Python Code for Monte Carlo Simulation of Selected Processes with Applications to Finance

Richard Dewey

This Version: 10/01/2016

1 Geometric Brownian Motion

$$\mathrm{d}S_t = \mathrm{u}S_t dt + \sigma S_t dW_t$$

code used to generate process:

```
for i in range(2,100):
        sp.iloc[i,1] = sp.iloc[i-1,1] + mu*delta + sigma*math.sqrt(delta)*np.random.normal(0,1)
print(jdx)
```

2 Ornstein - Uhlenbeck Process

$$\mathrm{d}S_t = \theta(u - xt) + \sigma S_t dW_t$$

code used to generate process:

```
for x in range(2,100):
        spou.iloc[x,1] = spou.iloc[x-1,1]*exp(-lambda1*delta1) + mu1*(1-math.exp(-lambda1*delta1)) + (sig1*sqrt((1-math
```

3 Process with Synthetic Trends

$$\mathrm{d}S_t = \mathrm{u}\Delta_t + \sigma\sqrt{\Delta_t} + \Delta_t\lambda\sum_i^N \frac{1}{N}\mathrm{r}_{t-i}$$

code used to generate process:

```
for i in range(21,100):
        for k in range(i-n,i):
                sum1 = sum1 + sp.iloc[k,1]
        spm.iloc[i,1] = spm.iloc[i-1,1] + mu*delta + sigma*math.sqrt(delta)*np.random.normal(0,1) + delta*lamda*(sum1/n
```

4 Process with Local Volatility

$$\mathrm{d}S_t = \theta(u - xt) + \sigma S_t dW_t$$

code used to generate process:

```
for i in range(2,100):
        spsk.iloc[i,1] = spsk.iloc[i-1,1] + mu*delta + sigma*math.sqrt(delta)*np.random.normal(0,1)
```

5 Jump Diffusion Process

Here, the jump magnitudes J are i.i.d. r.v.$'$s, i.e. the jump-size J is selected by drawing from a pre-specified probability distribution.

$$\mathrm{d}S_t = \mathrm{u}S_t dt + \sigma S_t dW_t + JdN$$

6 Short-term momentum plus Long-term mean reversion

This process is specified in the following way

$$\mathrm{d}S_t = \theta(u - xt) + \sigma S_t dW_t + \Delta_t \lambda \textstyle\sum_i^N \frac{1}{N} \mathrm{r}_{t-i}$$

code used to generate process: This needs to be changed.

```
for x in range(2,100):
        spou.iloc[x,1] = spou.iloc[x-1,1]*exp(-lambda1*delta1) + mu1*(1-math.exp(-lambda1*delta1)) + (sig1*sqrt((1-math
```