Cholesky Decomposition

The *Cholesky decomposition* of a *positive definite* matrix A is a decomposition of A into a lower triangular matrix L and the transpose of L, denoted L^T. This type of decomposition can only be applied to *positive definite* matrices, and it has important applications in quantitative finance, as will be discussed in a proceeding article. A *positive definite* matrix is a matrix A such that $\mathbf{x}^T A\mathbf{x} > 0 \ \forall$ n-dimensional vectors $\mathbf{x} \neq \mathbf{0}$. The *Cholesky decomposition* is of the following form, $A = LL^T$. Below is the C++ implementation of the *Cholesky algorithm.*

```
//Cholesky decomposition of matrix A
vector<vector<double> > cholesky(vector<vector<double> > A)
{
int n = A.size();
double sum1 = 0.0;
double sum2 = 0.0;
double sum3 = 0.0;
vector<vector<double> > l(n, vector<double> (n));

l[0][0] = sqrt(A[0][0]);

for (int j = 1; j <= n-1; j++)
l[j][0] = A[j][0]/l[0][0];

for (int i = 1; i <= (n-2); i++)
{
for (int k = 0; k <= (i-1); k++)
   sum1 += pow(l[i][k], 2);

    l[i][i]= sqrt(A[i][i]-sum1);

for (int j = (i+1); j <= (n-1); j++)
{
for (int k = 0; k <= (i-1); k++)
       sum2 += l[j][k]*l[i][k];

        l[j][i]= (A[j][i]-sum2)/l[i][i];
}
}

for (int k = 0; k <= (n-2); k++)
sum3 += pow(l[n-1][k], 2);

l[n-1][n-1] = sqrt(A[n-1][n-1]-sum3);
```

```
return l;
}
```