american_put_option

November 9, 2018

0.1 American Put Option Pricing

0.1.1 Preparation

In [1]:
```
import numpy as np
import math
import matplotlib
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib notebook

# Parameters set
T = 1.0
K = 100.0
r = 0.03
N = 100
sigma = 0.3
S0 = 100

# Boundary
b = 2 * sigma * np.sqrt(T) + np.log(K/S0) - (r - 0.5*sigma**2) * T
a = - 3 * sigma * np.sqrt(T) + np.log(K/S0) - (r - 0.5*sigma**2) * T
dx = (b - a) / N

M = math.ceil(T / (0.5 * dx**2 / sigma**2))
dt = T / M
```

0.1.2 Main Calculation

In [2]:
```
# Terminal Condition
u = np.zeros([N+1, M+1])
x_terminal = np.arange(N+1)/N * (b - a) + a
u[:,M] = np.maximum(K - S0 * np.exp(x_terminal), 0)

# Boundary Condition
t_boundary = np.arange(M+1)/M * (T - 0) + 0
u[N,:] = 0                    # Upper condition
u[0,:] = K - S0 * np.exp(a) # Lower condition
```

```
        # Prepare coefficients in Backward Calculation
        alpha_rdt   = 1 - (sigma**2) * dt / (dx**2) - r * dt
        alpha_plus  = 0.5 * sigma**2 * dt / (dx**2) + 0.5 * dt / dx * (r - 0.5 * sigma**2)
        alpha_minus = 0.5 * sigma**2 * dt / (dx**2) - 0.5 * dt / dx * (r - 0.5 * sigma**2)

        # Backward Calculation
        keep_flag = np.zeros([N+1, M+1])
        x_all_array = np.arange(0,N+1)/N * (b - a) + a
        x_internal_array = np.arange(1,N)/N * (b - a) + a
        for i in reversed(range(M)):
            u[1:N,i] = alpha_rdt * u[1:N,i+1] \
                     + alpha_plus * u[2:(N+1),i+1] \
                     + alpha_minus * u[0:(N-1),i+1]
            u[1:N,i] = np.maximum(u[1:N,i], K - S0 * np.exp(x_internal_array))
            keep_flag[:,i] = u[:,i] > K - S0 * np.exp(x_all_array)
```

0.1.3 3D plot

In [3]:
```
        fig = plt.figure(figsize=(12,8))
        ax = fig.gca(projection='3d')

        x = t_boundary
        y = S0 * np.exp(x_all_array)
        x, y = np.meshgrid(x, y)

        z = u
        ax.set_xlabel('Time')
        ax.set_ylabel('Stock Price')
        ax.set_zlabel('Option Price')
        ax.set_ylim([180.0, 50.0])
        ax.plot_wireframe(x, y, z, color='blue',linewidth=0.3)

        exercise_flag = u - (K - S0 * np.exp(x_all_array)).reshape(101, 1)
        exercise_flag[exercise_flag > 0] = None
        exercise_flag[exercise_flag <= 0] = 0
        ax.contourf(x, y, - exercise_flag, zdir='z', cmap="autumn")
        ax.set_zlim(0, 60)

        plt.show()
```

```
C:\Users\masay\Anaconda3\lib\site-packages\ipykernel_launcher.py:17: RuntimeWarning: invalid va
```