

[Help](#)

```
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2011+2) //The "#els
#else

#include "pnl/pnl_complex.h"
#include "
href../../common/math/libor_affine_model/libor_affine_framework_h_src.pdflibo
#include "
href../../common/math/libor_affine_model/libor_affine_models_h_src.pdflibo_a

///  

//***** CIR 1d Model*****//

void phi_psi_cir1d(PnlVect *ModelParams, double t, dcomplex u, dcomplex *phi_i,
{
    double lambda, theta, eta, SQR_eta;
    dcomplex z1, z2;
    double b_t, a_t;

    //x0      = GET(ModelParams, 0);
    lambda = GET(ModelParams, 1);
    theta  = GET(ModelParams, 2);
    eta    = GET(ModelParams, 3);
    SQR_eta = SQR(eta);

    a_t = exp(-lambda * t);

    if (lambda == 0.) b_t = t;
    else b_t = (1. - a_t) / lambda;

    z1 = RCsub(1., RCmul(2 * SQR_eta * b_t, u));
    *phi_i = RCmul(-lambda * theta / (2 * SQR_eta), Clog(z1));

    z1 = RCmul(a_t, u);
    z2 = RCsub(1., RCmul(2 * SQR_eta * b_t, u));
    *psi_i = Cdiv(z1, z2);
}

double MaxMgfArg_cir1d(PnlVect *ModelParams, double T)
{
```

```

double lambda, eta;
double b_t;

//x0      = GET(ModelParams, 0);
lambda = GET(ModelParams, 1);
//theta   = GET(ModelParams, 2);
eta      = GET(ModelParams, 3);

if (lambda == 0.) b_t = T;
else b_t = (1. - exp(-lambda * T)) / lambda;

return 1. / (2 * SQR(eta) * b_t);
}

///<***** Gamma-OU 1d Model*****///
void phi_psi_gould(PnlVect *ModelParams, double t, dcomplex u, dcomplex *phi_i,
{
    double lambda, alpha, beta;
    double a_t;
    dcomplex z0, z1, z2, z3;

    lambda = GET(ModelParams, 1);
    alpha  = GET(ModelParams, 2);
    beta   = GET(ModelParams, 3);

    a_t = exp(-lambda * t);

    z0 = RCmul(a_t, u);
    z1 = RCsub(alpha, z0);
    z2 = RCsub(alpha, u);
    z3 = RCmul(beta, Clog(Cdiv(z1, z2)));

    *phi_i = z3;
    *psi_i = z0;
}

double MaxMgfArg_gould(PnlVect *ModelParams, double T)
{
    // The maximum is alpha=GET(ModelParams, 2)
    return GET(ModelParams, 2);
}

```

#endif