Properties of Eigenvalues and Eigenvectors

If λ is an eigenvalue of A, then $1/\lambda$ is an eigenvalue of A^{-1}. The corresponding eigenvectors are the same.

Why is this the case? Consider an invertible matrix A with eigenvalue λ and eigenvector $\vec{x}$. Then, by definition, we know that $A\vec{x} = \lambda\vec{x}$. Now multiply both sides by A^{-1}:

$$A\vec{x} = \lambda\vec{x}$$
$$A^{-1}A\vec{x} = A^{-1}\lambda\vec{x}$$
$$\vec{x} = \lambda A^{-1}\vec{x}$$
$$\frac{1}{\lambda}\vec{x} = A^{-1}\vec{x}$$

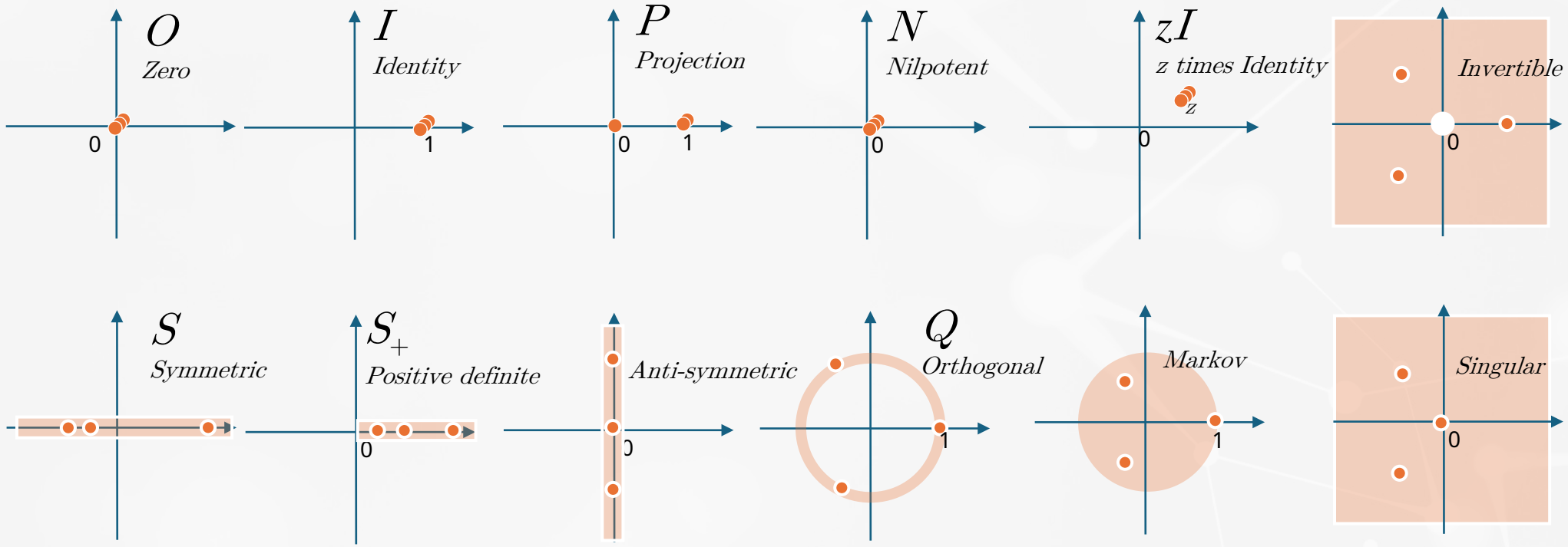
We have just shown that $A^{-1}\vec{x} = 1/\lambda\vec{x}$; this, by definition, shows that $\vec{x}$ is an eigenvector of A^{-1} with eigenvalue $1/\lambda$. This explains the result we saw above.

Properties of Eigenvalues and Eigenvectors

Let A be an $n \times n$ invertible matrix. The following are true:

1. If A is triangular, then the diagonal elements of A are the eigenvalues of A.
2. If λ is an eigenvalue of A with eigenvector $\vec{x}$, then $\frac{1}{\lambda}$ is an eigenvalue of A^{-1} with eigenvector $\vec{x}$.
3. If λ is an eigenvalue of A then λ is an eigenvalue of A^T.
4. The sum of the eigenvalues of A is equal to $\text{tr}(A)$, the trace of A.
5. The product of the eigenvalues of A is the equal to $\det(A)$, the determinant of A.

Map of Eigenvalues for real $n \times n$ square matrices



By Kenji Hiranabe with the kindest help of Prof. Gilbert Strang

Miscellaneous

- Eigenvector centrality of a graph
- QR Decomposition in Hückel Molecular Orbital Approximation

Eigenvector Centrality

- A natural extension of degree centrality is **eigenvector centrality**.

- In-degree centrality awards one centrality point for every link a node receives. But not all vertices are equivalent: some are more relevant than others, and, reasonably, endorsements from important nodes count more.

Eigenvector Centrality

- A natural extension of degree centrality is **eigenvector centrality**.
- In-degree centrality awards one centrality point for every link a node receives. But not all vertices are equivalent: some are more relevant than others, and, reasonably, endorsements from important nodes count more.
- EVC : *A node is important if it is linked to by other important nodes.*

Eigenvector Centrality

Definition of Dominant Eigenvalue and Dominant Eigenvector

Let $\lambda_1, \lambda_2, \ldots,$ and λ_n be the eigenvalues of an $n \times n$ matrix A. λ_1 is called the **dominant eigenvalue** of A if

$$|\lambda_1| > |\lambda_i|, \quad i = 2, \ldots, n.$$

The eigenvectors corresponding to λ_1 are called **dominant eigenvectors** of A.

Eigenvector Centrality

The Power Method

Like the Jacobi and Gauss-Seidel methods, the power method for approximating eigenvalues is iterative. First we assume that the matrix A has a dominant eigenvalue with corresponding dominant eigenvectors. Then we choose an initial approximation $\mathbf{x}_0$ of one of the dominant eigen*vectors* of A. This initial approximation must be a *nonzero* vector in R^n. Finally we form the sequence given by

$$\begin{aligned}
\mathbf{x}_1 &= A\mathbf{x}_0 \\
\mathbf{x}_2 = A\mathbf{x}_1 &= A(A\mathbf{x}_0) = A^2\mathbf{x}_0 \\
\mathbf{x}_3 = A\mathbf{x}_2 &= A(A^2\mathbf{x}_0) = A^3\mathbf{x}_0 \\
&\vdots \\
\mathbf{x}_k = A\mathbf{x}_{k-1} &= A(A^{k-1}\mathbf{x}_0) = A^k\mathbf{x}_0.
\end{aligned}$$

For large powers of k, and by properly scaling this sequence, we will see that we obtain a good approximation of the dominant eigenvector of A.

Why does it work?

If A has a set of **linearly independent eigenvectors**, we can write any vector x_0 as a combination of them:

$$x_0 = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$$

where $v_1, v_2, \ldots, v_n$ are the eigenvectors of A with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$.

Now, multiplying by A repeatedly:

$$Ax_0 = c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \cdots + c_n \lambda_n v_n$$

$$A^2 x_0 = c_1 \lambda_1^2 v_1 + c_2 \lambda_2^2 v_2 + \cdots + c_n \lambda_n^2 v_n$$

$$A^k x_0 = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n$$

As $k \to \infty$, if $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, the term $c_1 \lambda_1^k v_1$ dominates because λ_1^k grows the fastest.

$$x_k \approx c_1 \lambda_1^k v_1$$

Thus, after enough iterations, the vector x_k aligns with the eigenvector v_1, and the eigenvalue can be estimated as:

$$\lambda_k = \frac{x_k^T A x_k}{x_k^T x_k}$$

This is the **Rayleigh quotient**, which gives a good approximation of λ_1.

If A has eigenvalues λ_1 and λ_2 such that:

$$|\lambda_1| \approx |\lambda_2|$$

then the convergence of Power Iteration slows down. Why? Because the **rate at which the smaller eigenvector decays** depends on the **ratio**:

$$r = \left| \frac{\lambda_2}{\lambda_1} \right|$$

- If r is **small** (e.g., 0.1), the method works well because λ_2^k shrinks fast.
- If $r \approx 1$, the second eigenvector **remains significant** for a long time, causing **slow convergence**.

Mathematical Explanation

Assume the initial vector x_0 is a combination of eigenvectors:

$$x_0 = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$$

After k iterations:

$$x_k = A^k x_0 = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \dots$$

Dividing by λ_1^k:

$$x_k = c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \dots$$

If $|\lambda_2| \approx |\lambda_1|$, the term $\left(\frac{\lambda_2}{\lambda_1}\right)^k$ **does not decay quickly**, so v_2 remains **significant**, making convergence very slow.

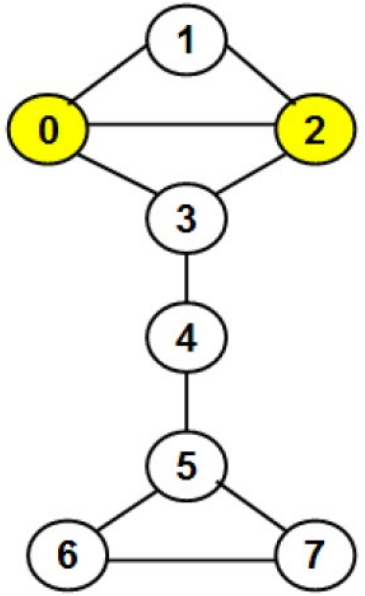
Eigenvector Centrality

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Iteration		*Approximation*
$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ -4 \end{bmatrix}$	→	$-4\begin{bmatrix} 2.50 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -10 \\ -4 \end{bmatrix} = \begin{bmatrix} 28 \\ 10 \end{bmatrix}$	→	$10\begin{bmatrix} 2.80 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_3 = A\mathbf{x}_2 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 28 \\ 10 \end{bmatrix} = \begin{bmatrix} -64 \\ -22 \end{bmatrix}$	→	$-22\begin{bmatrix} 2.91 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_4 = A\mathbf{x}_3 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -64 \\ -22 \end{bmatrix} = \begin{bmatrix} 136 \\ 46 \end{bmatrix}$	→	$46\begin{bmatrix} 2.96 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_5 = A\mathbf{x}_4 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 136 \\ 46 \end{bmatrix} = \begin{bmatrix} -280 \\ -94 \end{bmatrix}$	→	$-94\begin{bmatrix} 2.98 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_6 = A\mathbf{x}_5 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -280 \\ -94 \end{bmatrix} = \begin{bmatrix} 568 \\ 190 \end{bmatrix}$	→	$190\begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}$

Eigenvector Centrality



Iteration 1

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 3 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \end{bmatrix} \equiv \begin{bmatrix} 0.416 \\ 0.277 \\ 0.416 \\ 0.416 \\ 0.277 \\ 0.416 \\ 0.277 \\ 0.277 \end{bmatrix}$$

Normalized Value = 7.21

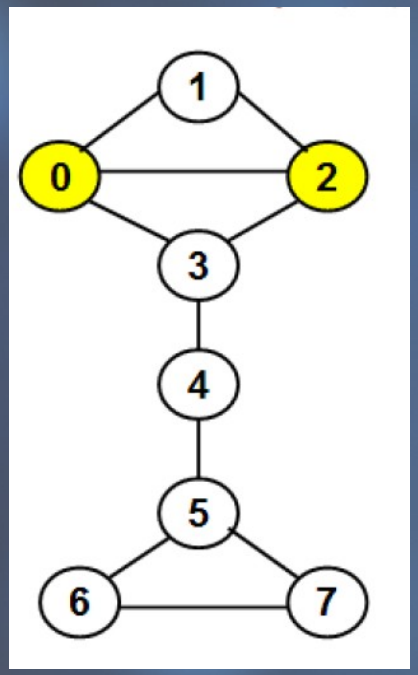
Iteration 2

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0.416 \\ 0.277 \\ 0.416 \\ 0.416 \\ 0.277 \\ 0.416 \\ 0.277 \\ 0.277 \end{bmatrix} = \begin{bmatrix} 1.109 \\ 0.832 \\ 1.109 \\ 1.109 \\ 0.832 \\ 0.831 \\ 0.693 \\ 0.693 \end{bmatrix} \equiv \begin{bmatrix} 0.428 \\ 0.321 \\ 0.428 \\ 0.428 \\ 0.321 \\ 0.321 \\ 0.268 \\ 0.268 \end{bmatrix}$$

Normalized Value = 2.59

Dr. Natarajan Meghanathan

Eigenvector Centrality



Iteration 3

$$\begin{bmatrix} 0&1&1&1&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&1&0&1&0&0&0&0 \\ 1&0&1&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0 \\ 0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&1&0&1 \end{bmatrix} \times \begin{bmatrix} 0.428 \\ 0.321 \\ 0.428 \\ 0.428 \\ 0.321 \\ 0.321 \\ 0.268 \end{bmatrix} = \begin{bmatrix} 1.177 \\ 0.856 \\ 1.284 \\ 1.177 \\ 0.749 \\ 0.857 \\ 0.589 \\ 0.589 \end{bmatrix} \equiv \begin{bmatrix} 0.441 \\ 0.321 \\ 0.481 \\ 0.441 \\ 0.281 \\ 0.321 \\ 0.221 \\ 0.221 \end{bmatrix}$$

= 2.67

Note that we typically stop when the EigenVector values converge.
For exam purposes, we will Stop when the Normalized value converges.

$$\begin{bmatrix} 0&0&0&0&0&1&1&0 \end{bmatrix} \begin{bmatrix} 0.221 \end{bmatrix} \quad \begin{bmatrix} 1.243 \\ 0.922 \\ 1.203 \\ 1.203 \\ 0.762 \\ 0.723 \\ 0.542 \\ 0.542 \end{bmatrix} \equiv \begin{bmatrix} 0.471 \\ 0.349 \\ 0.456 \\ 0.456 \\ 0.289 \\ 0.274 \\ 0.205 \\ 0.205 \end{bmatrix}$$

Normalized Value = 2.64

(i, j)

Normalized Value = 2.64

After 7 iterations

Vertex ID	0	1	2	3	4	5	6	7
Principal Eigenvector	0.489	0.364	0.489	0.467	0.264	0.232	0.155	0.155

QR Decomposition in Hückel Molecular Orbital Approximation

The **Hückel Molecular Orbital (HMO) theory** is used to approximate π-electron energies and molecular orbitals in conjugated hydrocarbons. It simplifies quantum chemical calculations by considering only π-electrons and assuming **nearest-neighbor interactions** in a molecule's **adjacency matrix** representation.

QR decomposition is useful in **diagonalizing** the Hückel **Hamiltonian matrix** to find molecular orbital energies.

QR Decomposition in Hückel Molecular Orbital Approximation

The **Hückel Molecular Orbital (HMO) theory** is used to approximate π-electron energies and molecular orbitals in conjugated hydrocarbons. It simplifies quantum chemical calculations by considering only π-electrons and assuming **nearest-neighbor interactions** in a molecule's **adjacency matrix** representation.

QR decomposition is useful in **diagonalizing** the Hückel **Hamiltonian matrix** to find molecular orbital energies.

For a linear butadine molecule C_4H_6, the **Hückel Hamiltonian matrix** $\boldsymbol{H}$ (in the basis of π atomic orbitals) is:

$$\begin{pmatrix} \alpha & \beta & 0 & 0 \\ \beta & \alpha & \beta & 0 \\ 0 & \beta & \alpha & \beta \\ 0 & 0 & \beta & \alpha \end{pmatrix}$$

where:

- α is the Coulomb integral (energy of an isolated p-orbital),
- β is the resonance integral (interaction energy between adjacent orbitals).

QR Decomposition in Hückel Molecular Orbital Approximation

QR decomposition is useful in **diagonalizing** the Hückel **Hamiltonian matrix** to find molecular orbital energies.

For a linear butadine molecule C_4H_6, the **Hückel Hamiltonian matrix H** (in the basis of π atomic orbitals) is:

$$\begin{pmatrix} \alpha & \beta & 0 & 0 \\ \beta & \alpha & \beta & 0 \\ 0 & \beta & \alpha & \beta \\ 0 & 0 & \beta & \alpha \end{pmatrix}$$

The molecular orbital energies are obtained by solving the eigenvalue problem:

$$Hc = Ec$$

Instead of computing eigenvalues directly, we iteratively diagonalize H using the QR algorithm.

$$H = QR$$

Iterative QR Algorithm for Molecular Orbitals energies

1.Initialize $H_0 = H$.
2.Compute QR decomposition: $H_k = Q_k R_k$
3.Update matrix: $\boldsymbol{H_{k+1} = R_K Q_k}$
4.Repeat until H_k converges to a diagonal form.

The QR algorithm finds eigenvalues because **it iteratively transforms the matrix into an upper triangular form, preserving eigenvalues at every step.** This avoids the instability of solving polynomials and provides a robust way to compute eigenvalues numerically.

Step-by-Step Explanation

Given a matrix A, the QR algorithm follows these steps:

1. QR Decomposition: Compute the decomposition
$A_k = Q_k R_k$ where Q_k is an orthogonal matrix and R_k is an upper triangular matrix.

2. Matrix Update: Construct the next iteration matrix as

$$A_{k+1} = R_k Q_k.$$

This transformation preserves eigenvalues because:

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k$$

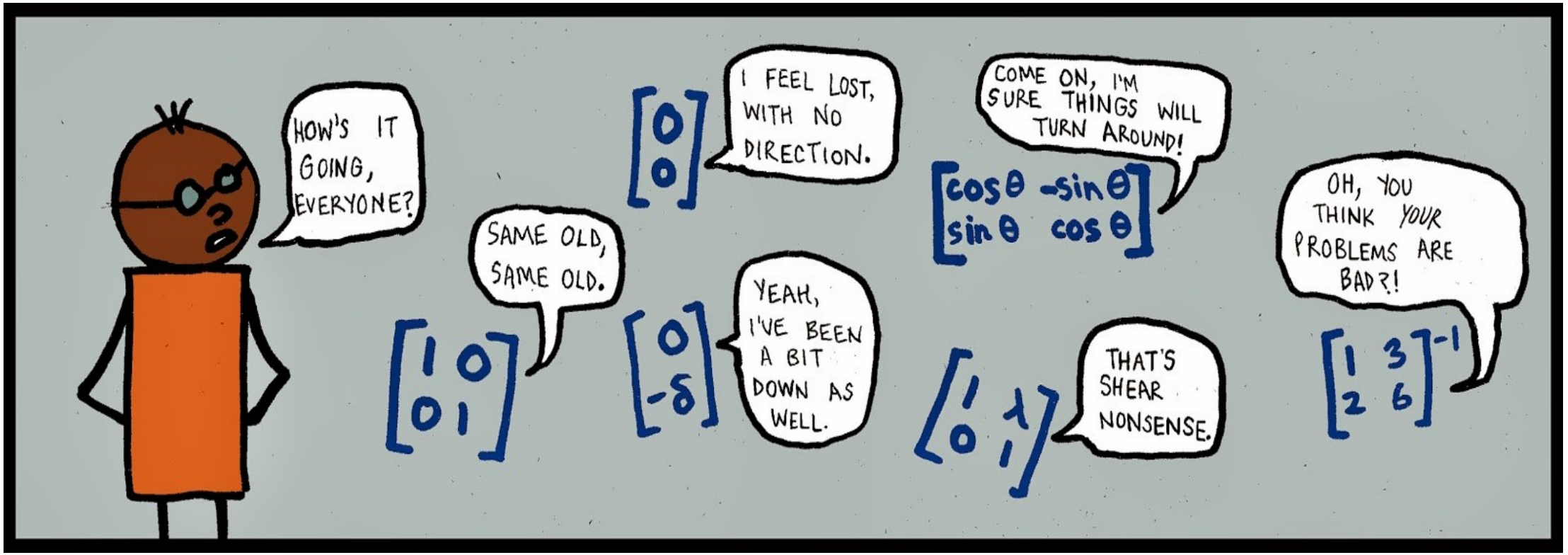
Since orthogonal transformations preserve eigenvalues, all A_k have the same eigenvalues as A.
So each step is an orthogonal similarity transformation.

3. Convergence: After several iterations, A_k converges to an upper triangular matrix, where the eigenvalues appear on the diagonal.

Iterative QR Algorithm for Molecular Orbitals energies

1. **Initialize** $H_0 = H$.
2. **Compute QR decomposition:** $H_k = Q_k R_k$
3. **Update matrix:** $\boldsymbol{H_{k+1} = R_K Q_k}$
4. **Repeat** until H_k converges to a diagonal form.

QR algorithm involves an orthogonal similarity transformation.
This preserves the eigenvalue


HOW'S IT GOING, EVERYONE?
SAME OLD, SAME OLD.
I FEEL LOST, WITH NO DIRECTION.
YEAH, I'VE BEEN A BIT DOWN AS WELL.
COME ON, I'M SURE THINGS WILL TURN AROUND!
THAT'S SHEAR NONSENSE.
OH, YOU THINK YOUR PROBLEMS ARE BAD?!