



```

double price[], double solution[], double *delta);

double Asian_CGMY_FusaiMeucci(double spot,
                                double strike,
                                double maturity,
                                double rf,
                                double dividend,
                                double CCGMY, double GCGMY,
                                double MCGMY, double YCGMY,
                                int nmonitoringdates,
                                double lowlim,
                                double uplim,
                                int nquadpoints,          /
                                /n. of quadrature points
                                long nfft,
                                double price[],
                                double solution[], double *delta);

double Asian_DE_FusaiMeucci(double spot,
                              double strike,
                              double maturity,
                              double rf,
                              double dividend,
                              double sgDE, double lambdaDE,
                              double pDE, double eta1DE, double eta2DE,
                              int nmonitoringdates,
                              double lowlim,
                              double uplim,
                              int nquadpoints,          //
                              n. of quadrature points
                              long nfft,
                              double price[],
                              double solution[], double *delta);

//OUTPUT: Contains the solution
double DiscreteAsian(int model,          //modello
                     double spot,
                     double strike,
                     double rf,

```

```

        double dt,
        int ndates,
        double lowlim,
        double uplim,
        int npoints,          //n. of quadra
ture points
        long nfft,           //n. of po
ints for the fft inversion
        double ModelParameters[], //the para
meters of the model
        double price[],
        double solution[], double *delta);
//OUTPUT: Contains the solution

//compute the moments of L
void newmomentsAM(int model, double rf, double dt, int max
moment,
        int ndates, double parameters[], double *
*montable);

//compute the moments of the arithmetic average given the
moments of L
void newmomentsArithM(int ndates, double Lmoments[],
double *AvgMoments);

//compute the probability bound
//using the moment bound
double boundAM(int model, double bound, double rf, double
dt, int maxmoment,
        int ndates, double parameters[], double
moments[]);

//We find in an automatic way the extremes of integration
int findlowuplimit(int model, double rf, double dt, int
maxnummoments,
        int ndates, double lowfactor, double up
factor,
        double parameters[], double extremes[]);

#endif //PremiaCurrentVersion

```

## References