Section 4.2.1: Given Rotations

Jim Lambers

February 8, 2021

Let $A \in \mathbb{R}^{m\times n}$ matrix with full column rank. The **QR Factorization** of A is a decomposition $A = QR$, where Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix. There are three ways to compute this decomposition:

1. Using **Givens rotations**, also known as **Jacobi rotations**, used by Givens and originally invented by Jacobi for use with in solving the symmetric eigenvalue problem in 1846.
2. Using **Householder reflections**, also known as **Householder transformations**, developed by Householder.
3. A third, less frequently used approach known as **Gram-Schmidt Orthogonalization**.

Givens Rotations

- We have seen how elementary row operations can be used to reduce a matrix to upper triangular form, resulting in the LU Decomposition $PA = LU$.
- To compute the factorization $A = QR$, we can use a similar approach, in which (non-elementary) row operations are applied to A to reduce A to upper triangular form. Specifically,

$$Q_k^T \cdots Q_2^T Q_1^T A = R \quad \Longrightarrow \quad A = Q_1 Q_2 \cdots Q_k R = QR.$$

  Here, we are using the fact that $Q_i^T = Q_i^{-1}$ (from $Q_i^T Q_i = Q_i Q_i^T = I$) and that if Q_1 and Q_2 are orthogonal, then $Q_1 Q_2$ is also orthogonal. This is similar to what happens in Gaussian elimination: being unit lower triangular is preserved by multiplication and inversion.
- Orthogonal matrices are our friends because: 1) trivial to invert, 2) perfectly conditioned! If Q is orthogonal, then $\kappa_2(Q) = \|Q\|_2 \|Q^{-1}\|_2 = \|Q\|_2 \|Q^T\|_2 = 1$ because $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2$.
- If each such row operation, designed to zero a_{ij} where $i > j$, can be implemented through pre-multiplication (multiplying on left) by an orthogonal matrix, then the accumulation of these row operations is implemented through pre-multiplication by the product of these orthogonal matrices, which is itself the orthogonal matrix Q^T.

We illustrate the process in the case where A is a 2×2 matrix, for which we need only zero a_{21}. The QR Factorization computes $Q^T A = R$, or

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix},$$

where $c^2 + s^2 = 1$ to ensure Q is orthogonal. We have $c = \cos\theta$, $s = \sin\theta$ for some θ.

From the relationship $-sa_{11} + ca_{21} = 0$ we obtain

$$c^2 a_{21}^2 = s^2 a_{11}^2 = (1 - c^2) a_{11}^2$$

which yields

$$c = \pm \frac{a_{11}}{\sqrt{a_{21}^2 + a_{11}^2}}.$$

It is conventional to choose the + sign. Then, we obtain

$$s^2 = 1 - c^2 = 1 - \frac{a_{11}^2}{a_{21}^2 + a_{11}^2} = \frac{a_{21}^2}{a_{21}^2 + a_{11}^2},$$

or

$$s = \pm \frac{a_{21}}{\sqrt{a_{21}^2 + a_{11}^2}}.$$

Again, we choose the + sign. As a result, we have

$$r_{11} = a_{11} \frac{a_{11}}{\sqrt{a_{21}^2 + a_{11}^2}} + a_{21} \frac{a_{21}}{\sqrt{a_{21}^2 + a_{11}^2}} = \sqrt{a_{21}^2 + a_{11}^2}.$$

- The matrix

  $$Q = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

  is called a **Givens rotation**.

- It is called a rotation because it is orthogonal, and therefore length-preserving, and also because there is an angle θ such that $\sin\theta = s$ and $\cos\theta = c$.

- The effect of pre-multiplying a vector by Q^T is to rotate the vector *clockwise* through the angle θ.

- In particular, if $a = r\cos\theta$ and $b = r\sin\theta$ for some angle θ, then

  $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad r = \sqrt{a^2 + b^2}.$$

  That is, the point (a, b) is rotated to the positive x-axis, effectively "undoing" the counterclockwise rotation by θ inherent in polar coordinates.

- Now, to see how Givens rotations can be used to zero entries of an $m \times n$ matrix A, suppose that we have the vector

  $$\begin{bmatrix} \times \\ \vdots \\ \times \\ a \\ \times \\ \vdots \\ \times \\ b \\ \times \\ \vdots \\ \times \end{bmatrix}$$

that is a column of A. Then

$$\begin{bmatrix} 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & c & & & & s & & & \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & -s & & & & c & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} \times \\ \vdots \\ \times \\ a \\ \times \\ \vdots \\ \times \\ b \\ \times \\ \vdots \\ \times \end{bmatrix} = \begin{bmatrix} \times \\ \vdots \\ \times \\ r \\ \times \\ \vdots \\ \times \\ 0 \\ \times \\ \vdots \\ \times \end{bmatrix}.$$

- So, to transform A into an upper triangular matrix R, we can find a product of rotations Q such that $Q^T A = R$.
- It is easy to see that $O(mn)$ rotations are required.
- Each rotation takes $O(n)$ floating-point operations, so the entire process of computing the QR Factorization requires $O(mn^2)$ operations.
- It is important to note that the straightforward approach to computing the entries c and s of the Givens rotation,

  $$c = \frac{a}{\sqrt{a^2+b^2}}, \quad s = \frac{b}{\sqrt{a^2+b^2}},$$

  is not always advisable, because in floating-point arithmetic, the computation of $\sqrt{a^2+b^2}$ could overflow.
- To get around this problem, suppose that $|b| \geq |a|$. Then, we can instead compute

  $$t = \frac{a}{b}, \quad s = \frac{\operatorname{sgn}(b)}{\sqrt{1+t^2}}, \quad c = st, \tag{1}$$

  which is guaranteed not to overflow since the only number that is squared is at most one in magnitude.
- Similarly, if $|a| \geq |b|$, then we compute

  $$t = \frac{b}{a}, \quad c = \frac{\operatorname{sgn}(a)}{\sqrt{1+t^2}}, \quad s = ct. \tag{2}$$

Example. We illustrate how Givens rotations can be used to compute the QR Factorization of

$$A = \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 0.1270 & 0.5469 & 0.9572 \\ 0.9134 & 0.9575 & 0.4854 \\ 0.6324 & 0.9649 & 0.8003 \end{bmatrix}.$$

First, we compute a Givens rotation that, when applied to a_{41} and a_{51}, zeros a_{51}:

$$\begin{bmatrix} 0.8222 & -0.5692 \\ 0.5692 & 0.8222 \end{bmatrix}^T \begin{bmatrix} 0.9134 \\ 0.6324 \end{bmatrix} = \begin{bmatrix} 1.1109 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 4 and 5 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.8222 & -0.5692 \\ 0 & 0 & 0 & 0.5692 & 0.8222 \end{bmatrix}^T \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 0.1270 & 0.5469 & 0.9572 \\ 0.9134 & 0.9575 & 0.4854 \\ 0.6324 & 0.9649 & 0.8003 \end{bmatrix} = \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 0.1270 & 0.5469 & 0.9572 \\ 1.1109 & 1.3365 & 0.8546 \\ 0 & 0.2483 & 0.3817 \end{bmatrix}.$$

Next, we compute a Givens rotation that, when applied to a_{31} and a_{41}, zeros a_{41}:

$$\begin{bmatrix} 0.1136 & -0.9935 \\ 0.9935 & 0.1136 \end{bmatrix}^T \begin{bmatrix} 0.1270 \\ 1.1109 \end{bmatrix} = \begin{bmatrix} 1.1181 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 3 and 4 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.1136 & -0.9935 & 0 \\ 0 & 0 & 0.9935 & 0.1136 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 0.1270 & 0.5469 & 0.9572 \\ 1.1109 & 1.3365 & 0.8546 \\ 0 & 0.2483 & 0.3817 \end{bmatrix} = \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 1.1181 & 1.3899 & 0.9578 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix}.$$

Next, we compute a Givens rotation that, when applied to a_{21} and a_{31}, zeros a_{31}:

$$\begin{bmatrix} 0.6295 & -0.7770 \\ 0.7770 & 0.6295 \end{bmatrix}^T \begin{bmatrix} 0.9058 \\ 1.1181 \end{bmatrix} = \begin{bmatrix} 1.4390 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 2 and 3 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.6295 & -0.7770 & 0 & 0 \\ 0 & 0.7770 & 0.6295 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 0.9058 & 0.2785 & 0.9706 \\ 1.1181 & 1.3899 & 0.9578 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix} = \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 1.4390 & 1.2553 & 1.3552 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix}.$$

To complete the first column, we compute a Givens rotation that, when applied to a_{11} and a_{21}, zeros a_{21}:

$$\begin{bmatrix} 0.4927 & -0.8702 \\ 0.8702 & 0.4927 \end{bmatrix}^T \begin{bmatrix} 0.8147 \\ 1.4390 \end{bmatrix} = \begin{bmatrix} 1.6536 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 1 and 2 yields

$$\begin{bmatrix} 0.4927 & -0.8702 & 0 & 0 & 0 \\ 0.8702 & 0.4927 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 0.8147 & 0.0975 & 0.1576 \\ 1.4390 & 1.2553 & 1.3552 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix} = \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix}.$$

Moving to the second column, we compute a Givens rotation that, when applied to a_{42} and a_{52}, zeros a_{52}:

$$\begin{bmatrix} 0.8445 & 0.5355 \\ -0.5355 & 0.8445 \end{bmatrix}^T \begin{bmatrix} -0.3916 \\ 0.2483 \end{bmatrix} = \begin{bmatrix} 0.4636 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 4 and 5 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.8445 & 0.5355 \\ 0 & 0 & 0 & -0.5355 & 0.8445 \end{bmatrix}^T \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.3916 & -0.8539 \\ 0 & 0.2483 & 0.3817 \end{bmatrix} = \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.4636 & -0.9256 \\ 0 & 0 & -0.1349 \end{bmatrix}.$$

This rotation does not change the first column, because both of the entries of the first column that would be affected are already equal to zero. Next, we compute a Givens rotation that, when applied to a_{32} and a_{42}, zeros a_{42}:

$$\begin{bmatrix} 0.8177 & 0.5757 \\ -0.5757 & 0.8177 \end{bmatrix}^T \begin{bmatrix} 0.6585 \\ -0.4636 \end{bmatrix} = \begin{bmatrix} 0.8054 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 3 and 4 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.8177 & 0.5757 & 0 \\ 0 & 0 & -0.5757 & 0.8177 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.6585 & -0.1513 \\ 0 & -0.4636 & -0.9256 \\ 0 & 0 & -0.1349 \end{bmatrix} =$$

$$\begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.8054 & 0.4091 \\ 0 & 0 & -0.8439 \\ 0 & 0 & -0.1349 \end{bmatrix}.$$

Next, we compute a Givens rotation that, when applied to a_{22} and a_{32}, zeros a_{32}:

$$\begin{bmatrix} 0.5523 & -0.8336 \\ 0.8336 & 0.5523 \end{bmatrix}^T \begin{bmatrix} 0.5336 \\ 0.8054 \end{bmatrix} = \begin{bmatrix} 0.9661 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 3 and 4 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5523 & -0.8336 & 0 & 0 \\ 0 & 0.8336 & 0.5523 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.5336 & 0.5305 \\ 0 & 0.8054 & 0.4091 \\ 0 & 0 & -0.8439 \\ 0 & 0 & -0.1349 \end{bmatrix} = \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.9661 & 0.6341 \\ 0 & 0 & -0.2163 \\ 0 & 0 & -0.8439 \\ 0 & 0 & -0.1349 \end{bmatrix}.$$

Moving to the third column, we compute a Givens rotation that, when applied to a_{43} and a_{53}, zeros a_{53}:

$$\begin{bmatrix} 0.9875 & -0.1579 \\ 0.1579 & 0.9875 \end{bmatrix}^T \begin{bmatrix} -0.8439 \\ -0.1349 \end{bmatrix} = \begin{bmatrix} 0.8546 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 4 and 5 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.9875 & -0.1579 \\ 0 & 0 & 0 & 0.1579 & 0.9875 \end{bmatrix}^T \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.9661 & 0.6341 \\ 0 & 0 & -0.2163 \\ 0 & 0 & -0.8439 \\ 0 & 0 & -0.1349 \end{bmatrix} = \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.9661 & 0.6341 \\ 0 & 0 & -0.2163 \\ 0 & 0 & -0.8546 \\ 0 & 0 & 0 \end{bmatrix}.$$

Finally, we compute a Givens rotation that, when applied to a_{33} and a_{43}, zeros a_{43}:

$$\begin{bmatrix} 0.2453 & -0.9694 \\ 0.9694 & 0.2453 \end{bmatrix}^T \begin{bmatrix} -0.2163 \\ -0.8546 \end{bmatrix} = \begin{bmatrix} 0.8816 \\ 0 \end{bmatrix}.$$

Applying this rotation to rows 3 and 4 yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.2453 & -0.9694 & 0 \\ 0 & 0 & 0.9694 & 0.2453 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.9661 & 0.6341 \\ 0 & 0 & -0.2163 \\ 0 & 0 & -0.8546 \\ 0 & 0 & 0 \end{bmatrix} =$$

$$\begin{bmatrix} 1.6536 & 1.1405 & 1.2569 \\ 0 & 0.9661 & 0.6341 \\ 0 & 0 & -0.8816 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = R.$$

Applying these Givens rotations, in the same order, to the *columns* of the identity matrix yields the orthogonal matrix $Q = G_{51}G_{41}G_{31}G_{21}G_{52}G_{42}G_{32}G_{53}G_{43}$ such that $Q^T A = R$ is upper triangular. □

Now, we can describe the entire algorithm for computing the QR Factorization of $A \in \mathbb{R}^{m\times n}$ using Givens rotations.

- Let $\mathbf{v} \in \mathbb{R}^n$ and let $v_i = a$ and $v_j = b$, with $j > i$.
- We compute $[c, s] = \texttt{givens}(a, b)$, where `givens` is a function that implements equations (**??**) and (**??**).
- We denote by $G(i, j, c, s)$ be the $m \times m$ Givens rotation matrix that rotates the ith and jth elements of the vector $\mathbf{v}$ clockwise by the angle θ such that $\cos\theta = c$ and $\sin\theta = s$.
- Then, in the updated vector $\mathbf{u} = G(i, j, c, s)^T \mathbf{v}$, $u_i = r = \sqrt{a^2 + b^2}$ and $u_j = 0$.

Based on the preceding example, the QR Factorization of an $m \times n$ matrix A is then computed as follows, using such Givens rotations.

Algorithm. (QR Factorization via Givens rotations) Let $m \geq n$ and let $A \in \mathbb{R}^{m\times n}$ have full column rank. The following algorithm uses Givens rotations to compute the QR Factorization $A = QR$, where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper triangular.

$Q = I$
$R = A$
for $j = 1, 2, \ldots, n$ **do**
  for $i = m, m-1, \ldots, j+1$ **do**
    $[c, s] = \texttt{givens}(r_{i-1,j}, r_{ij})$
    $R = G(i-1, i, c, s)^T R$
    $Q = QG(i-1, i, c, s)$
  end for
end for

- Note that the matrix Q is accumulated by *column* rotations of the identity matrix, because the matrix by which R is multiplied to reduce R to upper triangular form, a product of *row* rotations, is Q^T.
- We also note that in a practical implementation, the matrix $G(i, j, c, s)$ is not formed explicitly; rather, rows i and j of R are modified to compute $G(i, j, c, s)^T R$, or columns i and j of Q to compute $QG(i, j, c, s)$.
- We showed how to construct Givens rotations in order to rotate two elements of a column vector so that one element would be zero, and that approximately $mn - n^2/2$ such rotations could be used to transform A into an upper triangular matrix R.

- Because each rotation only modifies two rows of A, it is possible to interchange the order of rotations that affect different rows, and thus apply sets of rotations in parallel.
- This is the main reason why Givens rotations can be preferable to other approaches.
- Other reasons are that they are easy to use when the QR Factorization needs to be updated as a result of adding a row to A or deleting a column of A.
- They are also more efficient when A is sparse.

Homework Hints

3.2.20 Your `solveAxb` function must call `gausselim` (3.2.11, modified for partial pivoting), `forwsub` and `backsub` (Algorithm 3.1.3). To handle pivoting, `gausselim` must be modified as follows:

- To find the largest entry in `A(j : n, j)`, the relevant portion of the `j`th column of `A`, use the MATLAB function `max` with *two* outputs: `[v,i]=max(x)` finds the largest entry in the input vector `x`, stores the maximum value in the output `v`, and stores the *index* within `x` at which the maximum value is found in the output `i`. This will help you determine which row to swap with row `j`.
- You must keep in mind, though, that `i` is *not* the index of the row to swap, except when `j` is 1, because otherwise it is the index within a *portion* of a column of `A`. Therefore it must be adjusted accordingly.
- Consider an example such as when `j=3`, and the fourth entry of `A(3:n,3)` turns out to be the largest (that is, `i` is 4). That means the *sixth* row is the one to swap with the third row, because the first two entries of the third column were excluded.
- Also, keep in mind that you want the largest entry in absolute value, so use the `abs` function to take the absolute value of any numeric entity such as a vector.
- To perform the actual swap, use MATLAB's indexing features. To swap row `j` with row `p` within `A`, use the statement
  ```
  A([ j p ],:)  = A([ p j ],:)
  ```
  The `:` here means to take all columns of the indicated rows. This statement extracts rows `p` and `j` of `A`, in that order, and stores them in rows `j` and `p` of `A`, in that order.
- To keep track of row interchanges, to apply them to `b` later for solving the system `L*y=P*b`, there is no need to construct a full permutation matrix `P`, which would be a waste of storage since most of its entries are zero.
- Instead, before Gaussian elimination create a *vector* consisting of the indices 1 to `n` using the colon operator: `P=1:n`.
- Then, as rows are swapped within `A`, use the same approach to swap indices within `P`, except that because it is a row vector, and therefore uses only one index instead of two, you would not use `,:` from above. The final vector `P` must be returned as an output from `gausselim` since it is needed during forward substitution.
- Then, to rearrange `b`, you can simply use the indices in `P` as indices into `b`: `b(P)` is the rearranged vector.

3.3.8 To show that if A is symmetric positive definite (SPD), so is A^{-1}: use properties of SPD matrices to see why A is invertible (what do we know about the determinant?). To show that A^{-1} is positive definite: need to show that $\mathbf{x}^T A^{-1}\mathbf{x} > 0$ for nonzero $\mathbf{x}$ based on the same being true of A. How? If A is invertible, then for any nonzero vector $\mathbf{x}$ there exists a unique vector $\mathbf{y}$ such that $A\mathbf{y} = \mathbf{x}$, or $\mathbf{y} = A^{-1}\mathbf{x}$.

3.4.1 Besides using the fact that if $(A + \epsilon E)$ is singular, then $(A + \epsilon E)\mathbf{x} = \mathbf{0}$ for some nonzero $\mathbf{x}$, need the *submultiplicative property* of matrix norms: $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$

4.1.10 If $A = R_1^T R_1$ has negative diagonal entries, how to "fix" them so that they are positive? The "proper" Cholesky factorization of A would be $A = \tilde{R}_1^T \tilde{R}_1$ where $\tilde{R}_1$ has positive diagonal entries. What is an "easy" way to obtain $\tilde{R}_1$ from R_1. Consider $A = R_1^T (I) R_1$