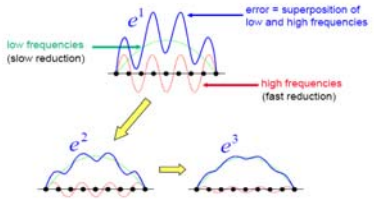
A Multigrid Method for the Black-Scholes equation

정 다래(tinayoyoyo@naver.com), 김 준석(junseok_kim@yahoo.com)

Introduction

기존의 일반적인 반복법(Jacobi, Gauss-Seidel, etc.)은 smoothing 특징을 가지고 있다.



위 그림에서 볼 수 있듯이 High-frequency error를 줄이는데 효과적인 특징을 지니지만 Low-frequency error는 상대적으로 거의 변하지 않는다는 단점을 지니고 있다. 이러한 단점을 보완해갈 수 있는 방법이 바로 Multigrid 방법이다.이는 효율적으로 error를 줄여가면서 정확한 수치 해를 구할 수 있다.

The Black-Scholes model

• 두 자산(x,y)에 대한 Black-Scholes 방정식

$$\frac{\partial u}{\partial \tau}=\frac{1}{2}(\sigma_1 x)^2\frac{\partial^2 u}{\partial x^2}+\frac{1}{2}(\sigma_2 y)^2\frac{\partial^2 u}{\partial y^2}+\sigma_1\sigma_2\rho xy\frac{\partial^2 u}{\partial x\partial y}$$
$$+rx\frac{\partial u}{\partial x}+ry\frac{\partial u}{\partial y}-ru,\qquad \tau=T-t$$

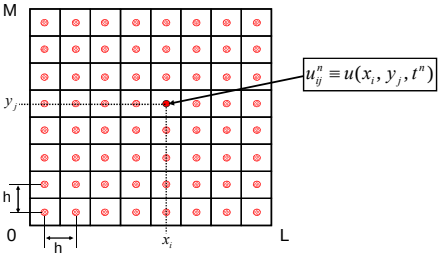
• Linear Boundary condition
경계에는 다음의 조건을 적용시키자.

$$\frac{\partial^2 u}{\partial x^2}(0,y,\tau)=\frac{\partial^2 u}{\partial x^2}(L,y,\tau)=0$$
$$\frac{\partial^2 u}{\partial y^2}(x,0,\tau)=\frac{\partial^2 u}{\partial y^2}(x,M,\tau)=0$$
$$\forall\tau\in[0,T],\quad for\quad 0\le x\le L,\ 0\le y\le M.$$

Discretization

수치 해는 다음과 같이 정의하자.

$$u_{ij}^n\equiv u(x_i,y_j,t^n)=u((i-0.5)h,(j-0.5)h,n\Delta t)$$
where $\Omega=[0,L]\times[0,M]$,
$h=L/Nx=M/Ny$,
$\Delta t=T/Nt$,
$i=1,\cdots,Nx$ and $j=1,\cdots,Ny$



Black-Scholes 방정식의 공간미분에 대하여는 중앙 차분법(central difference method)을, 시간미분에는 후방 차분법(implicit difference method)을 이용하자.

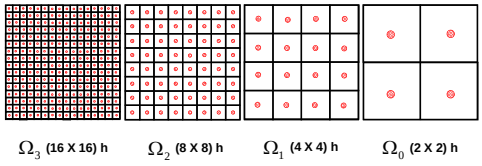
$$\frac{u_{ij}^{n+1}-u_{ij}^{n}}{\Delta t}=L_{BS}\,u_{ij}^{n+1},$$

유한차분 연산(L_{BS})을 다음과 같이 정의하자.

$$L_{BS}u_{ij}^{n+1}=\frac{(\sigma_1 x_i)^2}{2}\frac{u_{i-1,j}^{n+1}-2u_{ij}^{n+1}+u_{i+1,j}^{n+1}}{h^2}$$
$$+\frac{(\sigma_2 y_j)^2}{2}\frac{u_{i-1,j}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{h^2}$$
$$+\sigma_1\sigma_2\rho x_i y_j\frac{u_{i+1,j+1}^{n+1}-u_{i-1,j+1}^{n+1}-u_{i+1,j-1}^{n+1}+u_{i-1,j-1}^{n+1}}{4h^2}$$
$$+rx_i\frac{u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}}{2h}+ry_j\frac{u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}}{2h}-ru_{ij}^{n+1}.$$

Multigrid method

가장 fine한 grid 사이즈가 16x16이라고 가정한다면 다음 그림은 v-cycle동안 이용되는 grids이 된다. 오른쪽으로 갈수록 도메인이 그 전보다 coarse하다고 한다.



이제, 위의 2D Black-Scholes 방정식을 다음과 같이 다시 나타내자.

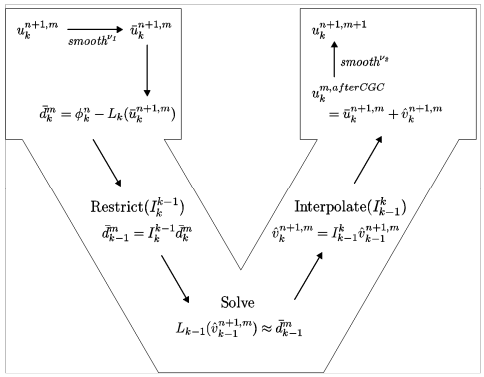
$$L_3(u_{3,ij}^{n+1})=\phi_{3,ij}^n\quad \text{on}\ \Omega_3$$

여기서 좌변과 우변은 다음을 나타낸다.

$$L_3(u_{3,ij}^{n+1})=u_{3,ij}^{n+1}-\Delta t\,L_{BS3}u_{3,ij}^{n+1}$$

$$\phi_{3,ij}^n=u_{3,ij}^n$$

이제 위의 식을 가지고 Multigrid algorithm을 알아보자. 다음은 이해를 돕기 위해 two grid (k, k-1)에서 Multigrid cycle에 대한 algorithm이다.



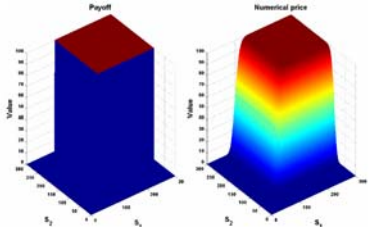
Algorithm 1: The multigrid method.
Data: Given $\mathbf{A}_h$, $\tilde{\mathbf{u}}_h$, $\mathbf{b}_h$, ν_1, ν_2 and γ;
Result: Return $\mathbf{u}_h$ that satisfies $\mathbf{A}_h\mathbf{u}_h=\mathbf{b}_h$.

```
Multigrid(A_h, ũ_h, b_h, ν1, ν2, γ)
begin
    if Coarsest Level then
        return u_h = Solve(A_h, b_h);
    end
    else
        for i=1 to γ do
            ũ_h = Smooth(A_h, ũ_h, b_h, ν1);
            Δb_h = b_h − A_h ũ_h;
            Δb_2h = Restriction(Δb_h);
            Compute A_2h;
            Δu_2h = Multigrid(A_2h, 0, Δb_2h, ν1, ν2, γ);
            Δu_h = Prolongation(Δu_2h);
            ũ_h = ũ_h + Δu_h;
            ũ_h = Smooth(A_h, ũ_h, b_h, ν2);
        end
        return u_h = ũ_h;
    end
end
```

Computational results

Multigrid의 수렴성 test를 위해 2D Black-Scholes 방정식에 다음의 cash or nothing option의 payoff를 initial condition 으로 적용하자.

$$u(x,y,0)=\begin{cases}F & \text{if}\quad x\ge K_1\quad\text{and}\quad y\ge K_2\\ 0 & \text{otherwise}\end{cases}$$



1. Convergence test

다음의 parameter

$$\Omega=[0,300]\times[0,300]$$
$$K=100\quad X_1=X_2=100\quad T=0.1$$
$$r=0.03\quad \sigma_1=\sigma_2=0.5\quad \rho=0.5$$

를 적용하여 각 mesh마다 error의 L2 norm을 구해본 결과이다.

Solution grid	$\|e\|_2$	Rate
32 X 32	**0.028161**	
64 X 64	**0.014562**	**0.95**
128 X 128	**0.006928**	**1.07**
256 X 256	**0.003572**	**0.96**

Error는 Mesh가 fine해 질수록 작아지며 이 scheme이 **first order accurate**하다.

2. Multigrid performance

Solution grid	Ave. no. of iteration per time step	CPU (s)
32 X 32	**1.00**	**0.156**
64 X 64	**1.00**	**0.641**
128 X 128	**2.00**	**4.922**
256 X 256	**2.24**	**21.922**

MG method는 mesh가 더 fine해 질수록 각각의 time step당 convergence에 대한 Multigrid iteration의 수가 조금 증가하였지만 이것은 거의 **grid independent** 하다고 할 수 있다.