

Help

```
/*
 * RfBm.cpp
 *
 * Created on: 30 Jan 2018
 * Author: bayerc
 */

#include "
href../../mod/roughbergomi2d/roughbergomi2d_std/RfBm_h_src.pdfRfBm.hpp"
#include "pnl/pnl_specfun.h"
#include <algorithm>

RfBm::RfBm() {
n = 0;
H = 0.0;
gamma = 0.0;
rng_ = 0;
L = std::vector<Vector>(0);
}

double RfBm::A(int i, int j) const {
double ret = 0.0;
if((i < n) && (j < n))
ret = cBm(i,j);
else if((i < n) && (j >= n))
ret = cMixed(i, j-n);
else if((i >= n) && (j < n))
ret = cMixed(j, i-n);
else if((i >= n) && (j >= n))
ret = cfBm(i-n, j-n);
return ret;
}

double RfBm::G(double x) const {
return 2.0 * H
* (pow(x, -gamma) / (1.0 - gamma)
+ gamma * pow(x, -(1.0 + gamma))
* pnl_sf_hyperm_2F1(1.0, 1.0 + gamma, 3.0 - gamma,
1.0 / x) / ((1.0 - gamma) * (2.0 - gamma)));
}
```

```

}

double RfBm::cBm(int i, int j) const {
return (i == j) ? 1.0 : 0.0;
}

double RfBm::cfBm(int i, int j) const {
double t = static_cast<double>(std::min(i,j) + 1) / n;
double s = static_cast<double>(std::max(i,j) + 1) / n;
return pow(t, 2*H) * G(s/t);
}

double RfBm::cMixed(int i, int j) const {
double ret = 0.0;
if(i <= j){
double u = static_cast<double>(i)/n;
double v = static_cast<double>(i+1)/n;
double t = static_cast<double>(j+1)/n;
ret = 2.0 * sqrt(2.0*H*n) * (pow(t - u, 0.5 + H) - pow(t - v, 0.5 + H)) / (1.0 +
}
return ret;
}

RfBm::RfBm(int nI, double HI, PnlRng *rng) {
n = nI;
H = HI;
gamma = 0.5 - H;
rng_ = rng;
x_ = Vector(2 * n);
wrap_x_ = pnl_vect_wrap_array(x_.data(), x_.size());
L = std::vector<Vector>(2 * n, Vector(2 * n, 0.0)); // fill with 0s first.
// Now fill the matrix L.
for (int k = 0; k < 2 * n; ++k) {
double temp = 0.0;
for (int j = 0; j < k; ++j)
temp += L[k][j] * L[k][j];
L[k][k] = sqrt(A(k, k) - temp);
for (int i = k + 1; i < 2*n; ++i) {
double temp = 0.0;
for (int j = 0; j < k; ++j)
temp += L[i][j] * L[k][j];

```

```

L[i][k] = (A(i, k) - temp) / L[k][k];
}
}
}

void RfBm::generate(Vector& W1, Vector& Wtilde) {
    pnl_vect_rng_normal(&wrap_x_, wrap_x_.size, rng_);
    for (int i = 0; i < n; ++i) {
        W1[i] = 0.0;
        for (int j = 0; j <= i; ++j)
            W1[i] += L[i][j] * x_[j];
    }
    for (int i = 0; i < n; ++i) {
        Wtilde[i] = 0.0;
        for (int j = 0; j <= (i + n); ++j)
            Wtilde[i] += L[i + n][j] * x_[j];
    }
}

void RfBm::operator ()(Vector& W1, Vector& Wtilde) {
    this->generate(W1, Wtilde);
}

std::vector<Vector> RfBm::GetA() const {
    std::vector<Vector> A_mat(2*n, Vector(2*n, 0.0));
    for(int i=0; i<2*n; ++i)
        for(int j=0; j<2*n; ++j)
            A_mat[i][j] = A(i,j);
    return A_mat;
}

```