

[Help](#)

```
#ifndef _ALFONSI_H
#define _ALFONSI_H

#include "
href.././common/optype_h_src.pdfoptype.h"
#include "pnl/pnl_mathtools.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_cdf.h"

/*////////////////////////////////////////*/
double psik(double t, double k);
double DiscLawMatch5(int generator);
double DiscLawMatch7(int generator);
void Heston01(double *x1, double *x2, double *x3, double *x4, double dt, double
void Heston02(double *x1, double *x3, double dw2, double rho);
void fct_Heston(double *x1, double *x2, double *x3, double *x4, double dt, doubl

/* see alfonsi.c*/
int HestonSimulation_Alfonsi(int flag_SpotPaths, PnlMat *SpotPaths, int flag_VarP

int HestonSimulation_Alfonsi_Modified(int flag_SpotPaths, PnlMat *SpotPaths, int

/* see alfonsi.c*/
int BatesSimulation_Alfonsi(int flag_SpotPaths, PnlMat *SpotPaths, int flag_VarP

/* Functions used in the regression basis in Longstaff-Schwartz algorithm*/
// Approximation formula for a european option under Heston model.
int ApAntonelliScarlattiHeston(double S, NumFunc_1 *p, double T, double r, doub

// Approximation formula for a european option under Heston model.
int ApAlosHeston(double S, NumFunc_1 *p, double T, double r, double divid, doub

// Approximation formula for a european option under Bates model.
int ApAlosBates(double S, NumFunc_1 *p, double T, double r, double divid, doubl

// Approximation formula for a european asian-option under Black-Scholes model.
int Ap_FixedAsian_BlackScholes(double Current_Spot, double Current_Avg, double C
```

#endif