A\mathbf{x} = \mathbf{b}$. The solution to this problem (if it exists) will not be unique but may be characterized as follows: Calculate a QR decomposition of A^T as $A^T = QR = Q\begin{bmatrix}\widehat{R}\\ \mathbf{0}\end{bmatrix}$. Then the solution set to $A\mathbf{x} = \mathbf{b}$ may be parameterized as $\hat{\mathbf{x}}(\mathbf{z}) = Q\begin{bmatrix}\widehat{R}^{-T}\mathbf{b}\\ \mathbf{z}\end{bmatrix}$ for arbitrary $\mathbf{z} \in \mathbb{R}^{n-m}$. The solution to $A\mathbf{x} = \mathbf{b}$ that has minimum norm is given by $\hat{\mathbf{x}}(\mathbf{0})$ (i.e., with $\mathbf{z} = \mathbf{0}$).
3. If $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $A = QR = Q\begin{bmatrix}\widehat{R}\\ \mathbf{0}\end{bmatrix}$ is the QR factorization of A, then $\widehat{R}^T$ is the Cholesky factor of the positive-semidefinite matrix, A^TA: $A^TA = \widehat{R}^T\widehat{R}$ (see Section 51.4).
4. The rank of A is the same as the rank of R. If A is full rank, the diagonal entries of R provide a bound on how near A is to rank deficiency:
$$\min\{\|E\| \mid \operatorname{rank}(A+E) < \operatorname{rank} A\} \leq \min_i |r_{ii}|.$$
This bound may be very pessimistic. There are refinements of QR factorization that incorporate pivoting among other strategies to obtain a *rank-revealing factorization.* See Sections 52.9 and 59.3.
5. Householder transformations are unitary matrices. The action of H on a vector $\mathbf{a}$ has a geometric interpretation as a reflection of $\mathbf{a}$ across the hyperplane with a normal $\mathbf{v}$.
6. Let $\mathbf{a} \in \mathbb{C}^n$ be a nonzero vector. Define $\mathbf{w} = \operatorname{sign}(a_1)\|\mathbf{a}\|\mathbf{e}_1+\mathbf{a}$ with $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{C}^n$ and $\mathbf{v} = \mathbf{w}/\|\mathbf{w}\|_2$ Then the Householder transformation $H = I - 2\mathbf{v}\mathbf{v}^*$ satisfies
$$H\mathbf{a} = \alpha\mathbf{e}_1 \quad \text{with} \quad \alpha = -\operatorname{sign}(a_1)\|\mathbf{a}\|,$$
i.e., H is a unitary matrix that introduces $n-1$ zeros into the vector $\mathbf{a}$.
7. [GV96, pp. 210–213]

Algorithm 6: Householder QR factorization of a rectangular matrix

Input: $A \in \mathbb{C}^{m\times n}$ with $m \geq n$
Output: the QR factorization $A = QR$; R overwrites A.

for $k = 1$ to n
  $\mathbf{x} \leftarrow A_{k:m,k}$
  $\mathbf{v}_k \leftarrow \operatorname{sign}(x_1)\|\mathbf{x}\|\mathbf{e}_1 + \mathbf{x}$, where $\mathbf{e}_1 \in \mathbb{C}^{m-k+1}$
  $\mathbf{v}_k \leftarrow \mathbf{v}_k/\|\mathbf{v}_k\|$
  $A_{k:m,k:n} \leftarrow (I_{m-k+1} - 2\mathbf{v}_k\mathbf{v}_k^*)A_{k:m,k:n}$
$Q \leftarrow I_m$
for $k = n$ to 1 by -1
  $Q_{k:m,k:m} \leftarrow Q_{k:m,k:m}(I_{m-k+1} - 2\mathbf{v}_k\mathbf{v}_k^*)$

8. In Algorithm 6, the Householder reflectors are accumulated to form Q in order of size, reversing the order in which they were generated. This is more efficient than accumulating the reflectors in the order that they are generated. In many applications,

it suffices to have Q represented *implicitly* as the product of Householder reflectors and it may not be necessary to compute Q explicitly. Data necessary to recreate the action of Q may be stored in subdiagonal entries of A as zeros are created there.

9. If $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, then the cost of Algorithm 6 without explicitly computing Q is $2n^2(m - n/3)$ flops.
10. If $\widehat{R} \in \mathbb{C}^{m\times n}$ is the *computed* upper triangular matrix provided by Algorithm 6, then there is a unitary matrix, $\widetilde{Q} \in \mathbb{C}^{m\times m}$ such that $\widetilde{Q}\widehat{R}$ is the exact QR factorization of a perturbation of A: $A + \delta A = \widetilde{Q}\widehat{R}$ where δA has columns $\delta A = [\delta\mathbf{a}_1\, \delta\mathbf{a}_2\, \ldots,\, \delta\mathbf{a}_n] \in \mathbb{C}^{m\times n}$ and $\|\delta\mathbf{a}_k\|_2 \leq \dfrac{c\,m\,n\,\epsilon}{1 - c\,m\,n\,\epsilon}\|\mathbf{a}_k\|_2$ for a $c > 0$ having modest magnitude. If $\widehat{Q}$ is the *computed* Q provided by Algorithm 6, then $\|\widehat{Q} - \widetilde{Q}\|_F \leq \dfrac{c\,m\,n\sqrt{n}\,\epsilon}{1 - c\,m\,n\,\epsilon}$.
11. An efficient and reliable implementation of the QR factorization using Householder reflectors (as in Algorithm 6) appears in the LAPACK software library as `xGEQRF` (see Section 93.2).
12. A Givens rotation is a unitary matrix. The action of G_{ij} on a vector $\mathbf{a}$ has a geometric interpretation as a (complex) rotation of $\mathbf{a}$ within the (i, j) coordinate plane.
13. For any scalars $x, y \in \mathbb{C}$, there exists a Givens rotation $G \in \mathbb{C}^{2\times 2}$ such that

$$G\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix},$$

where c, s, and r can be computed via

(a) If $y = 0$ (includes the case $x = y = 0$), then $c = 1,\ s = 0,\ r = x$.

(b) If $x = 0$ (y must be nonzero), then $c = 0,\ s = \text{sign}(\bar{y}),\ r = |y|$.

(c) If both x and y are nonzero, then $c = |x|/\sqrt{|x|^2 + |y|^2}$, $s = \text{sign}(x)\bar{y}/\sqrt{|x|^2 + |y|^2}$, $r = \text{sign}(x)\sqrt{|x|^2 + |y|^2}$.

14. [GV96, pp. 226–227]

Algorithm 7: Givens QR factorization of a rectangular matrix

Input: $A \in \mathbb{C}^{m\times n}$ with $m \geq n$

Output: the QR factorization $A = QR$; the upper triangular part of R is stored in the upper triangular part of A

$Q \leftarrow I_m$

for $k = 1$ to n

  for $i = k + 1$ to m

    Compute $G = \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}$ such that $G\begin{bmatrix} A_{kk} \\ A_{ik} \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$ (via Fact 13).

    $\begin{bmatrix} A_{k,k:n} \\ A_{i,k:n} \end{bmatrix} \leftarrow G\begin{bmatrix} A_{k,k:n} \\ A_{i,k:n} \end{bmatrix}$

    $[Q_{1:m,k}, Q_{1:m,i}] \leftarrow [Q_{1:m,k}, Q_{1:m,i}]G^*$

15. As noted above, in many applications it suffices to have Q represented *implicitly* removing the need to accumulate Givens rotations to form Q explicitly. Data needed to recreate Q may be stored in subdiagonal locations in A, as zeros are introduced.
16. If $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, then the cost of Algorithm 7 without explicitly computing Q is $3n^2(m - n/3)$ flops.

17. If $\widehat{R} \in \mathbb{C}^{m\times n}$ is the *computed* upper triangular matrix provided by Algorithm 7, then there is a unitary matrix, $\widetilde{Q} \in \mathbb{C}^{m\times m}$ such that $\widetilde{Q}\widehat{R}$ is the exact QR factorization of a perturbation of A: $A + \delta A = \widetilde{Q}\widehat{R}$ where δA has columns $\delta A = [\delta\mathbf{a}_1\, \delta\mathbf{a}_2\, \ldots,\, \delta\mathbf{a}_n] \in \mathbb{C}^{m\times n}$ and $\|\delta\mathbf{a}_k\|_2 \leq \dfrac{c\,(m+n-2)\,\epsilon}{1-c\,(m+n-2)\,\epsilon}\|\mathbf{a}_k\|_2$ for a $c > 0$ having modest magnitude.

Examples:

1. We shall use Givens rotations to transform $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$ to upper triangular form, as in Algorithm 7. First, to annihilate the element in position (2,1), we use Fact 5 with $(x, y) = (1, 1)$ and obtain $c = s = 1/\sqrt{2}$; hence:

$$A^{(1)} = G_1 A = \begin{bmatrix} 0.7071 & 0.7071 & 0 \\ -0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1.4142 & 2.1213 \\ 0 & 0.7071 \\ 1 & 3 \end{bmatrix}.$$

Next, to annihilate the element in position (3,1), we use $(x, y) = (1.4142, 1)$ in Fact 5 and get

$$A^{(2)} = G_2 A^{(1)} = \begin{bmatrix} 0.8165 & 0 & 0.5774 \\ 0 & 1 & 0 \\ -0.5774 & 0 & 0.8165 \end{bmatrix} A^{(1)} = \begin{bmatrix} 1.7321 & 3.4641 \\ 0 & 0.7071 \\ 0 & 1.2247 \end{bmatrix}.$$

Finally, we annihilate the element in position (3,2) using $(x, y) = (.7071, 1.2247)$:

$$A^{(3)} = G_3 A^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5000 & 0.8660 \\ 0 & -0.8660 & 0.5000 \end{bmatrix} A^{(2)} = \begin{bmatrix} 1.7321 & 3.4641 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix}.$$

As a result, $R = A^{(3)}$ and $\widehat{R}$ consists of the first two rows of $A^{(3)}$. The matrix Q can be computed as the product $G_1^T G_2^T G_3^T$.

2. We shall use Householder reflections to transform A from Example 1 to upper triangular form as in Algorithm 6. First, let $\mathbf{a} = A_{:,1} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$, $\gamma_1 = -\sqrt{3}$, $\widetilde{\mathbf{a}} = \begin{bmatrix} -\sqrt{3} & 0 & 0 \end{bmatrix}^T$, and $\mathbf{u}_1 = \begin{bmatrix} 0.8881 & 0.3251 & 0.3251 \end{bmatrix}^T$; then

$$A^{(1)} = \left(I - 2\mathbf{u}_1\mathbf{u}_1^T\right) A = A - \mathbf{u}_1 \overbrace{\begin{bmatrix} 3.0764 & 5.0267 \end{bmatrix}}^{2\mathbf{u}_1^T A} = \begin{bmatrix} -1.7321 & -3.4641 \\ 0 & 0.3660 \\ 0 & 1.3660 \end{bmatrix}.$$

Next, $\gamma_2 = -\|A^{(1)}_{2:3,2}\|_2$, $\mathbf{u}_2 = \begin{bmatrix} 0 & 0.7934 & 0.6088 \end{bmatrix}^T$, and

$$A^{(2)} = \left(I - 2\mathbf{u}_2\mathbf{u}_2^T\right) A^{(1)} = A^{(1)} - \mathbf{u}_2 \overbrace{\begin{bmatrix} 0 & 2.2439 \end{bmatrix}}^{2\mathbf{u}_2^T A^{(1)}} = \begin{bmatrix} -1.7321 & -3.4641 \\ 0 & -1.4142 \\ 0 & 0 \end{bmatrix}.$$

Note that $R = A^{(2)}$ has changed signs as compared with Example 1. The matrix Q can be computed as $\left(I - 2\mathbf{u}_1\mathbf{u}_1^T\right)\left(I - 2\mathbf{u}_2\mathbf{u}_2^T\right)$. Therefore, we have full information about the transformation if we store the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$.

References

[Dem97] J. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[GV96] G.H. Golub and C.F. Van Loan. *Matrix Computations*, 3rd ed. Johns Hopkins University Press, Baltimore, MD, 1996.

[Hig02] N.J. Higham. *Accuracy and Stability of Numerical Algorithms*, 2nd ed. SIAM, Philadelphia, 2002.

[Ste98] G.W. Stewart. *Matrix Algorithms, Vol I: Basic Decompositions*. SIAM, Philadelphia, 1998.

[SS90] G.W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, San Diego, CA, 1990.

[TB97] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.