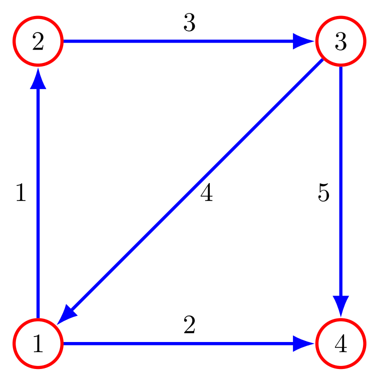

Figure 7.2 Directed graph with four vertices and five edges.

Image cropping. As a more interesting example, suppose that x is an image with $M \times N$ pixels, with M and N even. (That is, x is an MN-vector, with its entries giving the pixel values in some specific order.) Let y be the $(M/2) \times (N/2)$ image that is the upper left corner of the image x, *i.e.*, a cropped version. Then we have $y = Ax$, where A is an $(MN/4) \times (MN)$ selector matrix. The ith row of A is $e_{k_i}^T$, where k_i is the index of the pixel in x that corresponds to the ith pixel in y.

Permutation matrices. An $n \times n$ *permutation matrix* is one in which each column is a unit vector, and each row is the transpose of a unit vector. (In other words, A and A^T are both selector matrices.) Thus, exactly one entry of each row is one, and exactly one entry of each column is one. This means that $y = Ax$ can be expressed as $y_i = x_{\pi_i}$, where π is a permutation of $1, 2, \ldots, n$, *i.e.*, each integer from 1 to n appears exactly once in $\pi_1, \ldots, \pi_n$.

As a simple example consider the permutation $\pi = (3, 1, 2)$. The associated permutation matrix is

$$A = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right].$$

Multiplying a 3-vector by A re-orders its entries: $Ax = (x_3, x_1, x_2)$.

7.3 Incidence matrix

Directed graph. A *directed graph* consists of a set of *vertices* (or nodes), labeled $1, \ldots, n$, and a set of *directed edges* (or branches), labeled $1, \ldots, m$. Each edge is connected from one of the nodes and into another one, in which case we say the two nodes are connected or adjacent. Directed graphs are often drawn with the vertices as circles or dots, and the edges as arrows, as in figure 7.2. A directed graph can be described by its $n \times m$ *incidence matrix*, defined as

$$A_{ij} = \begin{cases} 1 & \text{edge } j \text{ points to node } i \\ -1 & \text{edge } j \text{ points from node } i \\ 0 & \text{otherwise.} \end{cases}$$

The incidence matrix is evidently sparse, since it has only two nonzero entries in each column (one with value 1 and other with value -1). The jth column is associated with the jth edge; the indices of its two nonzero entries give the nodes that the edge connects. The ith row of A corresponds to node i; its nonzero entries tell us which edges connect to the node, and whether they point into or away from the node. The incidence matrix for the graph shown in figure 7.2 is

$$A = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

A directed graph can also be described by its adjacency matrix, described on page 112. The adjacency and incidence matrices for a directed graph are closely related, but not the same. The adjacency matrix does not explicitly label the edges $j = 1, \ldots, m$. There are also some small differences in the graphs that can be represented using incidence and adjacency matrices. For example, self edges (that connect from and to the same vertex) cannot be represented in an incidence matrix.

7.3.1 Networks

In many applications a graph is used to represent a *network*, through which some commodity or quantity such as electricity, water, heat, or vehicular traffic flows. The edges of the graph represent the *paths* or *links* over which the quantity can move or flow, in either direction. If x is an m-vector representing a flow in the network, we interpret x_j as the flow (rate) along the edge j, with a positive value meaning the flow is in the direction of edge j, and negative meaning the flow is in the opposite direction of edge j. In a network, the direction of the edge or link does not specify the direction of flow; it only specifies which direction of flow we consider to be positive.

Flow conservation. When x represents a flow in a network, the matrix-vector product $y = Ax$ can be given a very simple interpretation. The n-vector $y = Ax$ can be interpreted as the vector of net flows, from the edges, into the nodes: y_i is equal to the total of the flows that come in to node i, minus the total of the flows that go out from node i. The quantity y_i is sometimes called the *flow surplus* at node i.

If $Ax = 0$, we say that *flow conservation* occurs, since at each node, the total in-flow matches the total out-flow. In this case the flow vector x is called a *circulation*. This could be used as a model of traffic flow (in a closed system), with the nodes

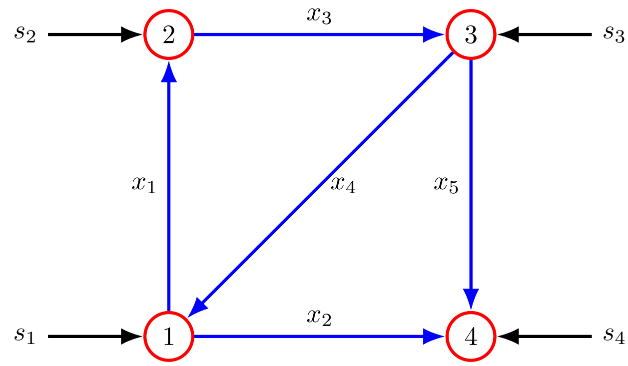

Figure 7.3 Network with four nodes and five edges, with source flows shown.

representing intersections and the edges representing road segments (one for each direction).

For a network described by the directed graph example above, the vector

$$x = (1, -1, 1, 0, 1)$$

is a circulation, since $Ax = 0$. This flow corresponds to a unit clockwise flow on the outer edges (1, 3, 5, and 2) and no flow on the diagonal edge (4). (Visualizing this explains why such vectors are called circulations.)

Sources. In many applications it is useful to include additional flows called *source flows* or *exogenous flows*, that enter or leave the network at the nodes, but not along the edges, as shown in figure 7.3. We denote these flows with an n-vector s. We can think of s_i as a flow that enters the network at node i from outside, *i.e.*, not from any edge. When $s_i > 0$ the exogenous flow is called a *source*, since it is injecting the quantity into the network at the node. When $s_i < 0$ the exogenous flow is called a *sink*, since it is removing the quantity from the network at the node.

Flow conservation with sources. The equation $Ax + s = 0$ means that the flow is conserved at each node, counting the source flow: The total of all incoming flow, from the incoming edges and exogenous source, minus the total outgoing flow from outgoing edges and exogenous sinks, is zero.

As an example, flow conservation with sources can be used as an approximate model of a power grid (ignoring losses), with x being the vector of power flows along the transmission lines, $s_i > 0$ representing a generator injecting power into the grid at node i, $s_i < 0$ representing a load that consumes power at node i, and $s_i = 0$ representing a substation where power is exchanged among transmission lines, with no generation or load attached.

For the example above, consider the source vector $s = (1, 0, -1, 0)$, which corresponds to an injection of one unit of flow into node 1, and the removal of one unit of flow at node 3. In other words, node 1 is a source, node 3 is a sink, and flow is conserved at nodes 2 and 4. For this source, the flow vector

$$x = (0.6,\ 0.3,\ 0.6,\ -0.1,\ -0.3)$$

satisfies flow conservation, *i.e.*, $Ax + s = 0$. This flow can be explained in words: The unit external flow entering node 1 splits three ways, with 0.6 flowing up, 0.3 flowing right, and 0.1 flowing diagonally up (on edge 4). The upward flow on edge 1 passes through node 2, where flow is conserved, and proceeds right on edge 3 towards node 3. The rightward flow on edge 2 passes through node 4, where flow is conserved, and proceeds up on edge 5 to node 3. The one unit of excess flow arriving at node 3 is removed as external flow.

Node potentials. A graph is also useful when we focus on the values of some quantity at each graph vertex or node. Let v be an n-vector, often interpreted as a *potential*, with v_i the potential value at node i. We can give a simple interpretation to the matrix-vector product $u = A^T v$. The m-vector $u = A^T v$ gives the potential differences across the edges: $u_j = v_l - v_k$, where edge j goes from node k to node l.

Dirichlet energy. When the m-vector $A^T v$ is small, it means that the potential differences across the edges are small. Another way to say this is that the potentials of connected vertices are near each other. A quantitative measure of this is the function of v given by

$$\mathcal{D}(v) = \|A^T v\|^2.$$

This function arises in many applications, and is called the *Dirichlet energy* (or *Laplacian quadratic form*) associated with the graph. It can be expressed as

$$\mathcal{D}(v) = \sum_{\text{edges }(k,l)} (v_l - v_k)^2,$$

which is the sum of the squares of the potential differences of v across all edges in the graph. The Dirichlet energy is small when the potential differences across the edges of the graph are small, *i.e.*, nodes that are connected by edges have similar potential values.

The Dirichlet energy is used as a measure the non-smoothness (roughness) of a set of node potentials on a graph. A set of node potentials with small Dirichlet energy can be thought of as smoothly varying across the graph. Conversely, a set of potentials with large Dirichlet energy can be thought of as non-smooth or rough. The Dirichlet energy will arise as a measure of roughness in several applications we will encounter later.

As a simple example, consider the potential vector $v = (1, -1, 2, -1)$ for the graph shown in figure 7.2. For this set of potentials, the potential differences across the edges are relatively large, with $A^T v = (-2, -2, 3, -1, -3)$, and the associated Dirichlet energy is $\|A^T v\|^2 = 27$. Now consider the potential vector $v = (1, 2, 2, 1)$. The associated edge potential differences are $A^T v = (1, 0, 0, -1, -1)$, and the Dirichlet energy has the much smaller value $\|A^T v\|^2 = 3$.

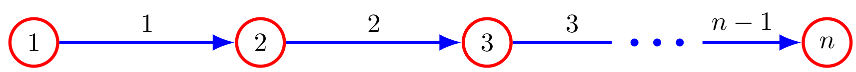

Figure 7.4 Chain graph.

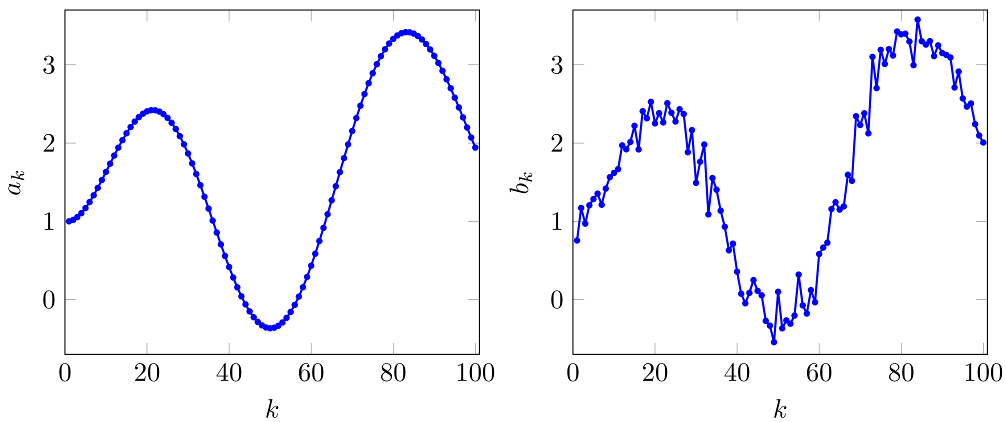

Figure 7.5 Two vectors of length 100, with Dirichlet energy $\mathcal{D}(a) = 1.14$ and $\mathcal{D}(b) = 8.99$.

Chain graph. The incidence matrix and the Dirichlet energy function have a particularly simple form for the *chain graph* shown in figure 7.4, with n vertices and $n-1$ edges. The $n \times (n-1)$ incidence matrix is the transpose of the difference matrix D described on page 119, in (6.5). The Dirichlet energy is then

$$\mathcal{D}(v) = \|Dv\|^2 = (v_2 - v_1)^2 + \cdots + (v_n - v_{n-1})^2,$$

the sum of squares of the differences between consecutive entries of the n-vector v. This is used as a measure of the non-smoothness of the vector v, considered as a time series. Figure 7.5 shows an example.

7.4 Convolution

The *convolution* of an n-vector a and an m-vector b is the $(n+m-1)$-vector denoted $c = a * b$, with entries

$$c_k = \sum_{i+j=k+1} a_i b_j, \quad k = 1, \ldots, n+m-1, \tag{7.2}$$

where the subscript in the sum means that we should sum over all values of i and j in their index ranges $1, \ldots, n$ and $1, \ldots, m$, for which the sum $i + j$ is $k + 1$. For