first coordinate. This would mean that the drawing lives in a hyperplane in $\mathbb{R}^n$, which is undesirable, especially when $n = 2$, where all vertices would be collinear. This is why we omit the first eigenvector u_1.

Observe that for any orthogonal $n \times n$ matrix Q, since

$$\mathrm{tr}(R^\top LR) = \mathrm{tr}(Q^\top R^\top LRQ),$$

the matrix RQ also yields a minimum orthogonal graph drawing. This amounts to applying the rigid motion $Q^\top$ to the rows of R.

In summary, if $\lambda_2 > 0$, an automatic method for drawing a graph in $\mathbb{R}^2$ is this:

1. Compute the two smallest nonzero eigenvalues $\lambda_2 \leq \lambda_3$ of the graph Laplacian L (it is possible that $\lambda_3 = \lambda_2$ if λ_2 is a multiple eigenvalue);

2. Compute two unit eigenvectors u_2, u_3 associated with λ_2 and λ_3, and let $R = [u_2\, u_3]$ be the $m \times 2$ matrix having u_2 and u_3 as columns.

3. Place vertex v_i at the point whose coordinates is the ith row of R, that is, (R_{i1}, R_{i2}).

This method generally gives pleasing results, but beware that there is no guarantee that distinct nodes are assigned distinct images since R can have identical rows. This does not seem to happen often in practice.

21.2 Examples of Graph Drawings

We now give a number of examples using `Matlab`. Some of these are borrowed or adapted from Spielman [163].

Example 1. Consider the graph with four nodes whose adjacency matrix is

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

We use the following program to compute u_2 and u_3:

```
A = [0 1 1 0; 1 0 0 1; 1 0 0 1; 0 1 1 0];
D = diag(sum(A));
L = D - A;
[v, e] = eigs(L);
gplot(A, v(:,[3 2]))
hold on;
gplot(A, v(:,[3 2]),'o')
```