NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Parallel CFD & Optimization Unit

Laboratory of Thermal Turbomachines

Μέθοδος Πολυπλέγματος (Multigrid)

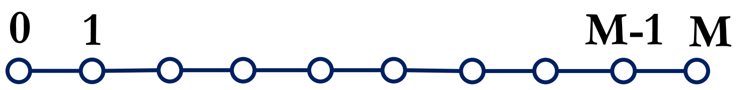
Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Βαρβάρα Ασούτη, Δρ. Μηχ. Μηχανικός ΕΜΠ

kgianna@central.ntua.gr

vasouti@mail.ntua.gr

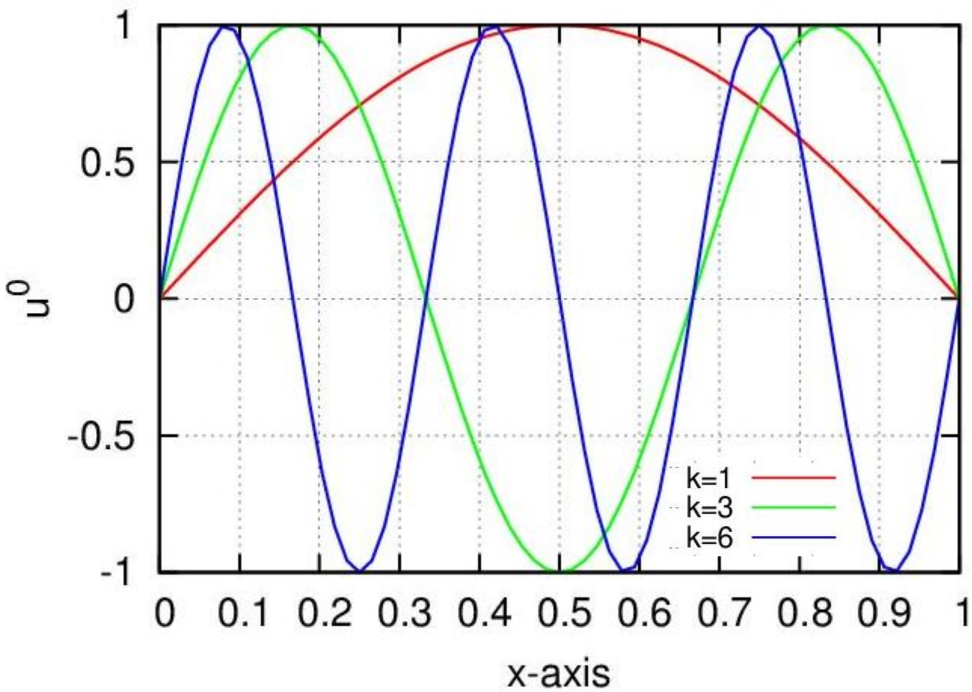
Example



- **Problem:** $-u_{i-1} + 2u_i - u_{i+1} = 0 \qquad 0 \le i \le M$

- **Boundary Conditions:** $u_0 = u_M = 0$

- **Initialization:** $u_i = \sin\left(\frac{ik\pi}{M}\right)$



Discretization

$$-\frac{\vartheta^2 u}{\vartheta x^2} = 0$$

$$\frac{-u_{i-1} + 2u - u_{i+1}}{\Delta x^2} = 0$$

$$-(u_{i-1} + \Delta u_{i-1}) + 2(u_i + \Delta u_i) - (u_{i+1} + \Delta u_{i+1}) = rhs \cdot \Delta x^2$$

$$\Delta u_i = \frac{1}{2}\left[rhs \cdot \Delta x^2 - \left(-u_{i-1} + 2u_i + -u_{i+1}\right) - (-\Delta u_{i-1} - \Delta u_{i+1})\right]$$

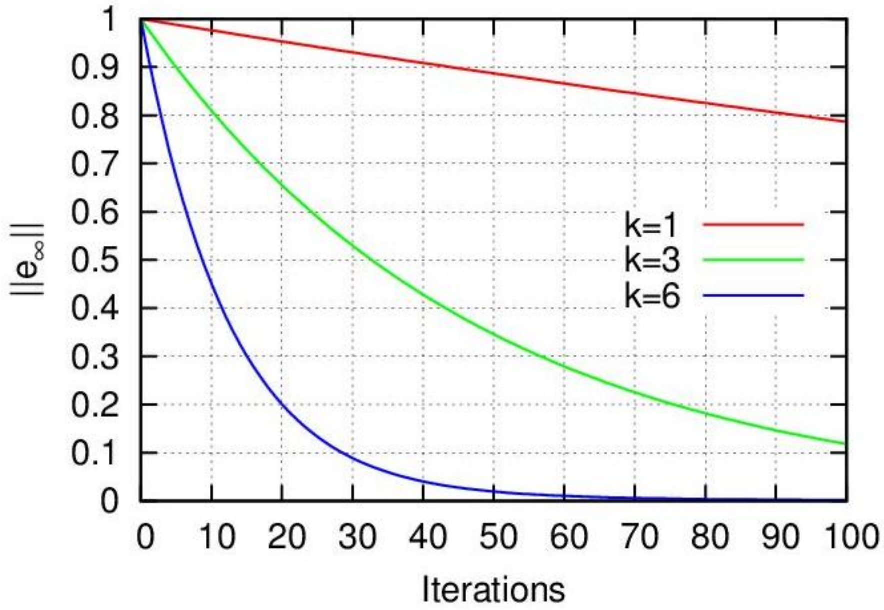
Coding

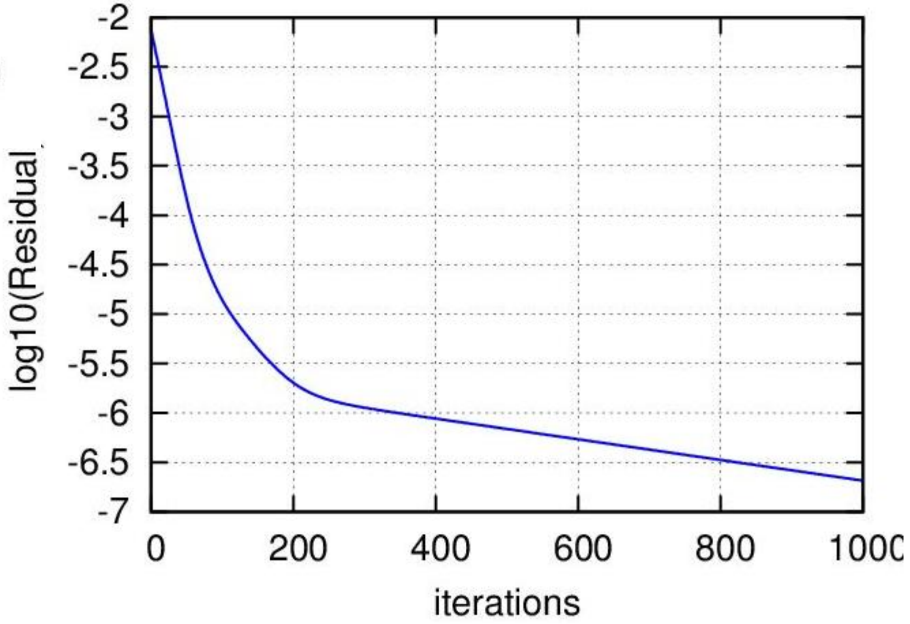
```
dx  = 1.d0/dfloat(nx)
c1p = -1.d0
c1m = -1.d0
cc  =  2.d0
do ic=0,ncyc
   do i=0,nx
      du(i) = 0.d0
   enddo
   restot = 0.d0
   do i=1,nx-1
      au = -u(i-1) + 2.d0*u(i) - u(i+1)
      resid(i) = rhs(i) - au
      du(i) = (dx*dx*rhs(i)-au - c1m*du(i-1) - c1p*du(i+1))/cc
      restot = restot + resid(i)*resid(i)
   enddo
   restot = dsqrt(restot)/dfloat(nx+1)
   error = 0.d0
   do i=1,nx-1
      u(i) = u(i) + du(i)
      error = u(i)*u(i)
   enddo
enddo
```

Compute Δu

Update u

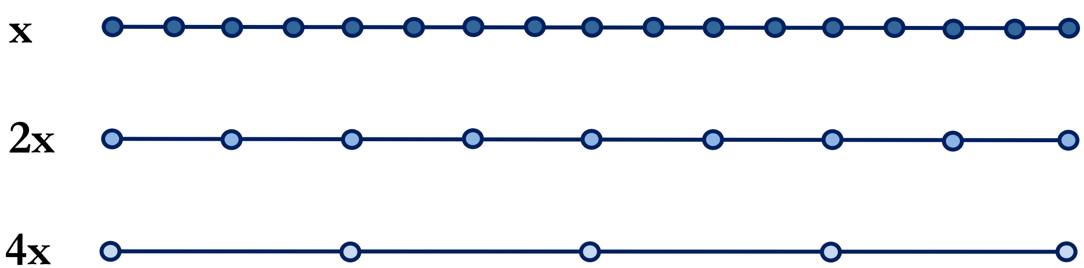
Convergence

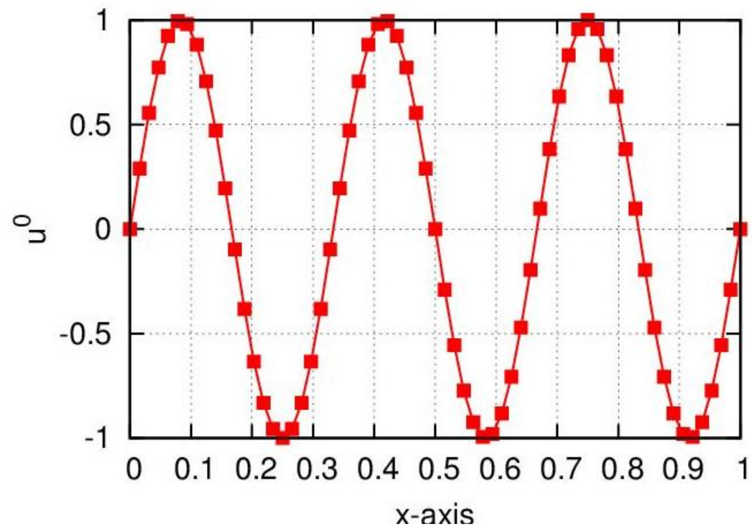


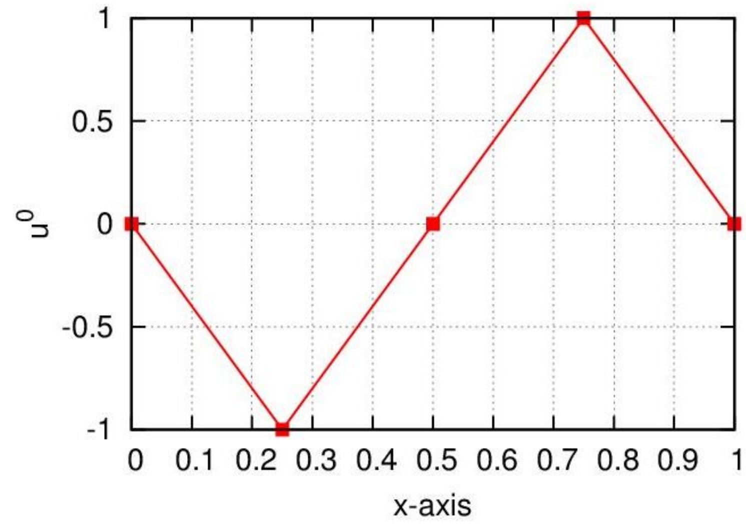


Geometric Multigrid: The Idea

- **Different grid sizes:**
  Ability to solve all wave lengths
  Rapid convergence







Operators

h: Fine grid

H: Coarse grid

- **Restriction**

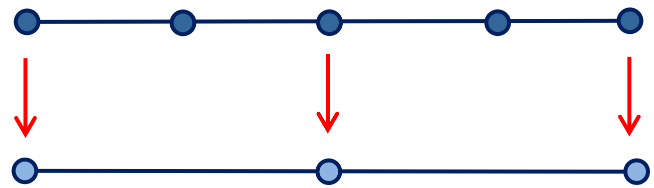
Mapping from the fine grid to the coarse grid

$$I_h^H : \Omega^h \rightarrow \Omega^H$$

$$\Phi^h = I_h^H \Phi^H$$

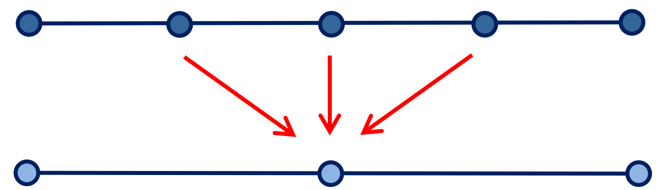
- **Injection**

$$u_i^H = u_{2i}^h$$



- **Full weighting**

$$u_i^H = \frac{1}{4}\left(u_{2i-1}^h + 2u_{2i}^h + u_{2i+1}^h\right)$$



- **Prolongation**

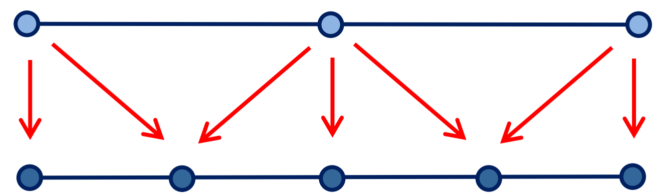
Mapping from the coarse grid to the fine grid

$$I_H^h : \Omega^H \rightarrow \Omega^h$$

$$\Phi^H = I_H^h \Phi^h$$

$$u_{2i}^h = u_i^H$$

$$u_{2i+1}^h = \frac{1}{2}\left(u_i^H + u_{i+1}^H\right)$$

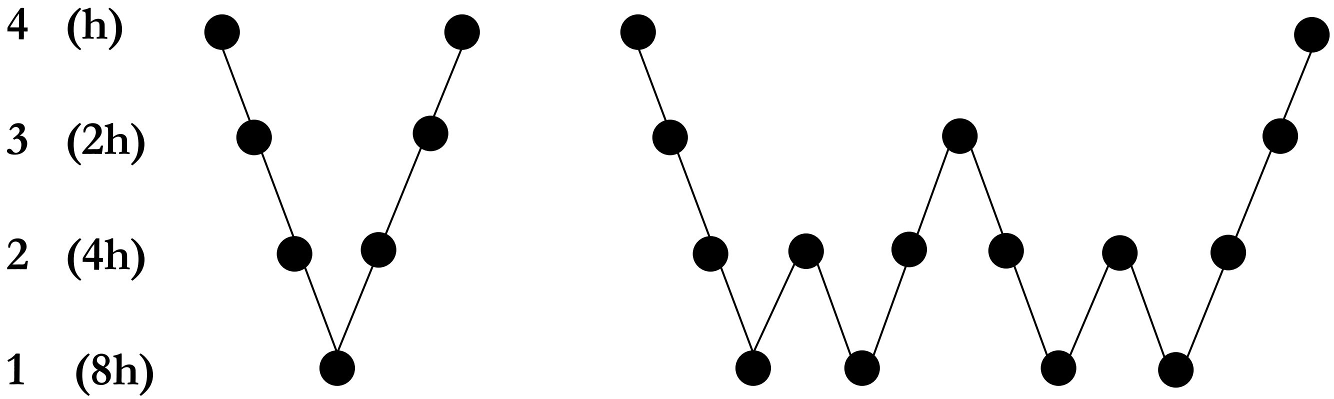


Multigrid Algorithm

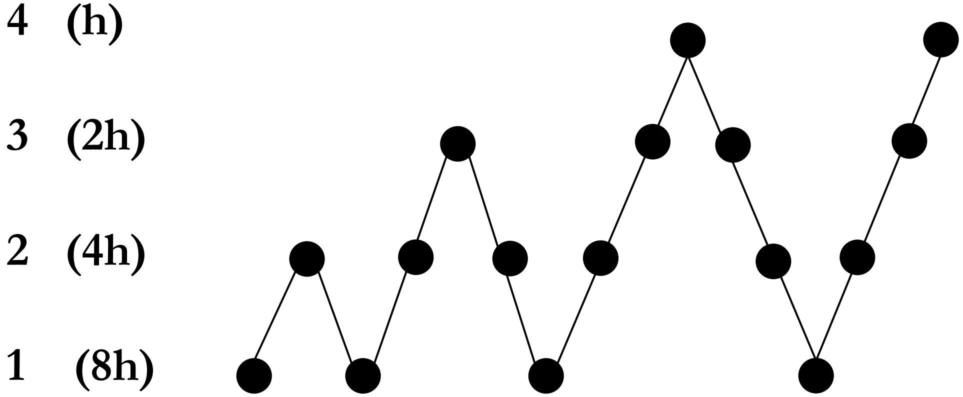
- **V-cycle using two grids (fine/h, coarse/H) :**

1. **Solve** (2-3 Jacobi or Gauss-Seidel iterations) $A_h u = b_h \rightarrow u_h$
2. **Restrict** the residual $r_h = b_h - A_h u_h$ to the coarse grid $r_H = I_h^H r_h$
3. **Solve** (2-3 Jacobi or Gauss-Seidel iterations) $A_H E_H = r_H \rightarrow E_H$
4. **Prolongate** (interpolate) E_H to fine grid $E_h = I_H^h E_H$
   and correct $u_h \leftarrow u_h + E_h$
5. **Go to step 1,** using the updated u_h

Multigrid Schemes



V-cycles
and
W-cycles

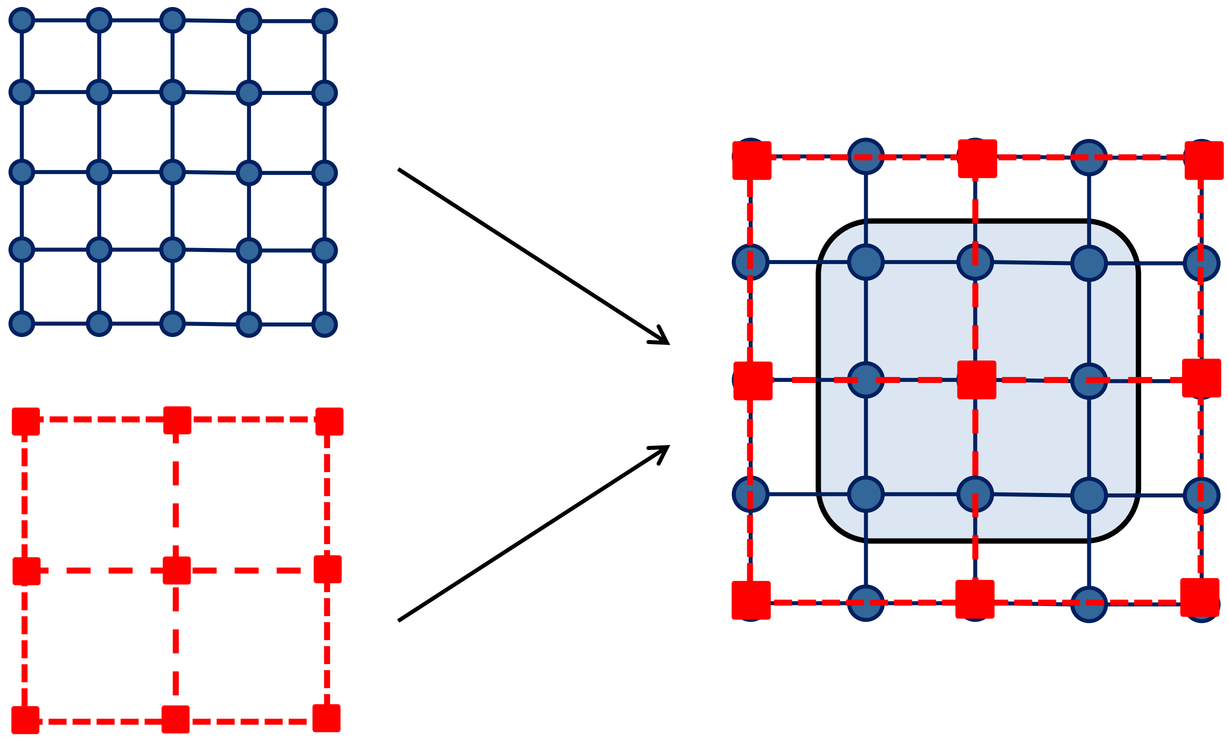


Full Multigrid
(FMG-cycles)

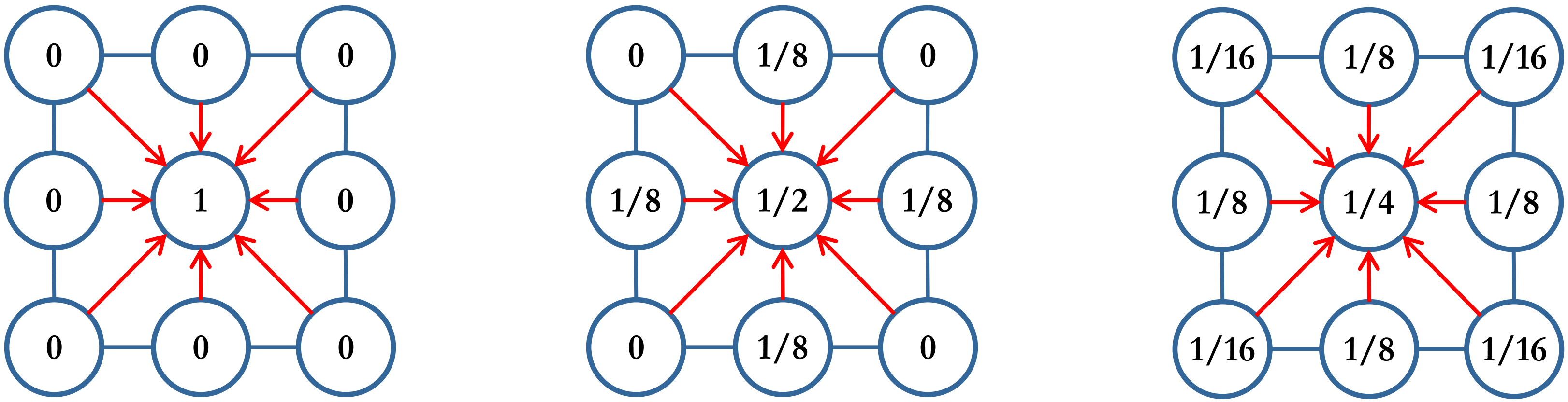
Multigrid in 2 dimensions

- Fine grid:

- Coarse grid:



2D Restriction



- **Injection:** $u^H_{i,j} = u^h_{2i,2j}$

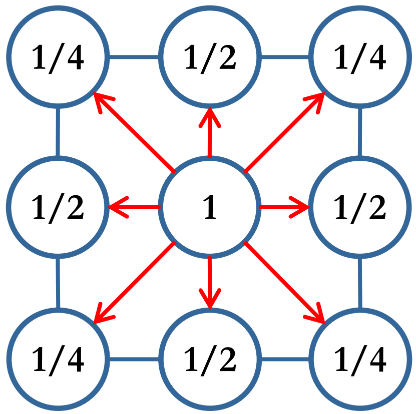
- **Weighting:**
  - **Half**
  - **Full**

$$u^H_{i,j} = \frac{1}{2}u^h_{2i,2j} + \frac{1}{8}\left(u^h_{2i-1,2j} + u^h_{2i+1,2j} + u^h_{2i,2j-1} + u^h_{2i,2j+1}\right)$$

$$u^H_{i,j} = \frac{1}{4}u^h_{2i,2j} + \frac{1}{8}\left(u^h_{2i-1,2j} + u^h_{2i+1,2j} + u^h_{2i,2j-1} + u^h_{2i,2j+1}\right) +$$

$$\frac{1}{16}\left(u^h_{2i-1,2j-1} + u^h_{2i+1,2j-1} + u^h_{2i-1,2j+1} + u^h_{2i+1,2j+1}\right)$$

2D Prolongation



$$u^h_{2i,2j} = u^H_{i,j}$$

$$u^h_{2i+1,2j} = \frac{1}{2}\left(u^H_{i,j} + u^H_{i+1,j}\right)$$

$$u^h_{2i,2j+1} = \frac{1}{2}\left(u^H_{i,j} + u^H_{i,j+1}\right)$$

$$u^h_{2i+1,2j+1} = \frac{1}{4}\left(u^H_{i,j} + u^H_{i+1,j} + u^H_{i,j+1} + u^H_{i+1,j+1}\right)$$

Algebraic Multigrid (AMG)

- Method to solve linear systems based on multigrid principles, without explicit knowledge of the problem geometry (only matrix coefficients)
- Determines coarse "grids", inter-grid transfer operators and the coarse grid equations are exclusively based on the matrix entries

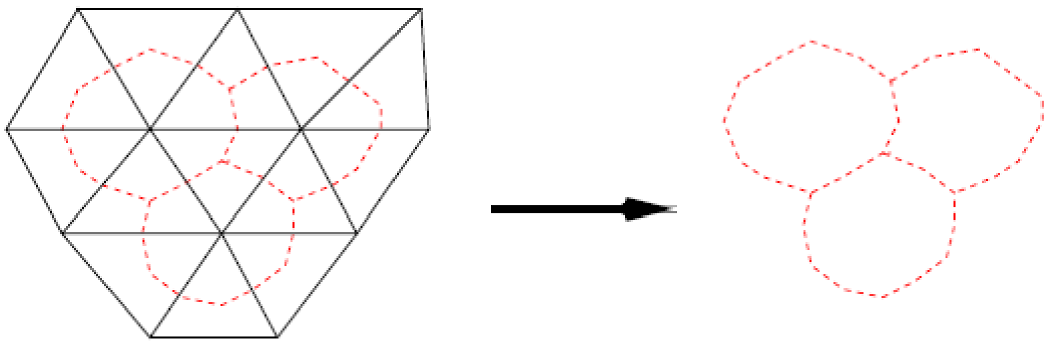
Two-grid algorithm:

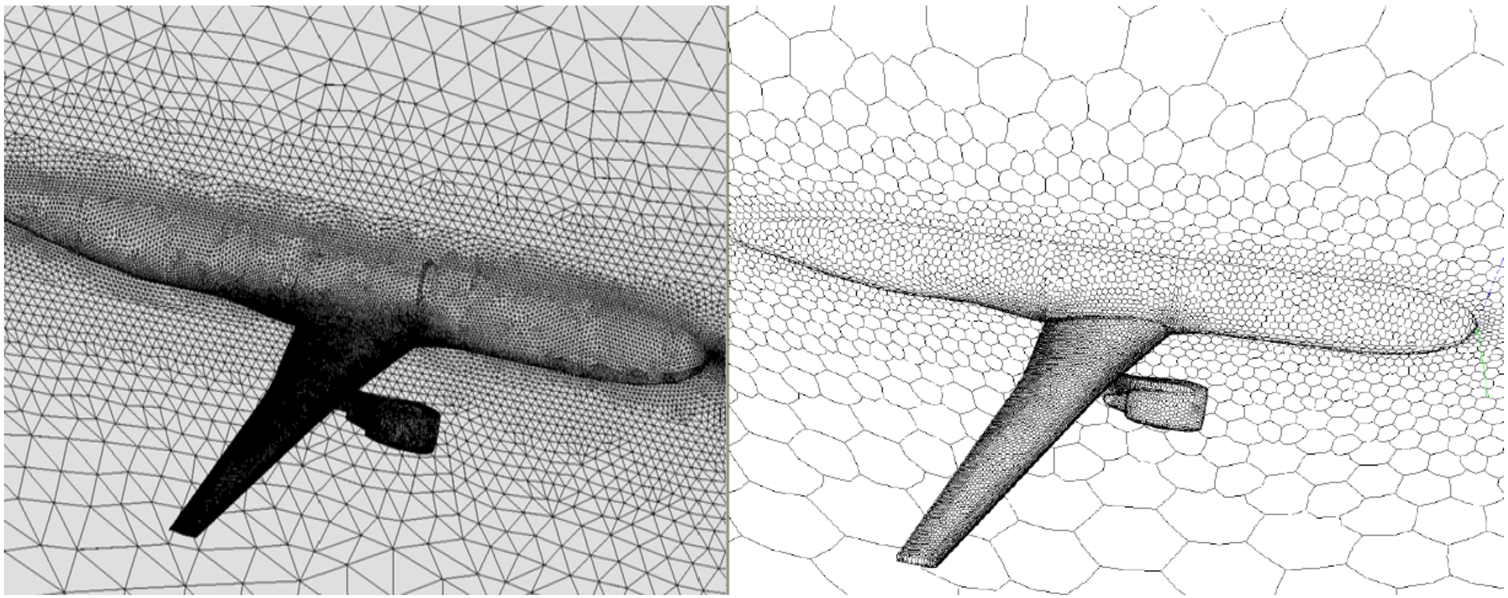
1. Solve $A_h u = b_h \rightarrow u_h$
2. Compute residual $r_h = b_h - A_h u_h = A_h e_h$
3. Solve $A_H e_H = P^T r_H$ with $A_H = R A_h P$
4. Correct $u_h \leftarrow u_h + P e_H$
5. Solve $A_h u = b_h \rightarrow u_h$

Prolongation: $P : R^{n_H} \rightarrow R^{n_h}$, $n_h \times n_H$ matrix

Restriction: R

Agglomeration



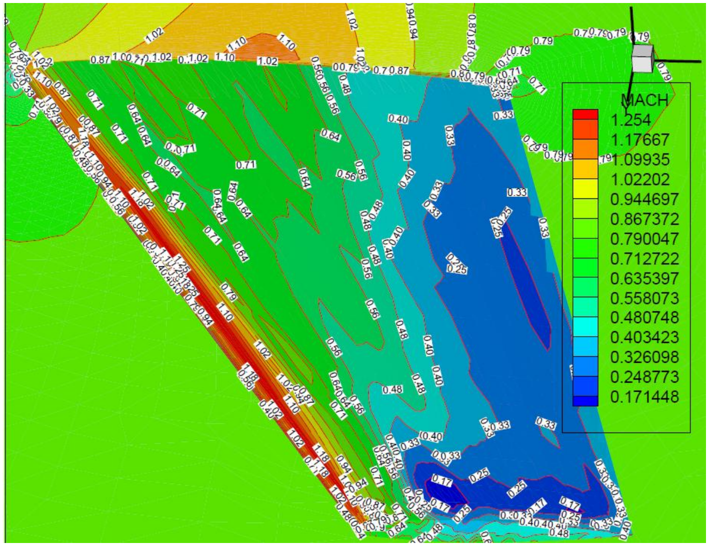


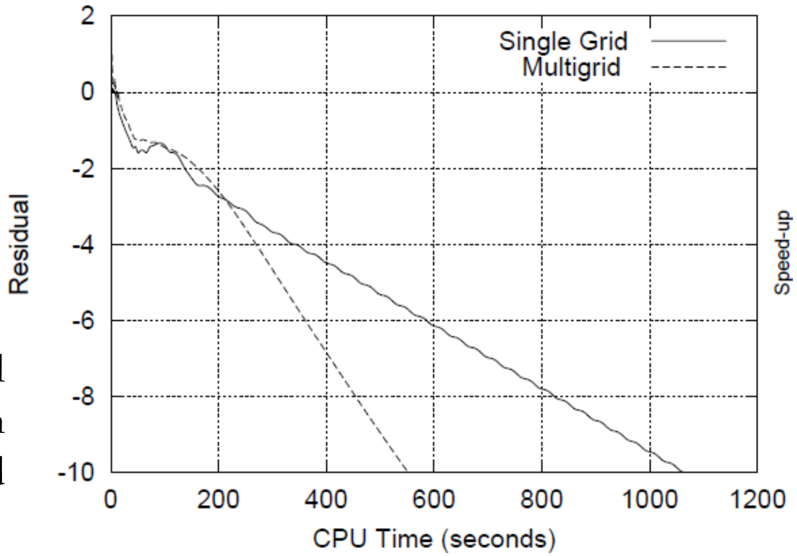
Mavriplis D., Mani K., "Unstructured Mesh Solution Techniques using the NSU3D Solver", AIAA Paper 2014-0081

Applications

- Transonic flow around the M6 wing
  - $M_\infty = 0.8395$
  - $\alpha_\infty = 3.06^o$
  - $Re = 11.72\ 10^6$

- Fine grid: ~67k nodes
- Coarse grid: ~11k nodes





Lambropoulos N., "Multigrid techniques and parallel processing for the numerical prediction of flowfields thtough thermal turbomachines, using unstructured grids", Phd Thesis, NTUA, 2005 (in greek)

References

- Brandt A. "Multi-level Adaptive Solutions to Boundary Value Problems" Mathematics of Computation, **31:**333-390, 1977.
- Brandt A. "Guide to Multigrid Development" 1984.
- Hackbusch W., "Multigrid Methods and Applications" Springer – Verlag 1985.
- Briggs W, Henson, V., McCormick S., "A Multigrid Tutorial", 2nd edition, SIAM publications 2000.