In [1]:
```
import numpy as np
import math
```

5.b)

In [2]:
```
def matrix (N,c):
    A = np.zeros((N+1, N+1))
    for i in range (1,N):
        A[i][i] = -c
        A[i][i+1] = 1
        A[i+1][i] = 1
        A[0][0] = 1
        # BC at the end of the hallway (last eqn of matrix)
        A[N][N] = 1
        A[N][N-1] = -1
        # BC at the start of the hallway (forst eqn of matrix)
        A[1][0] = 1
        A[0][1] = -1

    return A
```

5.c)

In [3]:
```
alpha_0 = -0.005
gamma = 1/3600
kappa = 0.05
x = 20
N = 6
c = math.pow((x/N),2)*gamma/kappa +2

A1 = matrix(N,c)
F = np.full(N+1, 0, dtype=np.float64)
F[1] = alpha_0
print('A1 = ', A1)
#print(F)
sol_A1 = np.linalg.solve(A1, F)
print('solution to A1 = ', sol_A1)
```

```
A1 =  [[ 1.         -1.          0.          0.          0.          0.
   0.        ]
 [ 1.         -2.0617284   1.          0.          0.          0.
   0.        ]
 [ 0.          1.         -2.0617284   1.          0.          0.
   0.        ]
 [ 0.          0.          1.         -2.0617284   1.          0.
   0.        ]
 [ 0.          0.          0.          1.         -2.0617284   1.
   0.        ]
 [ 0.          0.          0.          0.          1.         -2.0617284
   1.        ]
 [ 0.          0.          0.          0.          0.         -1.
   1.        ]]
solution to A1 =  [0.02149364 0.02149364 0.0178204  0.0152472  0.01361517 0.
01282359
 0.01282359]
```

5.d)

In [4]:
```python
cond = np.linalg.cond(A1)
print('A1 condition number = ', cond)
```

```
A1 condition number =  89.19782488209792
```

5.e)

In [5]:
```python
c2 = math.pow((x/N),2)*0/kappa + 2
A2 = matrix(N,c2)
cond_A2 = np.linalg.cond(A2)
print('A2 condition number = ', cond_A2)
sol_A2 = np.linalg.solve(A2,F)
print(sol_A2)
```

```
A2 condition number =  1.760929342622134e+17
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
Cell In[5], line 5
      3 cond_A2 = np.linalg.cond(A2)
      4 print('A2 condition number = ', cond_A2)
----> 5 sol_A2 = np.linalg.solve(A2,F)
      6 print(sol_A2)

File ~/miniforge3/envs/numeric_2024/lib/python3.12/site-packages/numpy/linal
g/linalg.py:409, in solve(a, b)
    407 signature = 'DD->D' if isComplexType(t) else 'dd->d'
    408 extobj = get_linalg_error_extobj(_raise_linalgerror_singular)
--> 409 r = gufunc(a, b, signature=signature, extobj=extobj)
    411 return wrap(r.astype(result_t, copy=False))

File ~/miniforge3/envs/numeric_2024/lib/python3.12/site-packages/numpy/linal
g/linalg.py:112, in _raise_linalgerror_singular(err, flag)
    111 def _raise_linalgerror_singular(err, flag):
--> 112     raise LinAlgError("Singular matrix")

LinAlgError: Singular matrix
```

The above results in an error: Singular Matrix. The matrix having no single solution for the system of linear equations, and the condition number has gotten really big. Physically, this is because gamma is the rate at which the smoke sticks to the walls so if gamma is zero, and we are in the steady state, the smoke isnt dissipating anywhere. Nothing is happening in the system.

5.f)

In [ ]:
```
alpha = np.full(N+1, 0, dtype=np.float64)
s = np.linalg.solve(A2, alpha)
```

This also results in the error: Singular Matrix, so there is also no singluar solution. We have the same conditions as before and there is no origin of the smoke either. So, again nothing is happening in the system.

In [ ]:
```
```