Answers to Exercises for Part 2

Exercises

0.1.

$$3x_2 = 6 \rightarrow x_2 = 2$$
$$2x_1 + 5x_2 = 8 \rightarrow x_1 = \frac{1}{2}(8 - 10) = -1$$

0.2. We will compute a Givens matrix by setting

$$c = \frac{3}{\sqrt{9+16}}, \quad s = \frac{4}{\sqrt{25}}.$$

Then if

$$\mathbf{Q} = \begin{bmatrix} 3/5 & 4/5 \\ 4/5 & -3/5 \end{bmatrix},$$

then

$$\mathbf{Q} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

so $z = 5$, the norm of the original vector.

0.3. Using Givens, $c = s = 3/\sqrt{18} = 1/\sqrt{2}$ so

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{Q}^T\mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 6 & 4 \\ 0 & 2 \end{bmatrix}$$

Alternatively, using Gram-Schmidt,

$$r_{11} = \sqrt{3^2 + 3^2} = 3\sqrt{2}$$

$$q_1 = \frac{1}{3\sqrt{2}} \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Then

$$r_{12} = q_1^T \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 4/\sqrt{2}$$

$$\hat{q}_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} - 4/\sqrt{2}q_1,$$

and r_{22} = the norm of this vector $= \sqrt{2}$, so $q_2 = \hat{q}_2/\sqrt{2}$. If we complete the arithmetic, we get the same $\mathbf{QR}$ as above.

0.4.

```
x = b;
```

```
for i=1:n,
  x(i) = x(i) / a(i,i);
  x(i+1:n) = x(i+1:n) - a(i+1:n,i)*x(i);
end
```

0.5. Some possibilities:

- Any right eigenvector $\mathbf{u}$ of $\mathbf{A}$ correponding to a zero eigenvalue satisfies $\mathbf{A}\mathbf{u} = 0\mathbf{u} = \mathbf{0}$. With roundoff, the computed eigenvalue will not be exactly zero, so we can choose the eigenvector of $\mathbf{A}$ corresponding to the smallest magnitude eigenvalue.
- Similarly, if $\mathbf{v}$ is a right singular vector of $\mathbf{A}$ corresponding to a zero singular value, then $\mathbf{A}\mathbf{v} = \mathbf{0}$, so choose a singular vector corresponding to the smallest singular value.
- Let $\mathbf{e}_n$ be the vector with a 1 in position n and zeros elsewhere. If we perform a rank-revealing QR decomposition of $\mathbf{A}^T$, so that $\mathbf{A}^T\mathbf{P} = \mathbf{Q}\mathbf{R}$, and let $\mathbf{q}_n$ be the last column of $\mathbf{Q}$, then $\mathbf{q}_n^T\mathbf{A}^T\mathbf{P} = \mathbf{q}_n^T\mathbf{Q}\mathbf{R} = \mathbf{e}_n^T\mathbf{R} = r_{nn}\mathbf{e}_n^T = \mathbf{0}$. Multiplying through by $\mathbf{P}^{-1}$ we see that $\mathbf{A}\mathbf{q}_n = \mathbf{0}$, so choose $\mathbf{z} = \mathbf{q}_n$.

0.6. Note that after we finish the iteration $k = 1$, we have $\mathbf{q}_{k+1} = \mathbf{q}_{k+1} - r_{1,k+1}\mathbf{q}_1$, so

$$\mathbf{q}_1^T\mathbf{q}_{k+1} = \mathbf{q}_1^T\mathbf{q}_{k+1} - r_{1,k+1}\mathbf{q}_1^T\mathbf{q}_1 = 0$$

by the definition of $r_{1,k+1}$ and the fact that $\mathbf{q}_1^T\mathbf{q}_1 = 1$.

Assume that after we finish iteration $i = j-1$, for a given value of k, we have $\mathbf{q}_\ell^T\mathbf{q}_{k+1} = 0$ for $\ell \le j-1$ and $\mathbf{q}_j^T\mathbf{q}_\ell = 0$ for $j < \ell \le k$. After we finish iteration $i = j$ for that value of k, we have $\mathbf{q}_j^T\mathbf{q}_{k+1} = 0$ by the same argument we used above, and we also have that $\mathbf{q}_\ell^T\mathbf{q}_{k+1} = 0$, for $\ell \le j-1$, since all we have done to $\mathbf{q}_{k+1}$ is to add a multiple of $\mathbf{q}_j$ to it, and $\mathbf{q}_j$ is orthogonal to $\mathbf{q}_\ell$. Thus, after iteration j, $\mathbf{q}_j^T\mathbf{q}_{k+1} = 0$ for $\ell \le j$, and the induction is complete when $j = k$ and $k = n-1$.

0.7.

$$\mathbf{A}_{new} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} a_{n+1,1} & \dots & a_{n+1,n} & a_{n+1,n+1} - 1 \end{bmatrix}$$
$$+ \begin{bmatrix} a_{1,n+1} \\ a_{2,n+1} \\ \vdots \\ a_{n,n+1} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} 0 & a_{1,n+1} \\ 0 & a_{2,n+1} \\ \vdots & \vdots \\ 0 & a_{n,n+1} \\ 1 & a_{n+1,n+1}-1 \end{bmatrix} \begin{bmatrix} a_{n+1,1} & \dots & a_{n+1,n} & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

So we can take

$$\mathbf{Z} = -\begin{bmatrix} 0 & a_{1,n+1} \\ 0 & a_{2,n+1} \\ \vdots & \vdots \\ 0 & a_{n,n+1} \\ 1 & a_{n+1,n+1}-1 \end{bmatrix}; \mathbf{V}^T = \begin{bmatrix} a_{n+1,1} & \dots & a_{n+1,n} & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

0.8. n Givens rotations.

0.9.

Form $\mathbf{y} = \mathbf{U}^T\mathbf{b}$	n^2 multiplications
Form $\mathbf{z} = \mathbf{\Sigma}^{-1}\mathbf{y}$	n multiplications
Form $\mathbf{x} = \mathbf{V}\mathbf{z}$	n^2 multiplications
Total:	$2n^2 + n$ multiplications

0.10. Since $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, we see that $||\mathbf{Q}\mathbf{y}||_2^2 = (\mathbf{Q}\mathbf{y})^T(\mathbf{Q}\mathbf{y}) = \mathbf{y}^T\mathbf{Q}^T\mathbf{Q}\mathbf{y} = \mathbf{y}^T\mathbf{y} = ||\mathbf{y}||_2^2$, Since norms are nonnegative quantities, take the square root and conclude that $||\mathbf{Q}\mathbf{y}||_2 = ||\mathbf{y}||_2$.

0.11.
```
s = zeros(m,1);
for j=1:n,
  s = s + abs(A(:,j));
end
```

0.12. (a) Find the null space of a matrix: QR (fast; relatively stable) or SVD (slower but more reliable)
(b) Solve a least squares problem: QR when the matrix is well conditioned. Don't try QR if the matrix is *not* well-conditioned; use the SVD method.
(c) Determine the rank of a matrix: RR-QR (fast, relatively stable); SVD (slower but more reliable).
(d) Find the determinant of a matrix: LU with pivoting.
(e) Determine whether a symmetric matrix is positive definite: Cholesky or Eigendecomposition (slower but more reliable) The $\mathbf{L}\mathbf{L}^T$ version of Cholesky will break down if the matrix has a negative eigenvalue by taking the square root of a negative number, so it is a good diagnostic. If the matrix is singular, (positive semi-definite), then you will get a 0 on the main diagonal, but with round-off error, this will be impossible to detect.

0.13.
```
for i=1:3,
  W = givens(A(i:i+1,i));
  %Note that the next instruction just operates on the part
  %of A that changes.  It is wasteful to do multiplications
  %on the rest.
  A(i:i+1,i:n) = W * A(i:i+1,i:n);
```

```
end
```

0.14. We use several facts to get an algorithm that is $O(kn^2)$ instead of $O(n^3)$:

- $\mathbf{x} = (\mathbf{A} - \mathbf{Z}\mathbf{V}^T)^{-1}\mathbf{b} = (\mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{Z}(\mathbf{I} - \mathbf{V}^T\mathbf{A}^{-1}\mathbf{Z})^{-1}\mathbf{V}^T\mathbf{A}^{-1})\mathbf{b}$.
- Forming $\mathbf{A}^{-1}$ from $\mathbf{LU}$ takes $O(n^3)$ operations, but forming $\mathbf{A}^{-1}\mathbf{b}$ as $\mathbf{U}\backslash(\mathbf{L}\backslash\mathbf{b})$ uses forward and backward substitution and just takes $O(n^2)$.
- $(\mathbf{I} - \mathbf{V}^T\mathbf{A}^{-1}\mathbf{Z})$ is only $k \times k$, so factoring it is cheap: $O(k^3)$.
- Matrix multiplication is associative.

```
y = U  (L  b);
Zh = U  (L  Z);
t = (eye(k) - V'*Zh)  (V'*y);
x = y + Zh*t;
```

0.15. TBD.

0.16. TBD.