**Crank-Nicholson with Black-Scholes**

The Black-Scholes PDE has the following form

where is defined on the domain . The boundary conditions and initial condition are as followings

Consider a transform of variable

therefore

with the boundary conditions and initial condition:

Call option:

Put option:

Based on the C-N scheme, the derivatives can be discretized as

Crank-Nicholson has 2nd order accuracy in time and space.

Plug into the original PDE

Define following auxiliary variables as

In the stencil form

In the matrix form w/ boundary conditions

Move everything known to LHS and then solve for backwardly

In the python code, the looks like following

