# Notes on Spectral Graph Theory

Mac Radigan

The Laplacian matrix is the product of an oriented incident matrix, say $\mathbb{M}$, and its transpose. Note that this matrix is Hermitian, which is a property we will exploit in the numerical computation of eigenvalues.

$$\mathbb{L} = \mathbb{M}\mathbb{M}^{\intercal} \tag{1}$$

We can determine if all nodes within a submatrix of an oriented incident matrix is connected from its algebraic connectivity. The second eigenvalue of the Laplacian matrix of a subgraph is the algebraic connectivity, and thus the subgraph is connected if the second eigenvalue is greater than zero.

$$\Delta_{\mathbb{L}}(s) = |s\mathbb{I} - \mathbb{L}| \tag{2}$$

$$\underline{\lambda} = \Delta_{\mathbb{L}}(0) \tag{3}$$

$$\lambda_2 > 0 \Leftrightarrow \text{ connected} \tag{4}$$

Of course, we may also compute the transitive closure from the adjacency matrix.

$$\mathbb{C}^* = (\mathbb{A} + \mathbb{I}_{N\times N})^N \tag{5}$$

We may exploit the fact that our Laplacian matrix is Hermitian, and use Kung's Algorithm for $\mathbb{QR}$ factorization (qrDecomp). We may then use the $\mathbb{QR}$ Algorithm (eigW) to compute the eigenvalues of the Laplacian matrix.

To support Kung's Algorithm, we introduce a notation for transvections, $T^N_{i,j}(x)$. The transvection is an $N \times N$ matrix with ones along the diaginal, the value $x$ at row $i$ and column $j$, and zeros elsewhere.

$$\mathbb{T}^N_{i,j}(x) = \mathbb{I}_N + x\mathbb{E}_{i,j} \tag{6}$$

Kung's $\mathbb{QR}$ Algorithm then computes a strictly positive diagonalization matrix, $\mathbb{E}$, and resultant diagonal matrix $\mathbb{D}$. We define $\mathbb{C} = \sqrt{\mathbb{D}}$, and can write the factorization of a matrix $\mathbb{M}_{N\times N}$ as follows.

$$\mathbb{E} = \prod_{i=1}^{N} \prod_{j \neq i} T_{i,j}^{N} \left(\mathbb{M}_{i,j}\right)^{*} \tag{7}$$

$$\mathbb{D} = \left(\mathbb{E}\mathbb{M}\right)^{*} \left(\mathbb{M}\mathbb{E}\right) = \mathbb{E}^{*}\mathbb{M}^{*}\mathbb{M}\mathbb{E} \tag{8}$$

$$\mathbb{C} = \sqrt{\mathbb{D}} \tag{9}$$

$$\mathbb{Q} = \mathbb{A}\mathbb{E}\mathbb{C}^{-1} \tag{10}$$

$$\mathbb{R} = \mathbb{C}\mathbb{E}^{-1} \tag{11}$$

**Algorithm 1** Transvection

```
function TRANSVECTION($\mathbb{M}_{N\times N}, i, j$)
                                        ▷ Initialize an identity matrix
    allocate $\mathbb{T}_{N\times N} \leftarrow 0$
    for $k \leftarrow 1 \ldots N$ do
        $\mathbb{T}_{k,k} \leftarrow 1$
    end for
                                        ▷ Conjugate and copy at i,j
    $\mathbb{T}_{i,j} \leftarrow \mathbb{M}^{*}_{i,j}$
                                        ▷ Return the transvection
    return $\mathbb{T}$
end function
```

The QR Algorithm iteratively computes the eigenvalues of a matrix using a QR decomposition. The algorithm converges when the elements in the subdiagonal approach zero.

**Algorithm 2** Kung's Algorithm

```
function qrDecomp($\mathbb{M}_{N\ timesN}$)
                                        ▷ Initialize an identity matrix
    allocate $\mathbb{E}_{N\times N} \leftarrow 0$
    for $k \leftarrow 1 \dots N$ do
        $\mathbb{E}_{k,k} \leftarrow 1$
    end for
        ▷ Create the diagonalizer by applying conjugate pairs of transvections to remove off-diagonal elements
    for $i \leftarrow 1 \dots N$ do
        for $j \leftarrow 1 \dots N$ do
            if $i \neq j$ then
                $\mathbb{E} = \mathbb{E} \cdot$ Transvection($\mathbb{M}, i, j$)
            end if
        end for
    end for
                        ▷ Apply the diagonalizer to create the diagonalization
    $\mathbb{D} \leftarrow \mathbb{E}^*\mathbb{M}^*\mathbb{M}\mathbb{E}$
    $\mathbb{C} \leftarrow \sqrt{\mathbb{D}}$
                                        ▷ Compute the $\mathbb{Q}$ and $\mathbb{R}$ matrices
    $\mathbb{Q} \leftarrow \mathbb{A}\mathbb{E}\mathbb{C}^{-1}$
    $\mathbb{R} \leftarrow \mathbb{C}\mathbb{E}^{-1}$
                ▷ Return the orthogonal and upper triangular decomposition
    return $\mathbb{Q}, \mathbb{R}$
end function
```

**Algorithm 3** QR Algorithm

```
function EIGW($\mathbb{A}_0^{N \times N}$)
                                                  ▷ Set the initial conditions
    $k \leftarrow 0$
    $\mathrm{Q}_0, \mathbb{R}_0 \leftarrow qrDecomp(\mathrm{A}_0)$
                                ▷ Iterate until the subdiagonal converges to zero
    while $\mathbb{A}_{k_{i,j}} < \epsilon, \forall j = i - 1 \forall i \in 1 \dots N$ do
        $k \leftarrow k + 1$
        $\mathbb{Q}_{k-1}, \mathbb{R}_{k-1} \leftarrow$ QRDECOMP($\mathbb{A}_{k-1}$)
        $\mathbb{A}_k \leftarrow \mathbb{R}_{k-1}\mathbb{Q}_{k-1}$
    end while
                              ▷ Create a vector from the diagonal elements of $\mathbb{A}_k$
    $\underline{\lambda} \leftarrow \{a_{i,j} | \forall a_{i,j} \in \mathbb{A}_k \wedge i = j\}$
                                                        ▷ Return eigenvalues
    return $\underline{\lambda}$
end function
```