An Accurate and Efficient Numerical Method for the Black-Scholes equation

Darae Jeong

Junseok Kim

Department of Mathematics

Korea University

Contents

The Black-Scholes model

- **Black-Scholes Equation with 2 underlying assets:**

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

- **Two asset cash or nothing option:**

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

Discretization of 2D BS PDE

- **Discretize the original PDE to a system of difference equation :**

Let

$$u_{ij}^n \equiv u(x_i, y_j, t^n) = u((i-0.5)h, (j-0.5)h, n\Delta t)$$

where

$$\Omega = [0, L] \times [0, M],$$

$$h = L / Nx = M / Ny \qquad \Delta t = T / Nt$$

$$i = 1, \ldots, Nx \quad \text{and} \quad j = 1, \ldots, Ny$$

Discretization of 2D BS PDE

- We apply

  - Implicit Scheme

  - Central difference for the first and second order spatial derivative

  - Linear boundary condition

$$\frac{\partial^2 u}{\partial x^2}(0,y,\tau)=\frac{\partial^2 u}{\partial x^2}(L,y,\tau)=\frac{\partial^2 u}{\partial y^2}(x,0,\tau)=\frac{\partial^2 u}{\partial y^2}(x,M,\tau)=0$$

$$\forall\, \tau \in [0,T], \qquad for \qquad 0 \le x \le L,\ 0 \le y \le M.$$

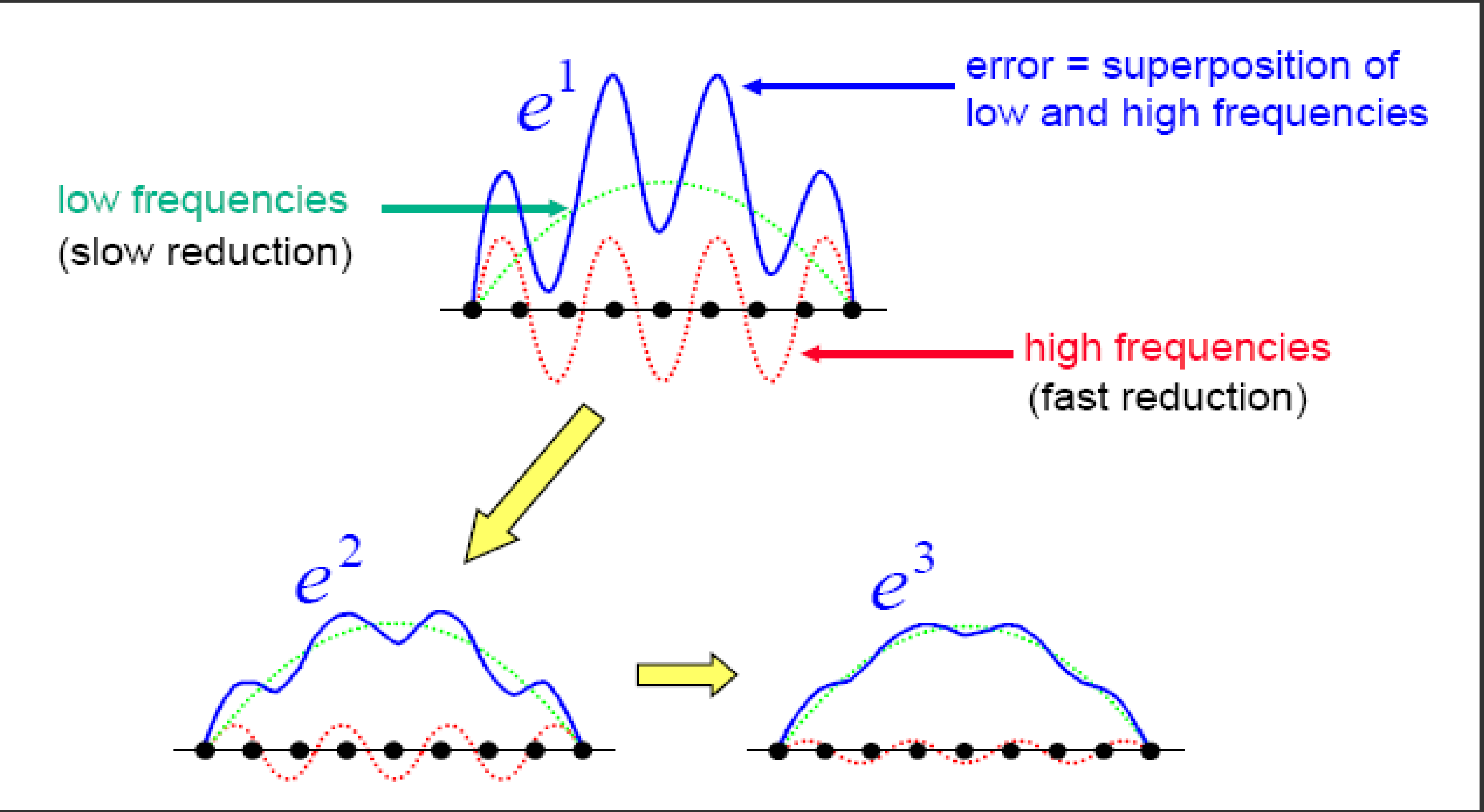
Discretization of 2D BS PDE

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

where the discrete difference operator L_{BS} **is defined by**

$$\begin{aligned} L_{BS} u_{ij}^{n+1} = & \frac{(\sigma_1 x_i)^2}{2} \frac{u_{i-1,j}^{n+1} - 2u_{ij}^{n+1} + u_{i+1,j}^{n+1}}{h^2} \\ & + \frac{(\sigma_2 y_j)^2}{2} \frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{h^2} \\ & + \sigma_1 \sigma_2 \rho x_i y_j \frac{u_{i+1,j+1}^{n+1} + u_{i-1,j-1}^{n+1} - u_{i-1,j+1}^{n+1} - u_{i+1,j-1}^{n+1}}{4h^2} \\ & + r x_i \frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2h} + r y_j \frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2h} - r u_{ij}^{n+1}. \end{aligned}$$

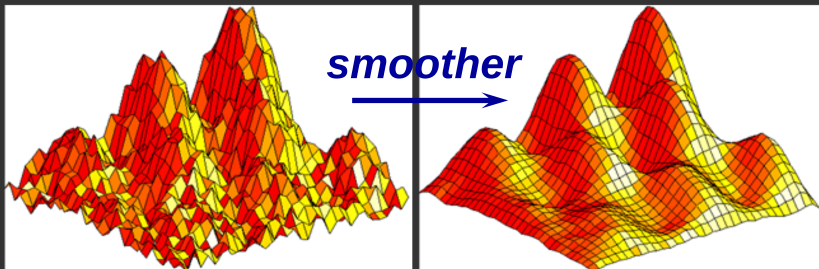
Relaxation method



Relaxation methods

smooth the error quickly **but** *converge slowly!*

Two-grid Algorithm



Fine

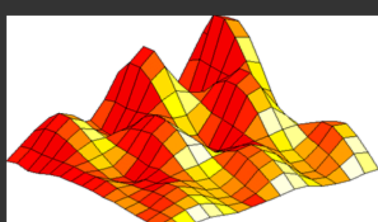
Relax on $A^h u^h = f^h$

Compute $r^h = f^h - A^h u^h$

Restrict

$r^{2h} = I_h^{2h} r^h$

Coarse



Solve $A^{2h} e^{2h} = r^{2h}$

Interpolate

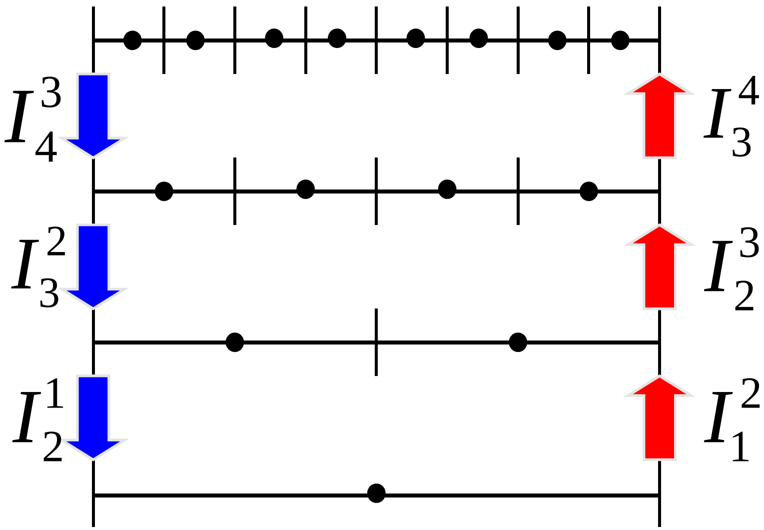
$e^h \approx I_{2h}^h e^{2h}$

Correct

$u^h \leftarrow u^h + e^h$

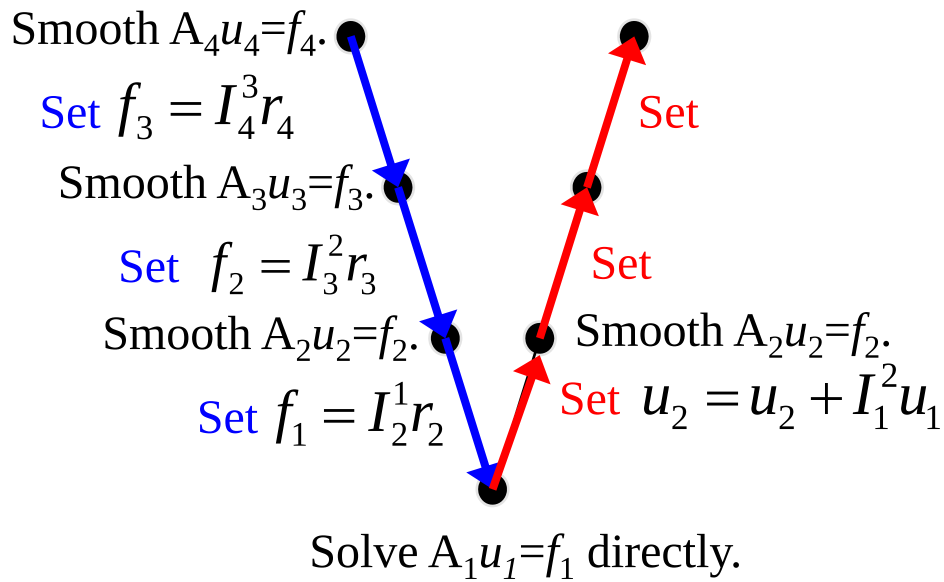
coarser grid has fewer cells (less work & storage)

Linear Multigrid



V Cycle to solve $A_4 u_4 = f_4$.

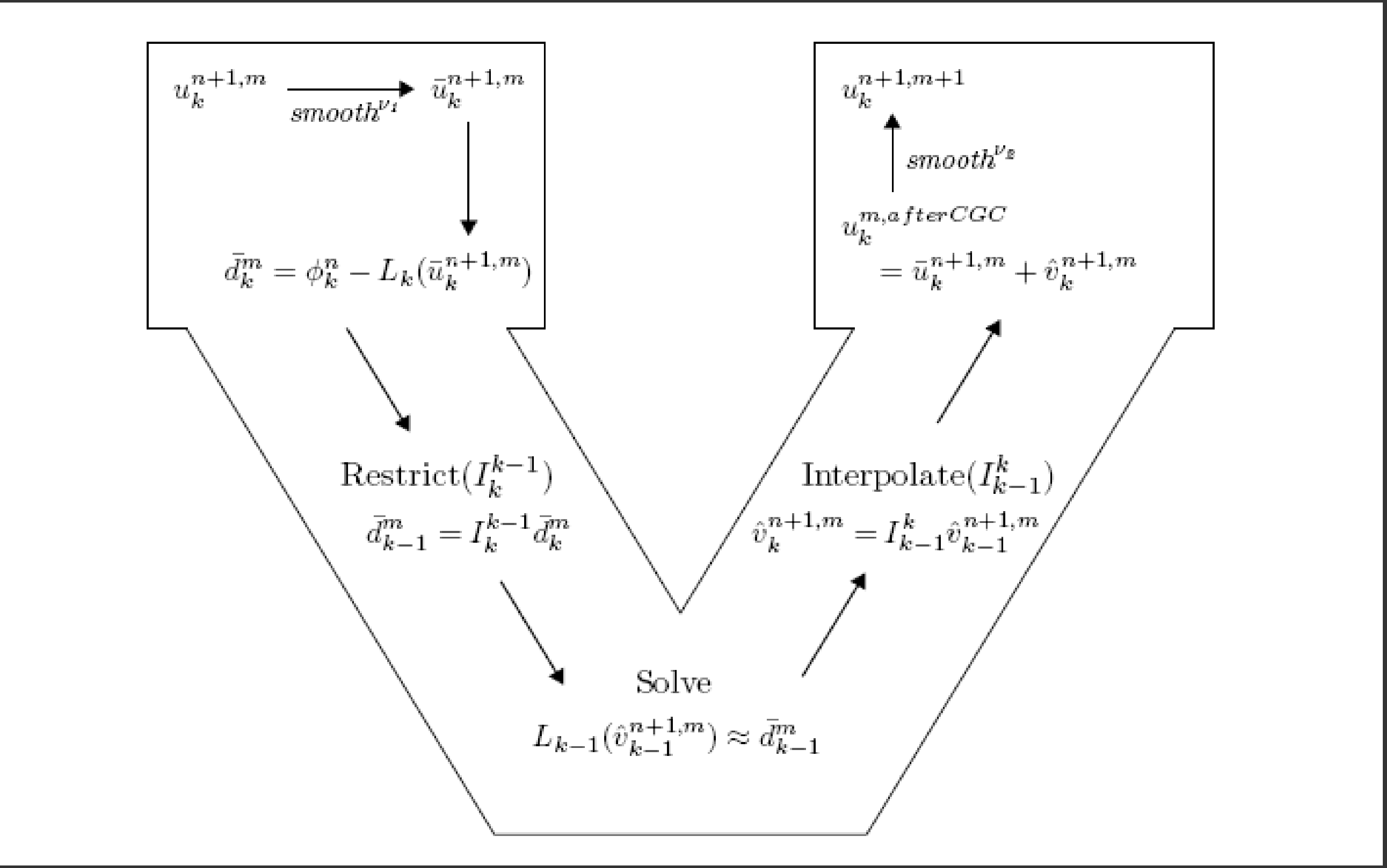
Note: $r_i = f_i - A_i u_i$ is the *residual*.



Transfer operators to transfer data between grids of different resolution.

The V shows the schedule of operations that the algorithm applies level by level.

Linear Multigrid



Computational Results

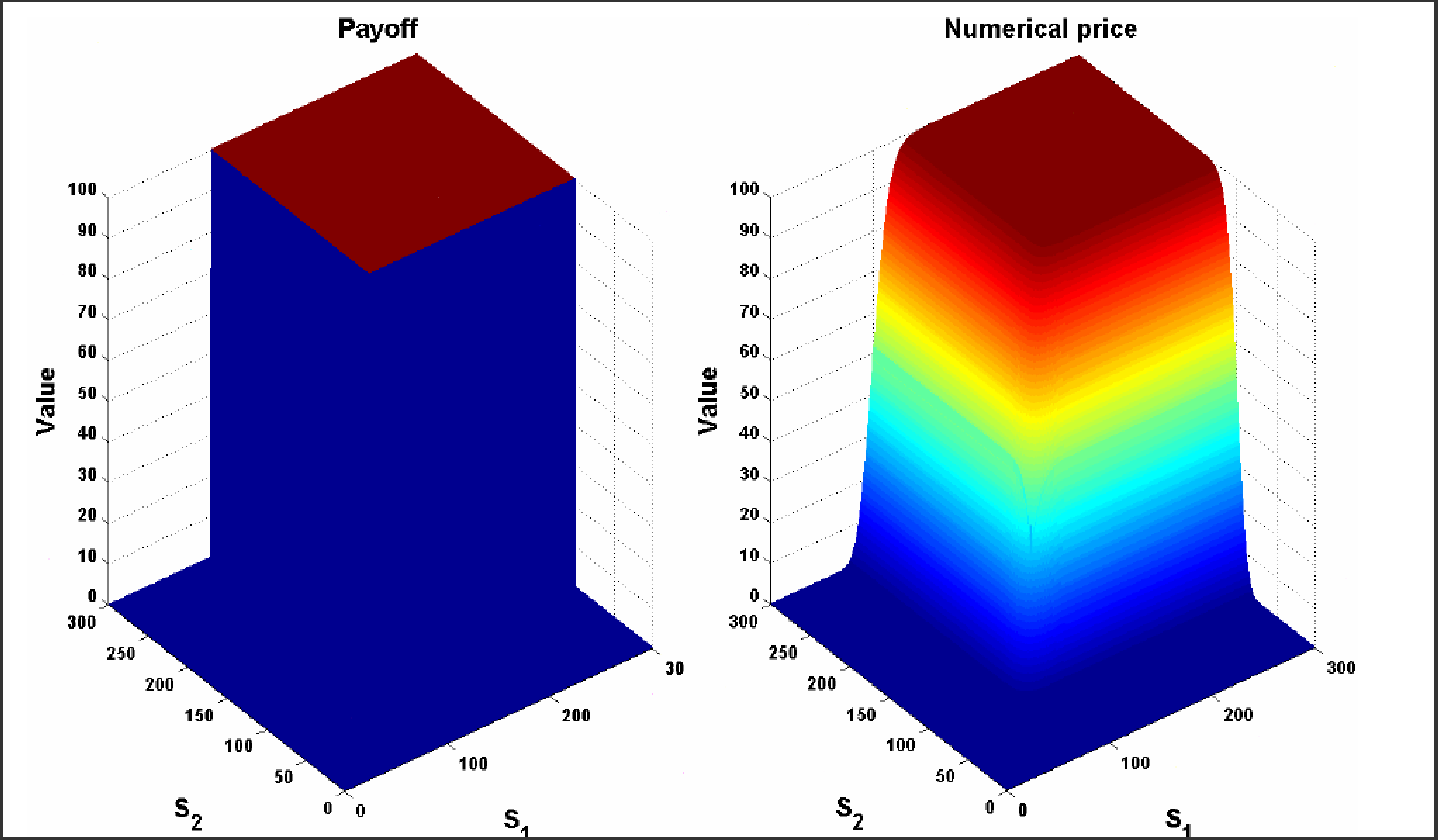
- We test the Multigrid method technique with two asset cash or nothing call option
- Linear multigrid
  - V-cycle (3 relaxation)
  - Gauss – Seidel relaxation.
- Parameter

$$\Omega = [0,300]\times[0,300]$$

$$K = 100 \quad X_1 = X_2 = 100 \quad T = 0.1$$

$$r = 0.03 \quad \sigma_1 = \sigma_2 = 0.5 \quad \rho = 0.5$$

Computational Results



Computational Results

- Multigrid scheme is *first-order accurate*.
- Table: The L2 norms of errors and convergence rates for u at time T=0.1

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

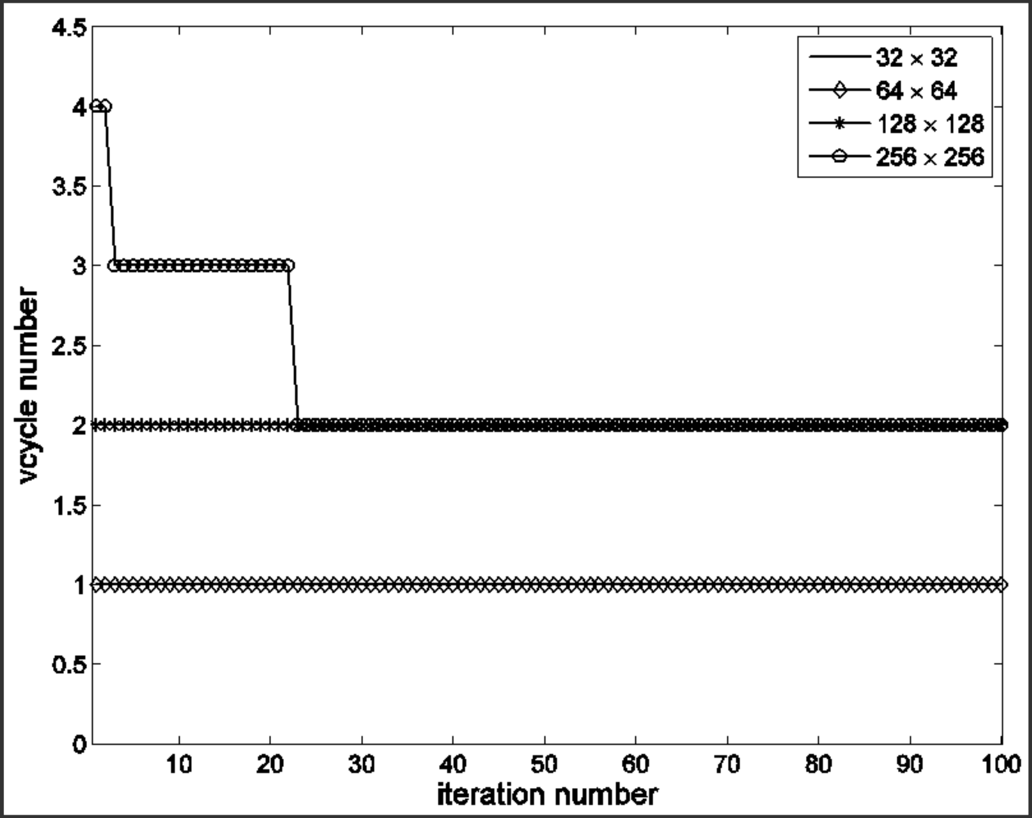
Computational Results

- **Although the number of multigrid iterations for convergence at each time step slowly increases, it is essentially *grid independent*.**

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

Computational Results

- Grid independence with an iteration convergence tolerance of 1.0e-5



Further Research

- Compare various interpolation methods in multigrid.

  - direct restriction, bilinear interpolation,

    bicubic interpolation and other customized

    interpolation

- Fourth order multigrid methods

  for the Black Scholes equations.

Thank you!