

## Help

```
// (c) J. Poirrot and P. Tankov, June-September 2006
// Permission granted to redistribute and modify this code if the above copyright
// Direct all questions concerning this code to tankov@math.jussieu.fr

// Simulation of the CGMY process with Levy measure truncated at level eps
// (jumps smaller than eps in absolute value are replaced with their mean)
// Uses the algorithm in Madan and Yor (), see also Poirrot and Tankov (2006)
// The Levy measure of CGMY process is
//  $C \exp(-M x) / x^{1+Y} 1_{\{x>0\}} + C \exp(-G |x|) / |x|^{1+Y} 1_{\{x<0\}}$ 
// the gamma parameter of the Levy triplet is chosen in such way that
// the mean of  $X_1$  is equal to  $C \gamma(1-Y) (M^{Y-1} - G^{Y-1})$ 
// this is the natural 'zero drift' version of the process
// corresponding to using a subordinator without drift

#ifndef _CGMYSIM
#define _CGMYSIM

#include "math/numerics.h"

class CGMYSimulator
{
    const double C, G, M, Y;
    const double eps;
    const int generator;
    double A, B, d, lambda, P;
public:
    CGMYSimulator(double xC, double xG, double xM, double xY, double xeps, int xge
    double sim(double t); // simulate a t-increment of the truncated CGMY process
    // returns the increment value
    bool simtojump(double &t, double &before, double &after); // simulate the (
    // up to the first jump or up to time t, if the jump arrives after t
    // on entry, t contains the time step
    // returns true if the jump arrives after t and false otherwise
    // if true is returned, before contains the increment value
    // if false is returned, t contains the jump moment, before contains the proce
    // and after contains the process value after the jump
    double cumulant(int n);
    // returns the n-th cumulant of  $X_1$  (computed theoretically):
```

```
// cumulant(1) corresponds to the mean, cumulant(2) to the variance etc.  
double gamma_mart();  
// returns the additional drift which makes the process martingale  
};  
  
#endif
```