Solving Mazes: A New Approach Based on Spectral Graph Theory

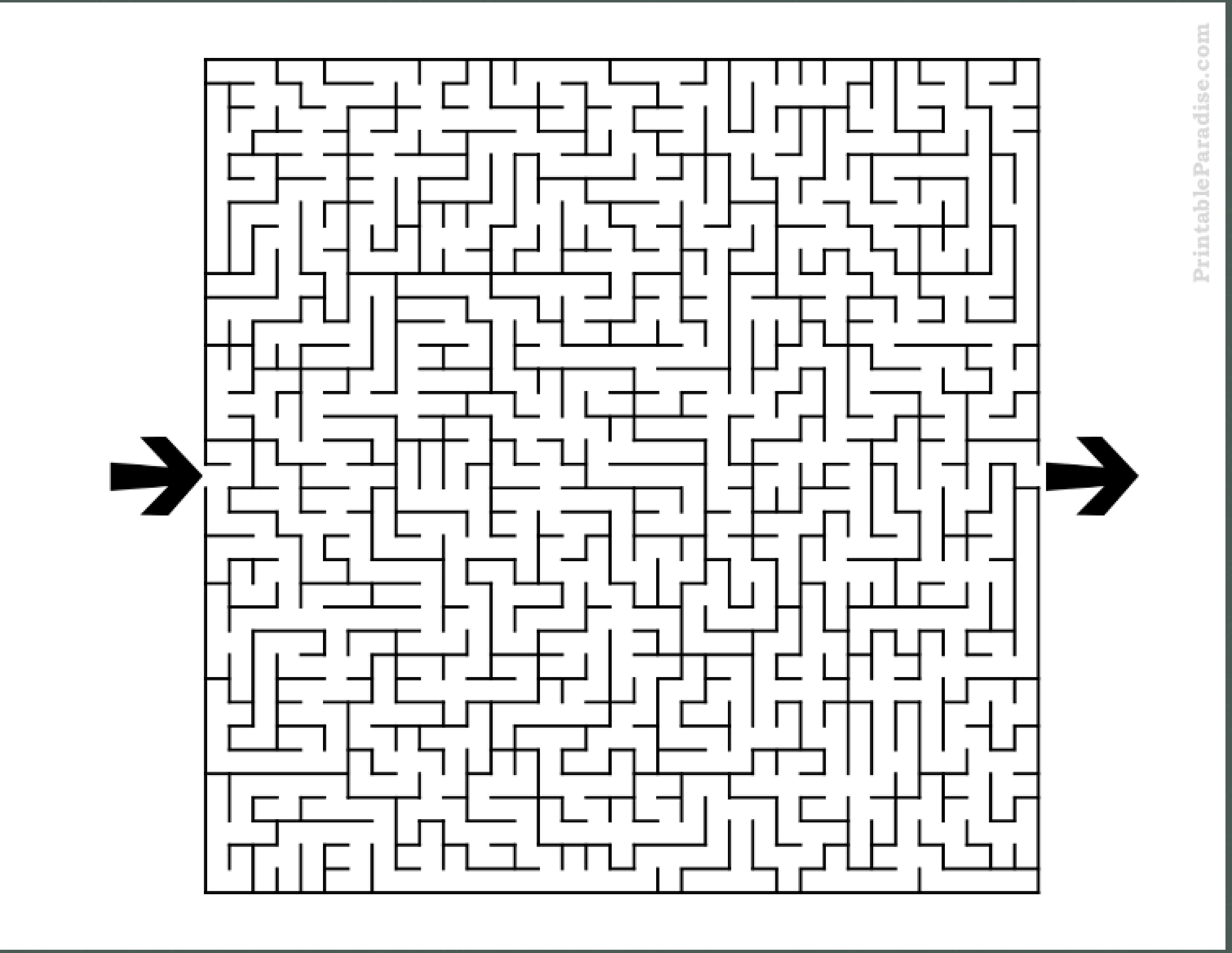
Article

Solving Mazes: A New Approach Based on Spectral Graph Theory

Marta Martín-Nieto [1,2], Damián Castaño [1,3], Sergio Horta Muñoz [1,4] and David Ruiz [1,5,*]

Maze Puzzles

- Entrance & Exit
- Could be Dead ends
- Could loop around
- Could have more than one solution
- Can you figure this out?



Graph Theory

- Graph, G
  - Set of vertices, V
  - $|V| = n$
  - Set of edges, E
- Adjacency Matrix, A
  - Entries of A denote the edge between node i and j

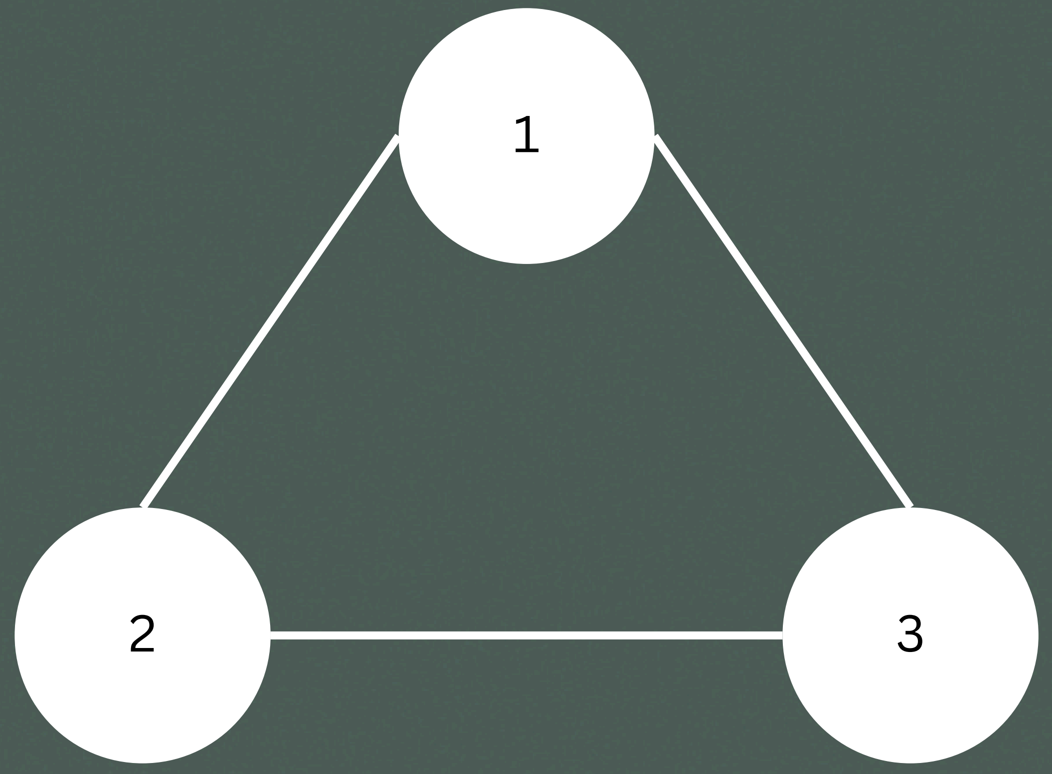
$$G(V, E)$$

$$\Updownarrow$$

$$A \in \mathcal{M}^{n \times n}$$

Graph Theory

- Assume that undirected graph
  - A is symmetric
- Assume entries are either {0,1}
  - 1 - exists an edge
  - 0 - otherwise



$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Graph Theory

- Graph Laplacian, L, gives insight on the connectivity over a graph
- Calculated as the difference between D, diagonal matrix with degrees of A, and A itself

$$L \sim Laplacian(A)$$

$$L = D - A$$

$$L_{i,j} = \begin{cases} d_i, & i = j \\ -a_{i,j}, & i \neq j \end{cases}$$

$$d_i = \sum_{j}^{n} a_{i,j}$$

Graph Theory

$$A \qquad\qquad L$$

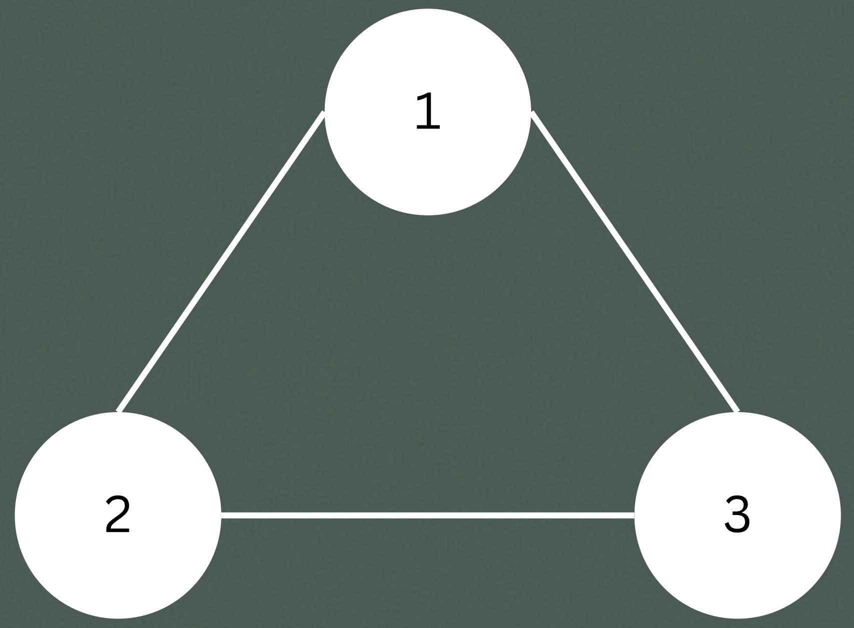
$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \Longrightarrow \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Graph Theory

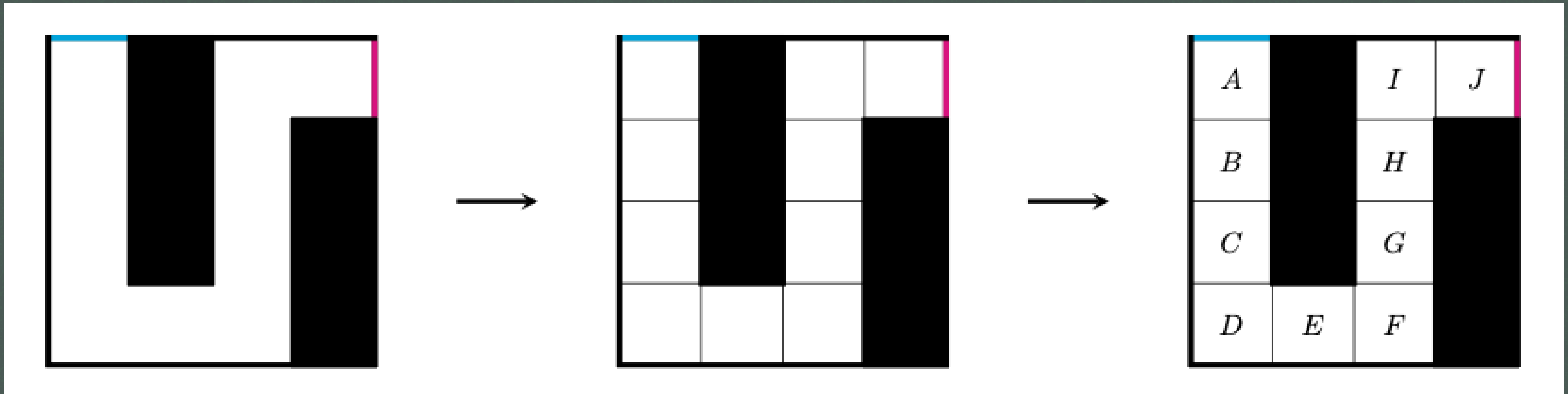
- The eigenvalues of L gives the following insight:
  - number of 0 eigenvalues number of connected components
  - If the second smallest eigenvalue (Fiedler Eigenvalue) is positive then G is connected

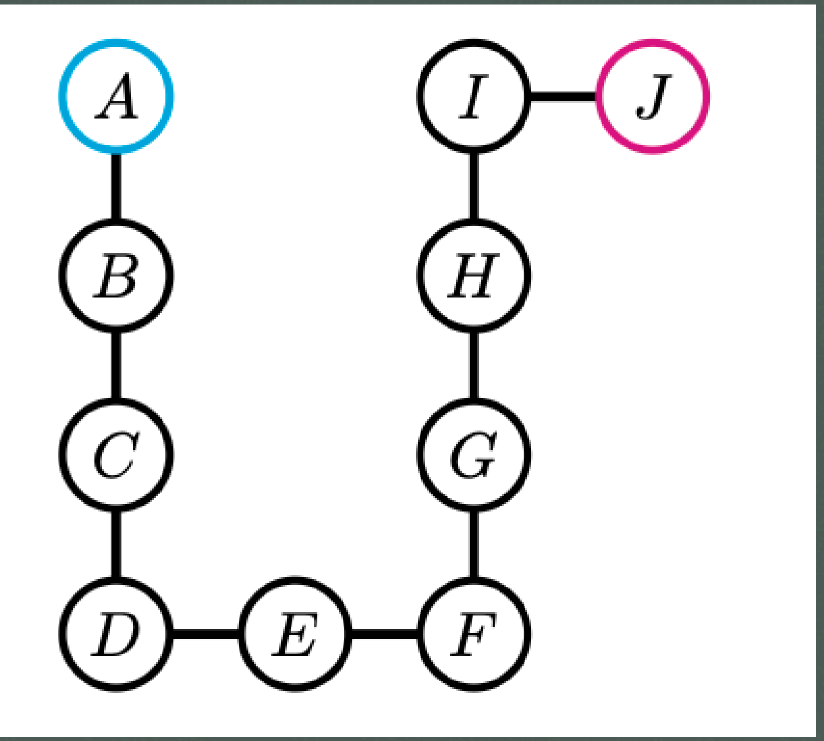
$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

$$\lambda = [0, 3, 3]$$



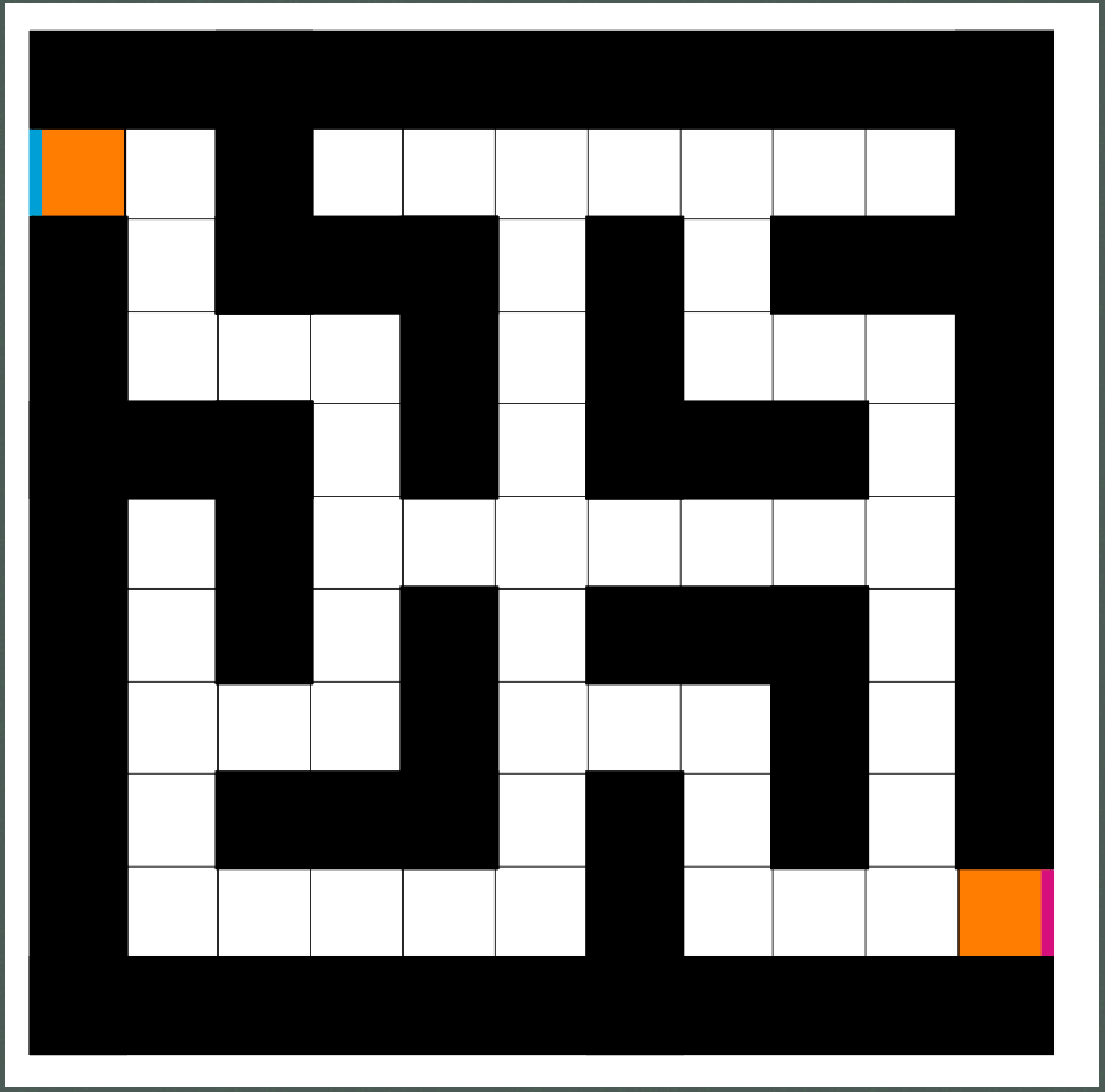
Maze Application





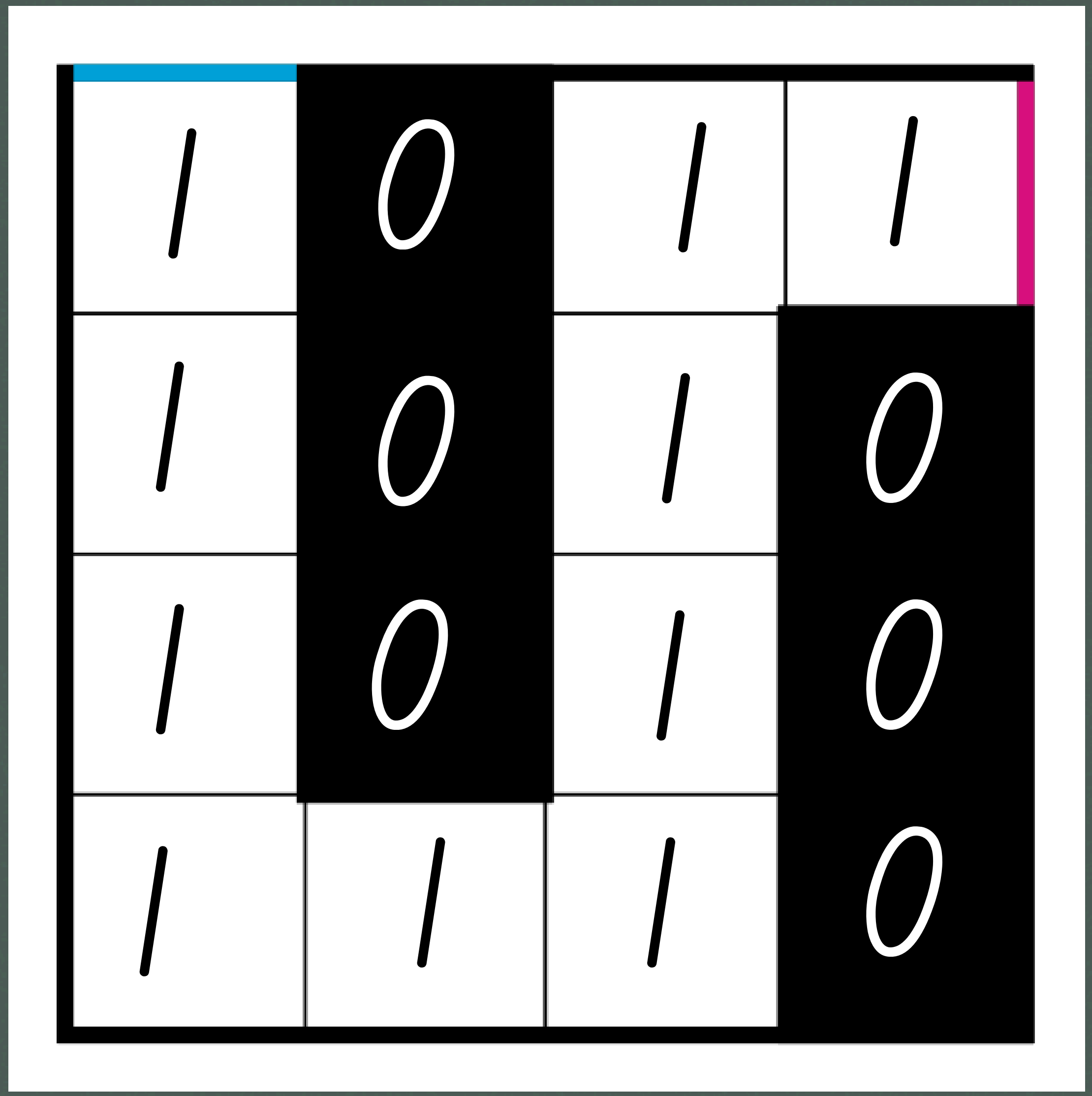
Maze Application

- Implement a new variable, $\varphi \in \mathbb{R}^n$ which is either {0,1}
  - 1 - if φ_i is a path
  - 0 - if φ_i is a wall



Maze Application

Discretizing a Maze



$$\Longrightarrow$$

$$A(\varphi) \in \mathcal{M}^{16*16}$$

$$a_{ij} = \begin{cases} \varphi_i \varphi_j & i \text{ is adjacent to } j \\ 0 & \text{otherwise} \end{cases}$$

Maze Application

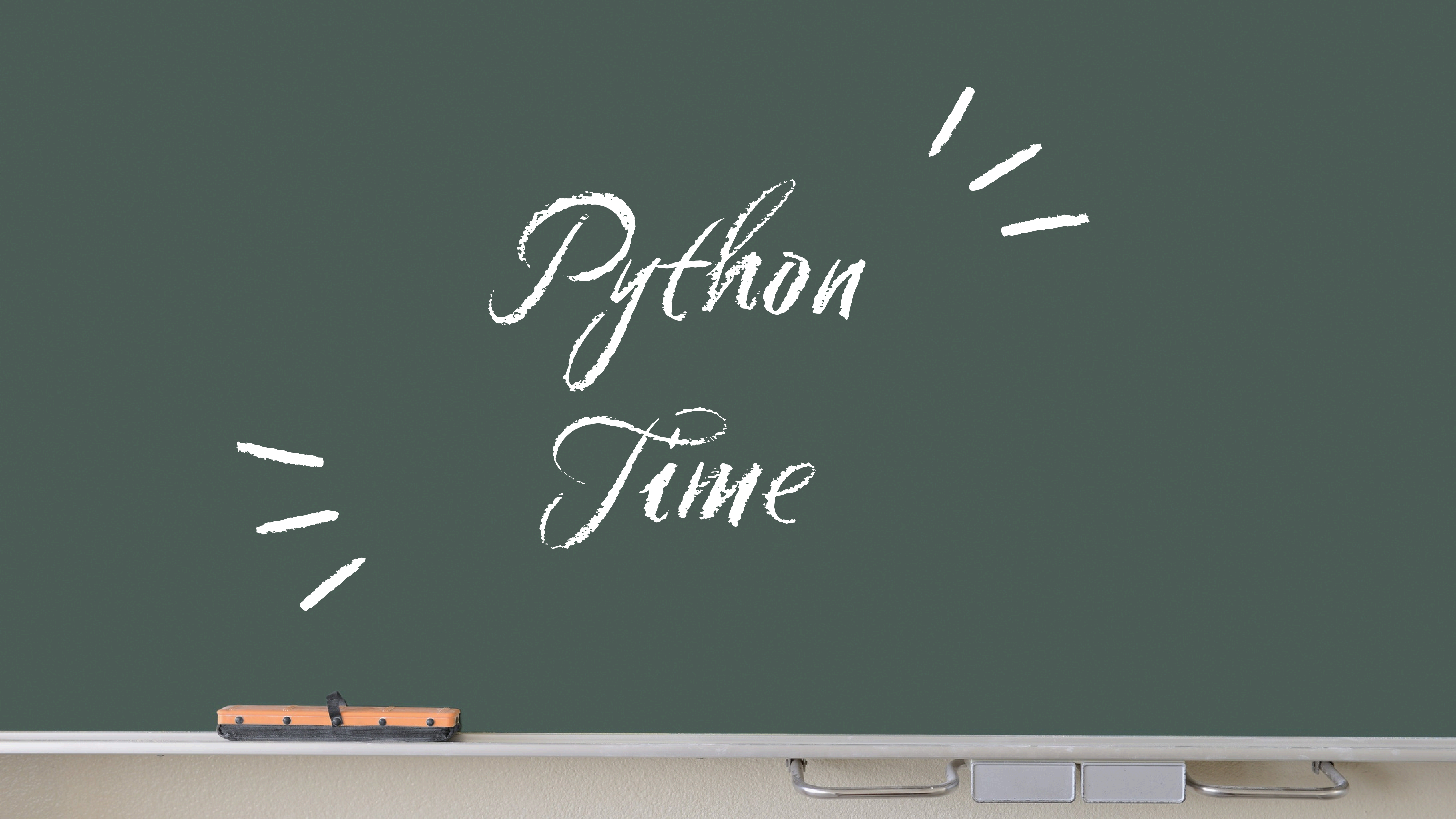
- To find the solution to our maze we optimize the following:

$$\min \mathbf{1}^T \varphi$$

with constraints:

$$(L(\varphi) - \lambda_k M(\varphi))\Phi_k = 0$$
$$\Phi_k^T M \Phi_k = 1$$
$$\lambda_2 > 0$$

$$\Phi \sim \text{eigenvectors}$$
$$\lambda \sim \text{eigenvalues}$$
$$M \sim diag(\varphi)$$


Python
Time

Topology Optimization

- We extend the optimization to a topological extent
- We transform, $\varphi \in \{0,1\}$, into a density $\rho \in [0,1]$

$$\min \mathbf{1}^T \bar{\rho}$$

with constraints:

$$(L(\bar{\rho}) - \lambda_k M(\bar{\rho}))\Phi_k = 0$$
$$\Phi_k^T M \Phi_k = 1$$
$$\lambda_2 > 0$$

$\Phi \sim$ eigenvectors

$\lambda \sim$ eigenvalues

$M \sim diag(\varphi)$

Topology Optimization

- We have the following modifications

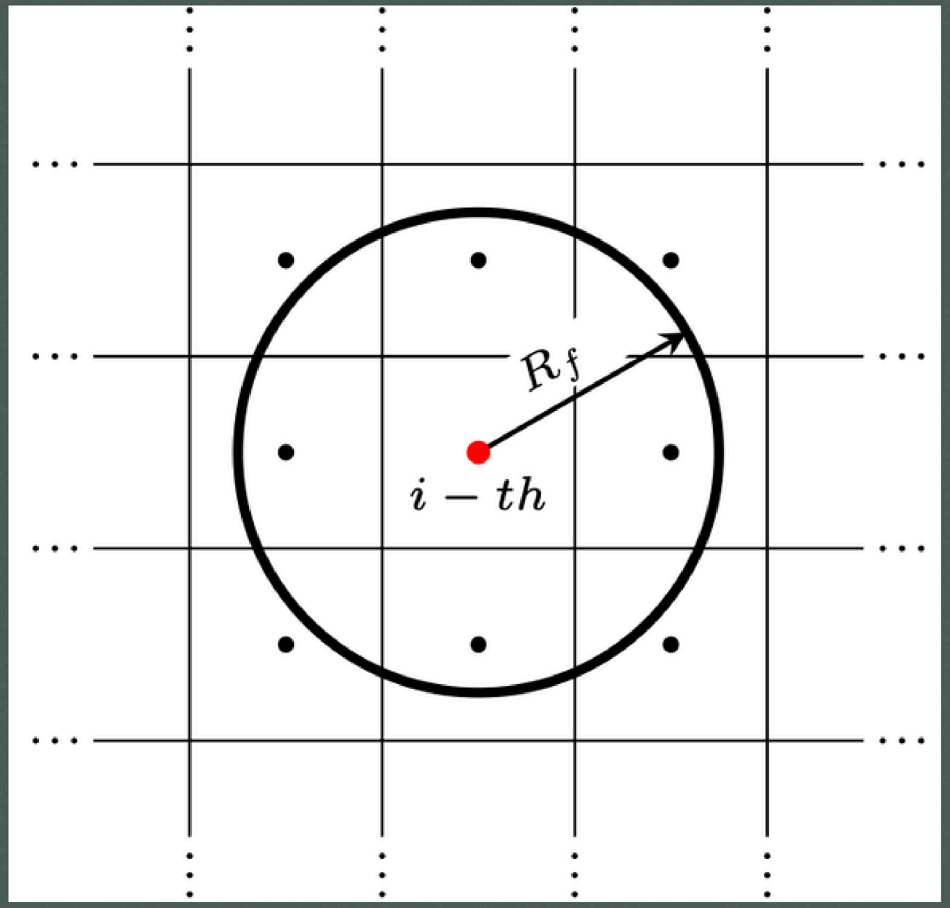
Heavyside Projection

$$\bar{\rho} = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}$$

Conic Filter

$$\tilde{\rho} = \frac{\sum_i^{n_e} d_e(x_i)\rho_i}{\sum_i^{n_e} d_e(x_i)}$$

$$d_e(x_i) = \max\{R_f - ||x_i - x_e||, 0\}$$



$$w_{ij} = (\rho_i\rho_j)^p + w_{\min}$$

Topology Optimization

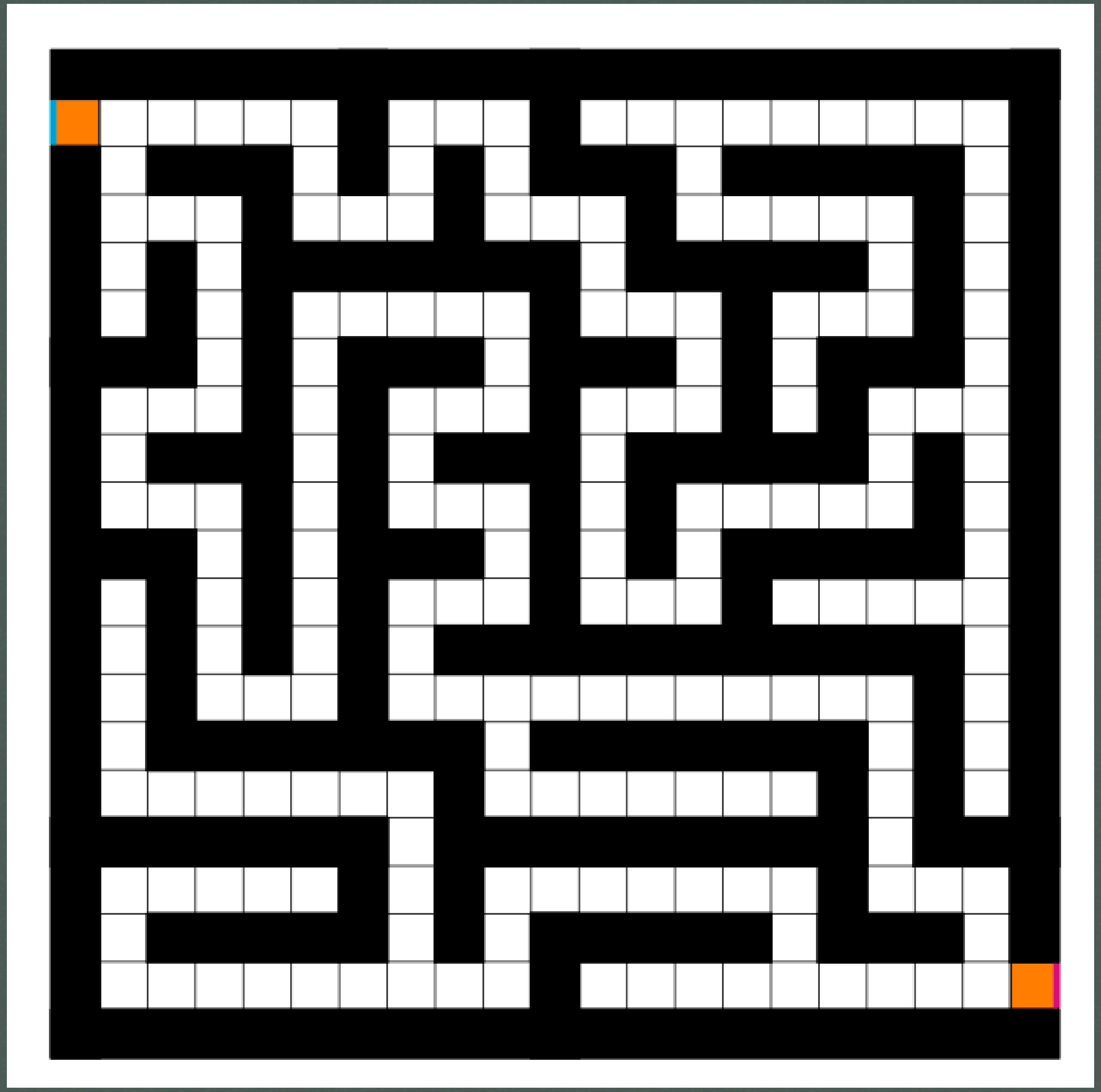
Set Geometry of the maze and mesh
Set Optimization parameters
Define initialization $\boldsymbol{\rho}$
Set Optimization method tolerance *tol*
While $e > tol$
- **Filtering and projection** $\boldsymbol{\rho} \rightarrow \tilde{\boldsymbol{\rho}} \rightarrow \bar{\boldsymbol{\rho}}$
- **Assembly** of global matrices $\mathbf{L}(\bar{\boldsymbol{\rho}})$ and $\mathbf{M}(\bar{\boldsymbol{\rho}})$
- **Compute** λ_k and $\boldsymbol{\Phi}_k$
- **Compute** objective function $c = \mathbf{1}^T \bar{\boldsymbol{\rho}}$
- **Calculate** derivatives
- **Update** variables with MMA $(\boldsymbol{\rho}^*)$
- **Define convergence variable** $e = ||\boldsymbol{\rho}^* - \boldsymbol{\rho}||$

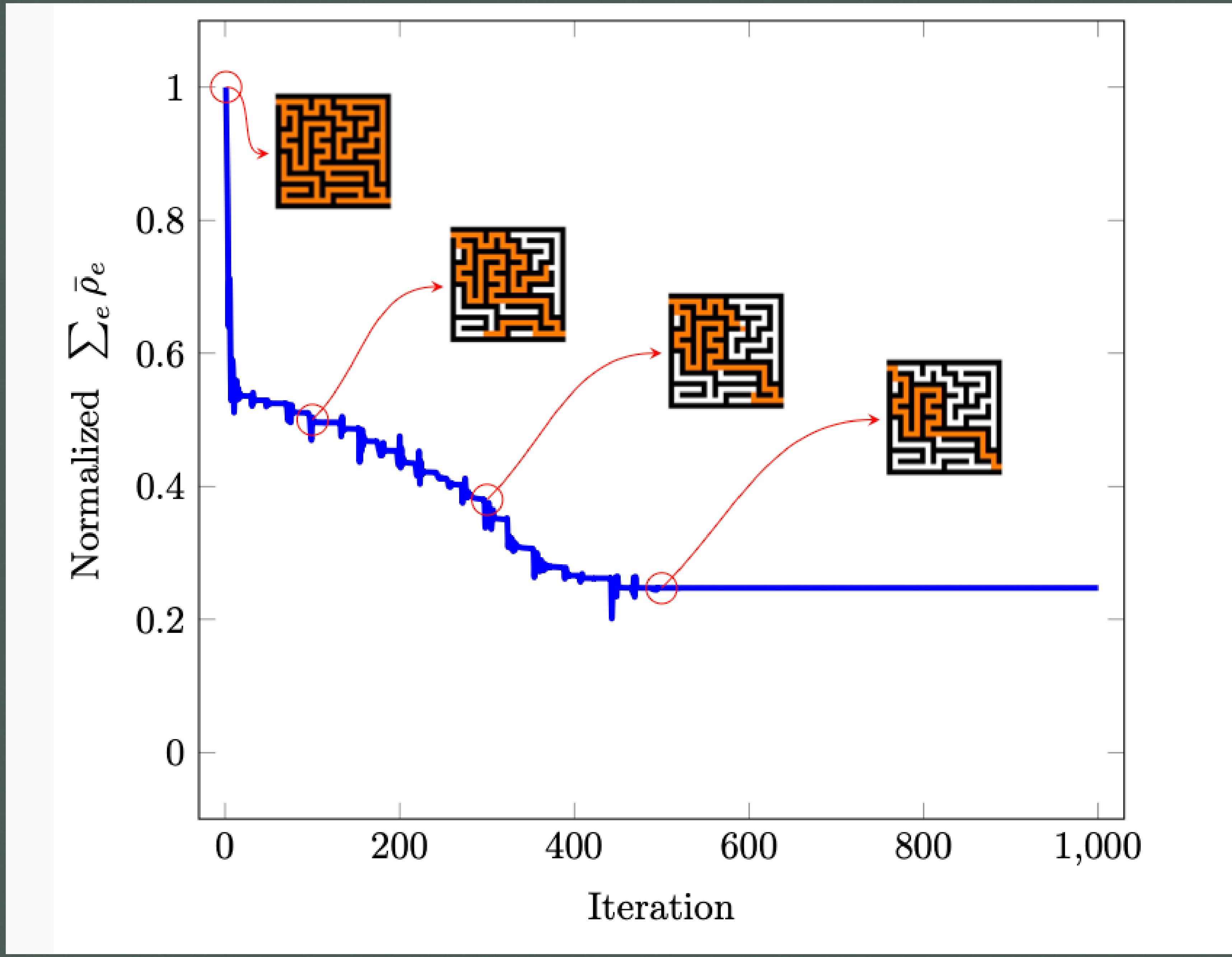
end

$$(\beta, \eta, R_f)$$

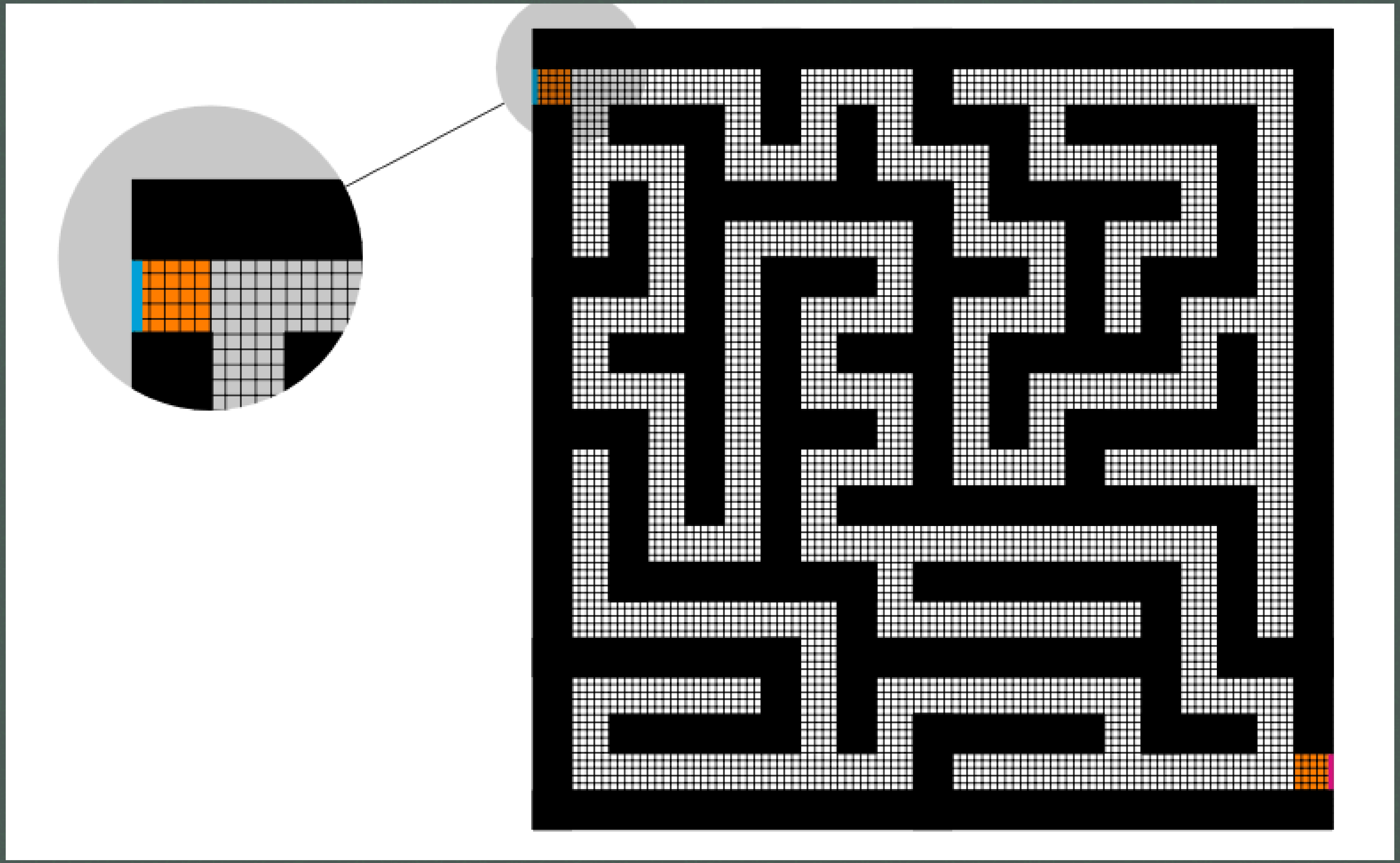
$$\boldsymbol{\rho} \in \mathbb{R}^n$$

Maze Application

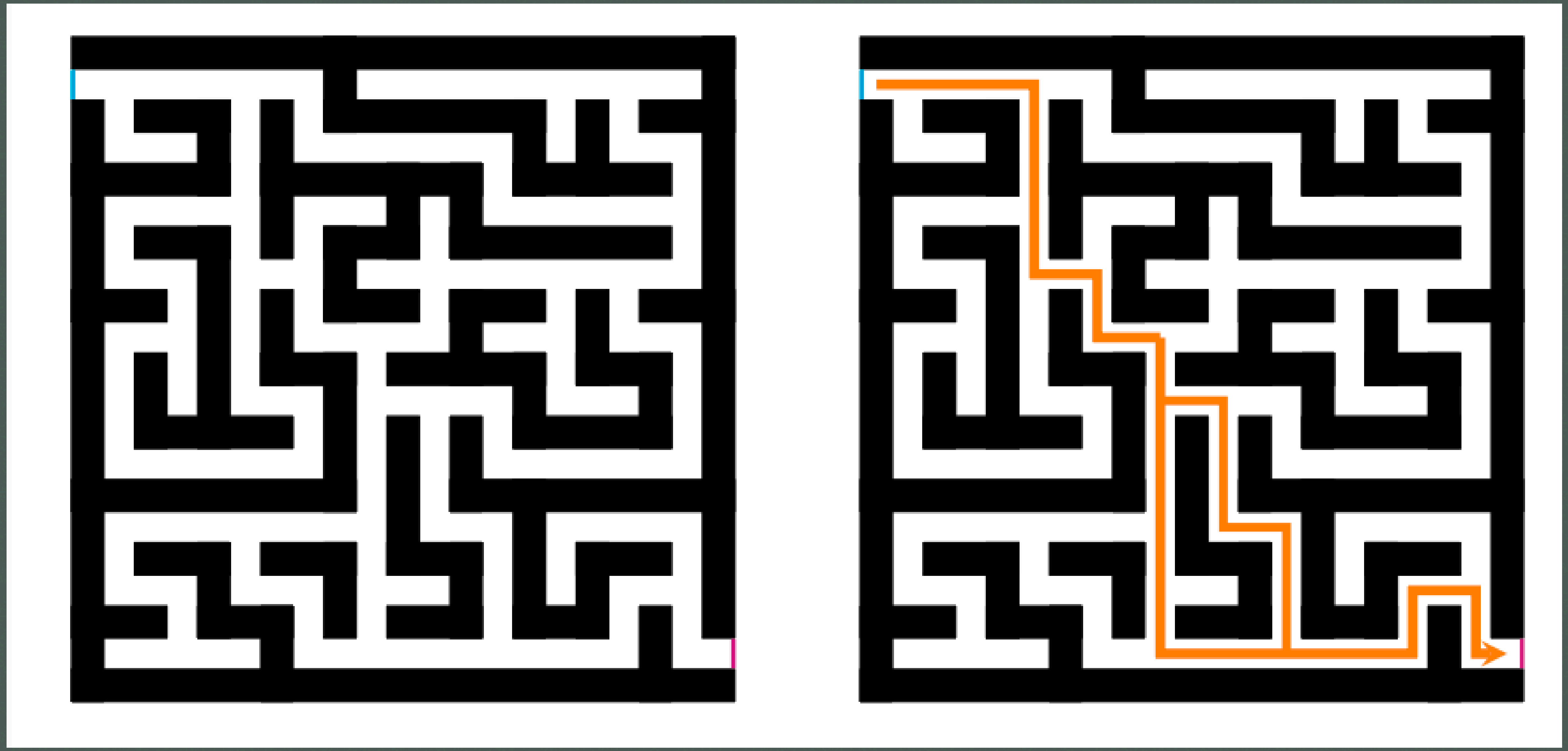




Maze Application



Maze Application




thank you!