Prof. Dr. Karsten Urban
Kristina Steih

Institute for Numerical Mathematics
Ulm University
Summer Term 2014

Numerical Finance – Sheet 4

(Exercise Class May 21, 2014)

Exercise 1: Geometric Brownian Motion

Let $(\Omega, \mathcal{P}, \mathbb{P})$ be a probability space equipped with a filtration $(\mathcal{P}_t)_{t\geq 0}$ for the Wiener process $(W_t)_{t\geq 0}$.

In the standard one-dimensional Black-Scholes model, the stock price S_t is modeled by the SDE

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where $\mu \in \mathbb{R}$ is the drift and $\sigma > 0$ the volatility.

a) Use Itô's lemma to show that the stochastic process

$$S_t := S_0 \exp\left\{\sigma W_t + \left(\mu - \tfrac{1}{2}\sigma^2\right) t\right\}, \qquad t \geq 0,$$

solves this SDE.

b) Show that $\mathbb{E}[S_t] = e^{\mu t} S_0$ and $\mathrm{Var}[S_t] = e^{2\mu t} S_0^2 (e^{\sigma^2 t} - 1)$ by using the fact that $\log(S_t/S_0)$ is normally distributed.

Programming Exercise 1: Sparse Grids and the Smolyak Method (35 Points)

We would like to compare numerical cubature based on sparse and on full grids in 2D.

As quadrature rules, we use the composite Simpson's formulae

$$Q^{(\ell)}(f) := \frac{h}{3}\left(f(0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(1)\right), \tag{1}$$

where $n := 2^\ell$ is the number of subintervals with length $h := \frac{1}{n}$ ($\ell \geq 1$ is called the *level* of the quadrature rule).

On the homepage, you find header files for two classes:

CompositeSimpson.h:

Provides the functionality for one-dimensional quadrature rules of the form (1), i.e. it computes the quadrature points and weights. Moreover, it defines a difference operator (necessary for the computation of $\Delta^{(i)}$.

SmolyakIndex.h:

Realizes the loop over the set of 2-dimensional indizes $i = (i_1, i_2)$ so that $|i| \leq k$. This allows the computation of $Q(k, d)$ in equation (3.5.2).

Example usage:

```
int k = 2;
SmolyakIndex index(k, d);

while(!index.done(k)){
    cout << index << endl;
    index.next(k);
}
```

Output:

```
{0,0}
{1,0}
{2,0}
{0,1}
{1,1}
{0,2}
```

Use these in order to implement sparse grid and full grid cubature for two-dimensional functions.

a) Visualize the full and sparse grids for different levels ℓ.

b) For each of the following functions,

   (i) plot the function,

   (ii) compute the full and sparse grid approximation of the integral $\int_{[0,1]^2} f(x, y) dx\, dy$ for different levels ℓ and

   (iii) visualize and compare the error convergence (you may use any tool you like for the computation of the exact integrals).

The functions are given by

- $f(x, y) = (x - 0.5)^2 + (y - 0.5)^2$. At which level do you obtain an exact solution (up to machine precision)? Why?
- $f(x, y) = e^{-(3x-1.5)^2} e^{-(3y-1.5)^2}$. What do you observe? Would you prefer full or sparse grids?
- $f(x, y) = e^{-50(x^2-y)^2}$. What do you observe? Why?