Chapter 4

Least-norm problems

We now consider underdetermined sets of linear equations

$$Ax = b$$

with $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$ and $m < n$. An underdetermined set of equations usually has infinitely many solutions. We can use the additional freedom to choose a solution that is more interesting than the others in some sense. The most common choice is to use the solution with the smallest norm $\|x\|$.

4.1 The least-norm solution

We use the notation

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Ax = b \end{array}$$

to denote the problem of finding an $\hat{x}$ that satisfies $A\hat{x} = b$, and $\|\hat{x}\|^2 \leq \|x\|^2$ for all x that satisfy $Ax = b$. In other words, $\hat{x}$ is the solution of the equations $Ax = b$ with the smallest norm. This is called a *least-norm problem* or *minimum-norm* problem.

Example

We take

$$A = \left[\begin{array}{rrr} 1 & -1 & 2 \\ 1 & 1 & -1 \end{array}\right], \qquad b = \left[\begin{array}{c} 1 \\ 0 \end{array}\right].$$

The corresponding least-norm problem is

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} & x_1 - x_2 + 2x_3 = 1 \\ & x_1 + x_2 - x_3 = 0. \end{array}$$

We can solve this particular problem as follows. (Systematic methods will be discussed later.) Using the second equation, we express x_3 in terms of x_1 and x_2

as $x_3 = x_1 + x_2$. Substituting in the first equation yields

$$x_1 - x_2 + 2(x_1 + x_2) = 3x_1 + x_2 = 1.$$

We eliminate x_2 from this equation,

$$x_2 = 1 - 3x_1.$$

This leaves us with one remaining variable x_1, which we have to determine by minimizing

$$\begin{aligned} x_1^2 + x_2^2 + x_3^2 &= x_1^2 + (1-3x_1)^2 + (x_1 + 1 - 3x_1)^2 \\ &= x_1^2 + (1-3x_1)^2 + (1-2x_1)^2. \end{aligned}$$

Setting to zero the derivative with respect to x_1 gives

$$2x_1 - 6(1-3x_1) - 4(1-2x_1) = 0.$$

The solution is $x_1 = 5/14$, $x_2 = 1 - 3x_1 = -1/14$, and $x_3 = x_1 + x_2 = 2/7$.

4.2 The normal equations

The counterpart of the normal equations of a least-squares problem is given by the following result. If $A \in \mathbf{R}^{m\times n}$ has rank m, then the solution of the least-norm problem

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Ax = b \end{array}$$

is unique and given by

$$\hat{x} = A^T(AA^T)^{-1}b. \tag{4.1}$$

This result means that we can solve the least-norm problem by solving

$$(AA^T)z = b, \tag{4.2}$$

and then calculating $\hat{x} = A^T z$. The equations (4.2) are a set of m linear equations in m variables, and are called the *normal equations* associated with the least-norm problem.

Proof

We first show that AA^T is positive definite. We have

$$x^T AA^T x = (A^T x)^T (A^T x) = \|A^T x\|^2 \geq 0$$

for all x. Moreover if $\mathbf{rank}\, A = m$, then $\|A^T x\| = 0$ only if $x = 0$. This means that AA^T is positive definite, hence nonsingular.

Next we have to verify that $\hat{x}$ satisfies $A\hat{x} = b$:

$$A\hat{x} = (AA^T)(AA^T)^{-1}b = b.$$

Finally, we have to show that any other solution of the equations has a norm greater than $\|\hat{x}\|$. Suppose x satisfies $Ax = b$. We have

$$\|x\|^2 = \|\hat{x} + (x - \hat{x})\|^2 = \|\hat{x}\|^2 + \|x - \hat{x}\|^2 + 2\hat{x}^T(x - \hat{x}).$$

The third term turns out to be zero, since

$$\begin{aligned} \hat{x}^T(x - \hat{x}) &= (A^T(AA^T)^{-1}b)^T(x - \hat{x}) \\ &= b^T(AA^T)^{-1}A(x - \hat{x}) \\ &= 0 \end{aligned}$$

because $Ax = A\hat{x} = b$. We therefore have

$$\|x\|^2 = \|x - \hat{x}\|^2 + \|\hat{x}\|^2 \geq \|\hat{x}\|^2$$

with equality only if $x = \hat{x}$. In conclusion, if $Ax = b$ and $x \neq \hat{x}$ then

$$\|x\|^2 > \|\hat{x}\|^2.$$

This proves that $\hat{x}$ is the unique solution of the least-norm problem.

4.3 Solving least-norm problems

We can solve the normal equations (4.2) using the Cholesky factorization of A^TA or the QR factorization of A^T.

4.3.1 Cholesky factorization

If **rank** $A = m$, we can solve the normal equations (4.2) using the Cholesky factorization.

Algorithm 4.1 Solving least-norm problems by Cholesky factorization.

given $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ with **rank** $A = m$.

1. Form $C = AA^T$.
2. Compute the Cholesky factorization $C = LL^T$.
3. Solve $Lw = b$ by forward substitution.
4. Solve $L^Tz = w$ by backward substitution.
5. Compute $x = A^Tz$.

The cost is nm^2 for step 1 (if we exploit the fact that C is symmetric), $(1/3)m^3$ for step 2, m^2 each for step 3 and step 4, and $2mn$ in step 5, which gives a total flop count of $nm^2 + (1/3)m^3 + 2m^2 + 2mn$, or roughly

$$nm^2 + (1/3)m^3.$$

Example

We solve the least-norm problem defined by

$$A = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 0 & 1/2 & 1/2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4.3}$$

The normal equations $AA^Tz = b$ are

$$\begin{bmatrix} 4 & 2 \\ 2 & 3/2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The Cholesky factorization of AA^T is

$$\begin{bmatrix} 4 & 2 \\ 2 & 3/2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1/\sqrt{2} \end{bmatrix}.$$

To find z we first solve

$$\begin{bmatrix} 2 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

by forward substitution, which yields $w_1 = 0$, $w_2 = \sqrt{2}$. Next we solve

$$\begin{bmatrix} 2 & 1 \\ 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix}$$

by backward substitution and find $z_1 = -1$, $z_2 = 2$. Finally, we obtain the solution x from the matrix-vector product

$$x = A^Tz = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 1/2 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

4.3.2 Solving least-norm problems by QR factorization

An alternative method is based on the QR factorization of the matrix A^T.

Suppose $A^T = QR$ with $Q^TQ = I$ and R upper triangular with positive diagonal elements. We can simplify the expression (4.1) as follows:

$$\begin{aligned} \hat{x} = A^T(AA^T)^{-1}b &= QR(R^TQ^TQR)^{-1}b \\ &= QR(R^TR)^{-1}b \\ &= QRR^{-1}R^{-T}b \\ &= QR^{-T}b. \end{aligned}$$

To find $\hat{x}$ we first compute $R^{-T}b$ (*i.e.*, solve the equation $R^Tz = b$), and then multiply with Q.

Algorithm 4.2 SOLVING LEAST-NORM PROBLEMS BY QR FACTORIZATION.

given $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ with $\mathbf{rank}\, A = m$.

1. Compute the QR factorization $A^T = QR$.
2. Solve $R^T z = b$ by forward substitution.
3. Compute $x = Qz$.

The cost is $2nm^2$ (for the QR factorization), plus m^2 for the forward substitution, plus $2mn$ for the matrix-vector product Qz. The total is $2nm^2 + m^2 + 2mn$ or roughly

$$2nm^2.$$

The advantages and disadvantages of the two methods are exactly the same as for least-squares problems. The QR factorization method is slower than the Cholesky factorization method (by a factor of about two if $n \gg m$), but it is more accurate. It is the preferred method if n and m are not too large. For very large sparse problems, the Cholesky factorization method is useful, because it is much more efficient than the QR factorization method.

Example

We use the matrix A and b given in (4.3). The QR factorization of A^T is

$$\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/\sqrt{2} \\ -1/2 & 1/\sqrt{2} \\ 1/2 & 0 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1/\sqrt{2} \end{bmatrix}.$$

The second step is to solve $R^T z = b$:

$$\begin{bmatrix} 2 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The solution is $z_1 = 0$, $z_2 = \sqrt{2}$. From z, we find the solution of the least-norm problem by evaluating $x = Qz$:

$$x = \begin{bmatrix} 1/2 & 1/\sqrt{2} \\ -1/2 & 1/\sqrt{2} \\ 1/2 & 0 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

4.3.3 Solving least-norm problems in Matlab

If $A \in \mathbf{R}^{m\times n}$ with $\mathbf{rank}\, A = m$, then the Matlab command `x = A\b` computes a solution to $Ax = b$, but it is *not* the least-norm solution.

We can use the command `x = A'*((A*A')\b)` to compute the least-norm solution using the Cholesky factorization method. (Matlab will recognize that AA^T is positive definite and use the Cholesky factorization to solve $AA^T z = b$.)

We can also use the QR factorization method, using the code

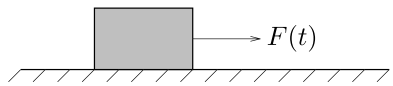

Figure 4.1 Unit mass moving along a straight line.

```
[Q,R] = qr(A');
Q = Q(:,1:m);
R = R(1:m:,:);
x = Q*(R'\b);
```

The second and third lines are added for the following reason. The definition of the QR factorization in Matlab is slightly different from the definition in §3.5. For an $m \times n$ matrix A with rank m, the Matlab command `[Q,R] = qr(A')` will return an $n \times n$ matrix Q and $n \times m$ matrix R that satisfy $A^T = QR$. The matrix R has the form

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where R_1 is an $m \times m$ upper triangular matrix with nonzero diagonal elements. The matrix Q is orthogonal, so if we partition Q as

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix},$$

with $Q_1 \in \mathbf{R}^{n\times m}$, and $Q_2 \in \mathbf{R}^{n\times(n-m)}$, then

$$Q^TQ = \begin{bmatrix} Q_1^TQ_1 & Q_1^TQ_2 \\ Q_2^TQ_1 & Q_2^TQ_2 \end{bmatrix} = I.$$

It follows that Q_1 is orthogonal ($Q_1^TQ_1 = I$), and that

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1R_1.$$

We conclude that $A^T = Q_1R_1$ is the QR factorization according to our definition (except that we choose the triangular matrix to have positive diagonal elements, while Matlab simply makes them nonzero, but that does not make much difference).

Example

We consider a simple optimal control problem. A unit mass, moving along a straight line, is subjected to a force $F(t)$, where $F(t) = x_i$ for $i-1 < t \leq i$, and $i = 1, \ldots, 10$. (See figures 4.1 and 4.2).

We denote the position of the mass at time t as $s(t)$, and we assume the initial position and velocity are zero:

$$s(0) = s'(0) = 0.$$

We are interested in finding an efficient (minimum-energy) force $F(t)$ that moves the mass in 10 steps to the position $s(10) = 1$, with final velocity $s'(10) = 0$. We

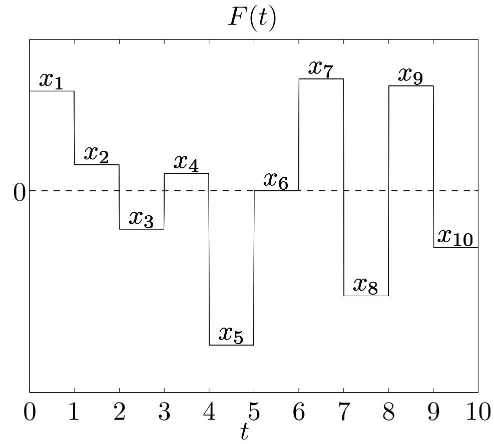

Figure 4.2 The force $F(t)$ is piecewise-constant, with $F(t) = x_i$ for $i - 1 \leq t \leq i$.

assume that the total energy used is proportional to

$$\int_{t=0}^{10} F(t)^2 \, dt = \sum_{i=1}^{10} x_i^2 = \|x\|^2.$$

By integrating Newton's law $s''(t) = F(t)$ twice, we can derive expressions for the velocity and position at time $t = 10$:

$$\begin{array}{rcl} s'(10) & = & x_1 + x_2 + \cdots + x_{10} \\ s(10) & = & (19/2)x_1 + (17/2)x_2 + \cdots + (1/2)x_{10}. \end{array}$$

The control problem can therefore be posed as a least-norm problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 + \cdots + x_9^2 + x_{10}^2 \\ \text{subject to} & \left[\begin{array}{ccccc} 1 & 1 & \cdots & 1 & 1 \\ 19/2 & 17/2 & \cdots & 3/2 & 1/2 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_9 \\ x_{10} \end{array}\right] = \left[\begin{array}{c} 0 \\ 1 \end{array}\right], \end{array}$$

and solved in Matlab as follows:

```
A = [ones(1,10); (19/2):-1:(1/2)];
x = A'*((A*A')\[0;1]);
```

The solution x, and the resulting position $s(t)$, are shown in figure 4.3. The norm of the least-norm solution x is 0.1101.

It is interesting to compare the least norm solution of $Ax = b$ with a few other solutions. An obvious choice of x that also satisfies $Ax = b$ is $x = (1, -1, 0, \ldots, 0)$. The resulting position is shown in figure 4.4. This solution has norm $\sqrt{2}$, but requires only two time steps, so we might call it the 'minimum-time' solution.

Another solution is $x = (0, \ldots, 0, 1, -1)$, which also has norm $\sqrt{2}$.

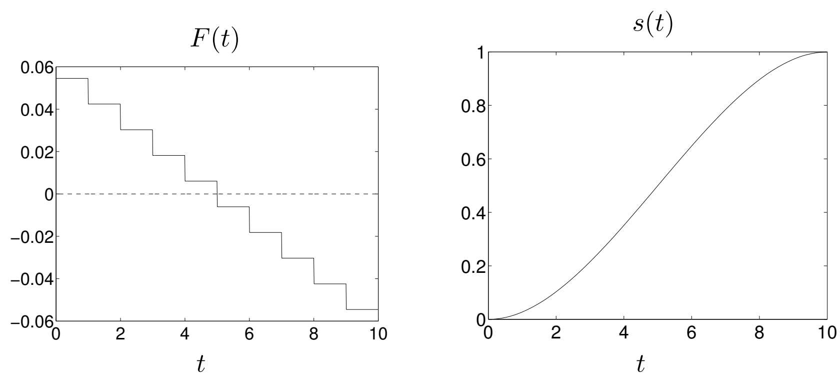

Figure 4.3 Left: The optimal force $F(t)$ that transfers the mass over a unit distance in 10 steps. Right: the resulting position of the mass $s(t)$.

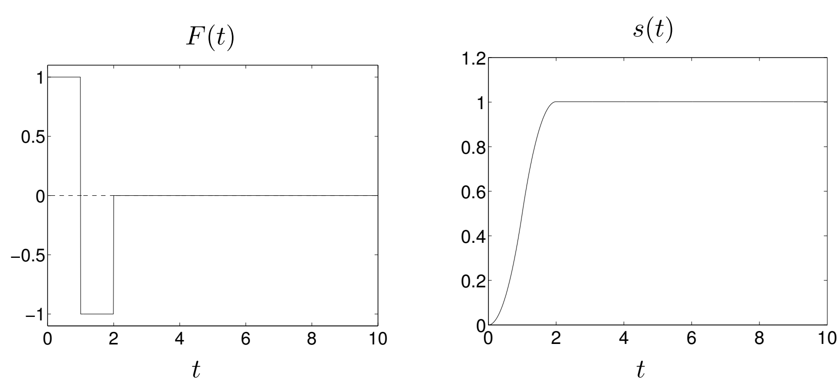

Figure 4.4 Left: A non-minimum norm force $F(t)$ that that transfers the mass over a unit distance in 10 steps. Right: the resulting position of the mass $s(t)$.

Exercises

Examples and applications

4.1 u, v, and w are three points in $\mathbf{R}^4$, given by

$$u = \begin{bmatrix} -1 \\ 7 \\ 1 \\ 1 \end{bmatrix}, \qquad v = \begin{bmatrix} -5 \\ -5 \\ 5 \\ 0 \end{bmatrix}, \qquad w = \begin{bmatrix} -2 \\ 9 \\ -1 \\ 0 \end{bmatrix}.$$

Each of the following four sets is a (two-dimensional) plane in $\mathbf{R}^4$.

(a) $\mathcal{H}_1 = \{x \in \mathbf{R}^4 \mid x = \alpha u + \beta v \text{ for some } \alpha, \beta \in \mathbf{R}\}$.

(b) $\mathcal{H}_2 = \{x \in \mathbf{R}^4 \mid x = \alpha u + \beta v + w \text{ for some } \alpha, \beta \in \mathbf{R}\}$.

(c) $\mathcal{H}_3 = \{x \in \mathbf{R}^4 \mid u^T x = 0 \text{ and } v^T x = 0\}$.

(d) $\mathcal{H}_4 = \{x \in \mathbf{R}^4 \mid u^T x = 1 \text{ and } v^T x = 1\}$.

For each set $\mathcal{H}_i$, find the projection of the point $y = (1, 1, 1, 1)$ on $\mathcal{H}_i$ (*i.e.*, find the point in $\mathcal{H}_i$ closest to y).

Hint. Formulate each problem as a least-squares or a least-norm problem, and solve it using Matlab.

4.2 *Minimum-energy optimal control.* A simple model of a vehicle moving in one dimension is given by

$$\begin{bmatrix} s_1(t+1) \\ s_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t), \qquad t = 0, 1, 2, \ldots.$$

$s_1(t)$ is the position at time t, $s_2(t)$ is the velocity at time t, and $u(t)$ is the actuator input. Roughly speaking, the equations state that the actuator input affects the velocity, which in turn affects the position. The coefficient 0.95 means that the velocity decays by 5% in one sample period (for example, because of friction), if no actuator signal is applied. We assume that the vehicle is initially at rest at position 0: $s_1(0) = s_2(0) = 0$.

We will solve the *minimum energy optimal control problem*: for a given time horizon N, choose inputs $u(0), \ldots, u(N-1)$ so as to minimize the total energy consumed, which we assume is given by

$$E = \sum_{t=0}^{N-1} u(t)^2.$$

In addition, the input sequence must satisfy the constraint $s_1(N) = 10$, $s_2(N) = 0$. In other words, our task is to bring the vehicle to the final position $s_1(N) = 10$ with final velocity $s_2(N) = 0$, as efficiently as possible.

(a) Formulate the minimum energy optimal control problem as a least-norm problem

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Ax = b. \end{array}$$

Clearly state what the variables x, and the problem data A and b are.

(b) Solve the problem for $N = 30$. Plot the optimal $u(t)$, the resulting position $s_1(t)$, and velocity $s_2(t)$.

(c) Solve the problem for $N = 2, 3, \ldots, 29$. For each N calculate the energy E consumed by the optimal input sequence. Plot E versus N. (The plot will look better if you use a logarithmic scale for E, *i.e.*, `semilogy` instead of `plot`.)

(d) Suppose we allow the final position to deviate from 10. However, if $s_1(N) \neq 10$, we have to pay a penalty, equal to $(s_1(N) - 10)^2$. The problem is to find the input sequence that minimizes the sum of the energy E consumed by the input and the terminal position penalty,

$$\sum_{t=0}^{N-1} u(t)^2 + (s_1(N) - 10)^2,$$

subject to the constraint $s_2(N) = 0$.

Formulate this problem as a least-norm problem, and solve it for $N = 30$. Plot the optimal input signals $u(t)$, the resulting position $s_1(t)$ and the resulting velocity $s_2(t)$.

4.3 Two vehicles are moving along a straight line. For the first vehicle we use the same model as in exercise 4.2:

$$\begin{bmatrix} s_1(t+1) \\ s_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t), \qquad t = 0, 1, 2, \ldots,$$

$s_1(t)$ is the position at time t, $s_2(t)$ is the velocity at time t, and $u(t)$ is the actuator input. We assume that the vehicle is initially at rest at position 0: $s_1(0) = s_2(0) = 0$.

The model for the second vehicle is

$$\begin{bmatrix} p_1(t+1) \\ p_2(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0.8 \end{bmatrix} \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} v(t), \qquad t = 0, 1, 2, \ldots,$$

$p_1(t)$ is the position at time t, $p_2(t)$ is the velocity at time t, and $v(t)$ is the actuator input. We assume that the second vehicle is initially at rest at position 1: $p_1(0) = 1$, $p_2(0) = 0$.

Formulate the following problem as a least-norm problem, and solve it in Matlab. Find the control inputs $u(0), u(1), \ldots, u(19)$ and $v(0), v(1), \ldots, v(19)$ that minimize the total energy

$$\sum_{t=0}^{19} u(t)^2 + \sum_{t=0}^{19} v(t)^2$$

and satisfy the following three conditions:

$$s_1(20) = p_1(20), \qquad s_2(20) = 0, \qquad p_2(20) = 0. \tag{4.4}$$

In other words, at time $t = 20$ the two vehicles must have velocity zero, and be at the same position. (The final position itself is not specified, *i.e.*, you are free to choose any value as long as $s_1(20) = p_1(20)$.)

Plot the positions $s_1(t)$ and $p_1(t)$ of the two vehicles, for $t = 1, 2, \ldots, 20$.

Solving least-norm problems via QR factorization

4.4 Explain how you would solve the following problems using the QR factorization. State clearly

- what the matrices are that you factor, and why you know that they have a QR factorization
- how you obtain the solution of each problem from the results of the QR factorizations
- what the cost (number of flops) is of your method.

(a) Find the solution of $Ax = b$ with the smallest value of $\sum_{i=1}^n d_i x_i^2$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n d_i x_i^2 \\ \text{subject to} & Ax = b. \end{array}$$

The problem data are $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, and $d \in \mathbf{R}^n$. We assume that $\mathbf{rank}(A) = m$ and $d_i > 0$ for all i.

(b) Find the solution of $Ax = b$ with the smallest value of $\|x - x_0\|^2$:

$$\begin{array}{ll} \text{minimize} & \|x - x_0\|^2 \\ \text{subject to} & Ax = b. \end{array}$$

The variable is $x \in \mathbf{R}^n$. The problem data are $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, and $x_0 \in \mathbf{R}^n$. We assume that $\mathbf{rank}(A) = m$.

(c) Find the solution of $Ax = b$ with the smallest value of $\|Cx\|^2$:

$$\begin{array}{ll} \text{minimize} & \|Cx\|^2 \\ \text{subject to} & Ax = b. \end{array}$$

The problem data are $C \in \mathbf{R}^{p\times n}$, $A \in \mathbf{R}^{m\times n}$, and $b \in \mathbf{R}^m$. We assume that $\mathbf{rank}(A) = m$, and $\mathbf{rank}\, C = n$.

(d) Find the solution of $Ax = b$ with the smallest value of $\|x\|^2 - c^T x$:

$$\begin{array}{ll} \text{minimize} & \|x\|^2 - c^T x \\ \text{subject to} & Ax = b. \end{array}$$

The problem data are $c \in \mathbf{R}^n$, $A \in \mathbf{R}^{m\times n}$, and $b \in \mathbf{R}^m$. We assume that $\mathbf{rank}(A) = m$.

(e) Find the solution of $Ax = b$ with the smallest value of $\|C(x - x_0)\|^2$:

$$\begin{array}{ll} \text{minimize} & \|C(x - x_0)\|^2 \\ \text{subject to} & Ax = b. \end{array}$$

$C \in \mathbf{R}^{p\times n}$, $A \in \mathbf{R}^{m\times n}$, $x_0 \in \mathbf{R}^n$, and $b \in \mathbf{R}^m$ are given. We assume that $\mathbf{rank}(A) = m$ and $\mathbf{rank}\, C = n$. Note that C is not necessarily square ($p \geq n$).

(f) Find the solution of $Ax = b$ with the smallest value of $\|Cx - d\|^2$:

$$\begin{array}{ll} \text{minimize} & \|Cx - d\|^2 \\ \text{subject to} & Ax = b. \end{array}$$

$C \in \mathbf{R}^{p\times n}$, $A \in \mathbf{R}^{m\times n}$, $d \in \mathbf{R}^p$ and $b \in \mathbf{R}^m$ are given. We assume that $\mathbf{rank}(A) = m$ and $\mathbf{rank}\, C = n$. Note that C is not necessarily square ($p \geq n$).

4.5 Show how to solve the following problems using the QR factorization of A. In each problem A is an $m \times n$ matrix with rank n. Clearly state the different steps in your method. Also give a flop count, including all the terms that are quadratic (order m^2, mn, or n^2), or cubic (order m^3, m^2n, mn^2, n^3).

If you know several methods, give the most efficient one.

(a) Solve the set of linear equations

$$\begin{bmatrix} 0 & A^T \\ A & I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}.$$

The variables are $x \in \mathbf{R}^n$, $y \in \mathbf{R}^m$.

(b) Solve the least-squares problem

$$\text{minimize} \quad 2\|Ax - b\|^2 + 3\|Ax - c\|^2.$$

The variable is $x \in \mathbf{R}^n$.

(c) Solve the least-norm problem

$$\begin{array}{ll} \text{minimize} & \|x\|^2 + \|y\|^2 \\ \text{subject to} & A^T x - 2A^T y = b. \end{array}$$

The variables are $x \in \mathbf{R}^m$ and $y \in \mathbf{R}^m$.

(d) Solve the quadratic minimization problem

$$\text{minimize} \quad x^T A^T A x + b^T x + c.$$

The variable is $x \in \mathbf{R}^n$.

4.6 Consider the underdetermined set of linear equations

$$Ax + By = b$$

where $b \in \mathbf{R}^p$, $A \in \mathbf{R}^{p\times p}$, and $B \in \mathbf{R}^{p\times q}$ are given. The variables are $x \in \mathbf{R}^p$ and $y \in \mathbf{R}^q$. We assume that A is nonsingular, and that $\mathbf{rank}(B) = q$ (which implies $q \leq p$). The equations are underdetermined, so there are infinitely many solutions. For example, we can pick any y, and solve the set of linear equations $Ax = b - By$ to find x.

Below we define four solutions that minimize some measure of the magnitude of x, or y, or both. For each of these solutions, describe the factorizations (QR, Cholesky, or LU) that you would use to calculate x and y. Clearly specify the matrices that you factor, and the type of factorization. If you know several methods, you should give the most efficient one.

(a) The solution x, y with the smallest value of $\|x\|^2 + \|y\|^2$

(b) The solution x, y with the smallest value of $\|x\|^2 + 2\|y\|^2$.

(c) The solution x, y with the smallest value of $\|y\|^2$.

(d) The solution x, y with the smallest value of $\|x\|^2$.