Accurate and Efficient Simulation and Design Using High-Order CFD Methods

Dr. Li Wang and Dr. W. Kyle Anderson

SimCenter: National Center for Computational Engineering
University of Tennessee at Chattanooga
Chattanooga, Tennessee, USA

July 9, 2014

Modern Techniques for Aerodynamic Analysis and Design
2014 CFD Summer School, Beijing, China, July 7-11, 2014

A Brief Overview

1. High-Order Discontinuous Galerkin Discretizations and Implicit Schemes
2. Multigrid Solution Acceleration Strategies
3. Adjoint-Based Mesh Adaptation and Shape Optimization
4. Simulation of Turbulence Using High-Order Discontinuous Galerkin Methods

Outline (Lecture 2)

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

Outline (Lecture 2)

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

Introduction

- Increasing demands for simulation accuracy requires efficient computation algorithms.
  - Error decreasing rate, e^{n+1}/e^{n}, is $1 - \mathcal{O}(h^2)$ for classical iteration techniques like point Jacobi or Gauss-Seidel.

- Multigrid methods have been developed for convergence acceleration.
  - Originally introduced to numerically solve elliptic PDEs
  - Applied to various problems in many disciplines
    - Fluid dynamics and elasticity
    - Geodetics and molecular structures
    - Image reconstruction and tomography
    - Statistical mechanics and etc.
  - An efficient and versatile approach for computational problems

- Basic concept of multigrid methods is to transfer the original problem onto a coarser grid to effectively eliminate low frequency errors.
  - Involving deliberate interpolating procedures between fine and coarse meshes.
  - Similar idea can be applied to the finite element method where various approximation spaces are treated as different "grid" levels.

Outline (Lecture 2)

SimCenter National Center for Computational Engineering

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

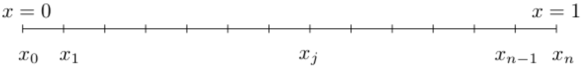
Model Problem

- Consider a one-dimensional boundary-value problem

$$\begin{cases} u''(x) = 0 & 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases}$$

  - Analytical solution for the specific boundary conditions is $u_{ex}(x) = 0$, $x \in [0, 1]$.
  - Present aim is to solve the second-order equation numerically.

- Partition the domain into N subintervals with constant width of $h = 1/N$



- Discretize the second-order term using a central difference scheme

$$u''(x_j) = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + \mathcal{O}(h^2)$$

- The discretized system becomes

$$u_{j+1} - 2u_j + u_{j-1} = 0 \qquad 1 \le j \le N - 1$$

Model Problem

- Express the system ($u_{j+1} - 2u_j + u_{j-1} = 0$) into a matrix form as $Au = b$

$$A = \begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & & \cdots & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \\ \vdots \\ u_{n-1} \end{bmatrix} \quad b = \mathbf{0}$$

- Next we can solve the system using a weighted (or damped) Jacobi iterative method ($1 \leq j \leq N-1$)

$$u_j^* = \frac{u_{j+1}^n + u_{j-1}^n}{2}$$

$$\mathrm{u}_j^{n+1} = (1-\omega)u_j^n + \omega u_j^* = (1-\omega)u_j^n + \tfrac{\omega}{2}(u_{j+1}^n + u_{j-1}^n)$$

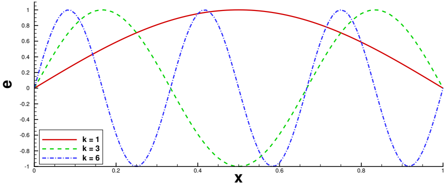
- The convergence behavior of errors can be examined by specifying an initial guess consisting of Fourier modes

$$u_j^0 = sin\left(\frac{jk\pi}{N}\right)$$

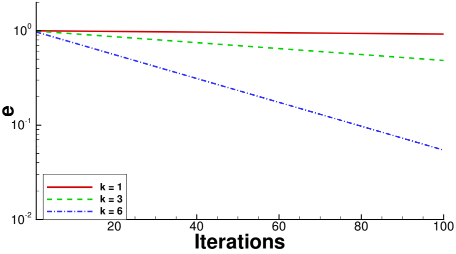
  - k is the wavenumber (or frequency) indicating the number of half sine waves.

Model Problem

- Note that the initial solution error for this problem is $e_j^0 = -sin\left(\frac{jk\pi}{N}\right)$

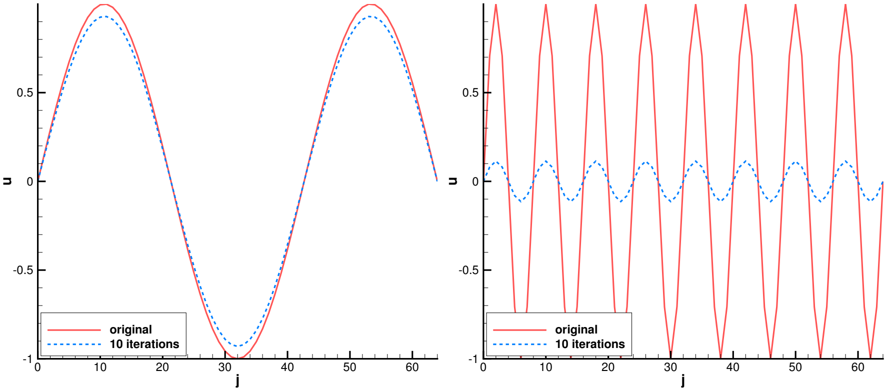


- For any Jacobi iteration, the solution error is $e_j^n = -u_j^n$. Based on this fact, we examine the error convergence by taking 100 weighted Jacobi iterations ($\omega = 2/3$)



  - Higher frequency errors are damped much more rapidly than the lower frequency ones.

- Error damping behavior using the weighted Jacobi method with $\omega = \frac{2}{3}$ for initial guess consisting of $k = 3$ (left) and $k = 16$ (right).



Model Problem

- Examine this behavior more precisely assuming Fourier modes for the error

$$e_j^n = V^n e^{i(j\theta)} \quad e_{j+1}^n = V^n e^{i(j+1)\theta}$$
$$e_j^{n+1} = (1-\omega)V^n e^{i(j\theta)} + \tfrac{\omega V^n}{2}(e^{i(j+1)\theta} + e^{i(j-1)\theta})$$

  - $\theta = k\pi/N$

- The amplification of errors

$$\begin{aligned} g &= \frac{e_j^{n+1}}{e_j^n} = (1-\omega) + \frac{\omega}{2}(e^{i\theta} + e^{-i\theta}) \\ &= (1-\omega) + \omega\cos\theta \\ &= 1 - 2\omega\sin^2(\frac{\theta}{2}) \\ &= 1 - 2\omega\sin^2(\frac{k\pi}{2N}) \qquad 1 \le k \le N-1 \end{aligned}$$

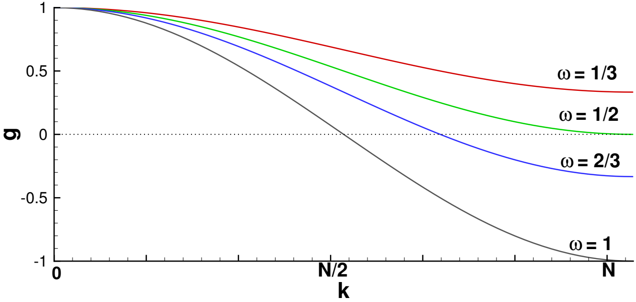
$$\text{or} \quad e_j^{n+1} = g e_j^n$$

- Note that if $|g| < 1$, the errors are damped, and this requires $0 < \omega \le 1$.

Model Problem

SimCenter
National Center for Computational Engineering

- Examine how the value of ω affects the damping of all frequencies



  - All values of ω are not effective to damp low-frequency or smooth components of the error, for example wavenumbers k close to one.
  - For $\omega = 1$ both the high and low-frequency components of the error are damped very slowly, but those near $N/2$ wavenumbers are damped rapidly.
  - $\omega = 2/3$ is effective to damp high-frequency (or oscillatory) components of the error $(N/2 < k < N)$.
- A method is in need to effectively eliminate errors of all frequencies.

- Stem from the idea of using a coarser grid to provide a better initial guess
  - Relaxation is cheaper on the coarse grid.
  - A better convergence rate can be obtained.
  - Recall the amplification of errors:

$$g_k = 1 - 2\omega \sin^2(\frac{k\pi}{2N})$$

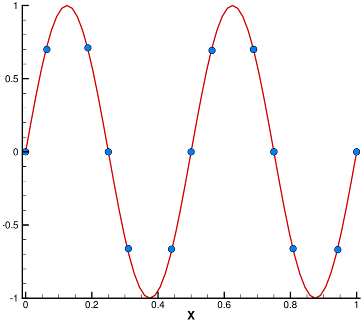
  - g_1 is associated with the smoothest mode ($k = 1$)

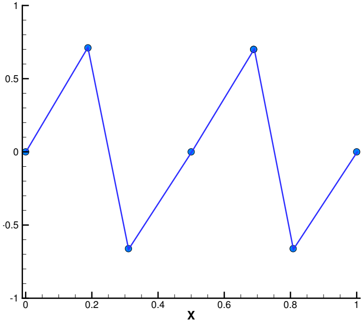
$$\begin{aligned} g_1 &= 1 - 2\omega \sin^2(\frac{\pi}{2N}) \\ &= 1 - 2\omega \sin^2(\frac{\pi h}{2}) \\ &\approx 1 - \mathcal{O}(h^2) \end{aligned}$$

  - Error convergence rate is $-\log_{10}(|g_1|)$.
  - Coarsening the grid by a factor of 2 makes g_1 go from $1 - \mathcal{O}(h^2)$ to $1 - \mathcal{O}(4h^2)$, thus resulting in a larger convergence rate.

Two-Level Multigrid Method

- Low-frequency errors on the fine mesh appear as higher frequencies on a coarser mesh.
  - As an example, use a 4 mode ($k = 4$) wave on a $N = 12$ point mesh projected onto a $N = 6$ point mesh.





  - For the same mode, the wavelength on the fine mesh is $6h$ versus $3h$ on the coarser mesh.
  - The wave on the coarse grid is more oscillatory than that on the fine grid.
- The fine-grid problem should be transferred to a coarser grid to effectively damp the low-frequency errors on the fine grid.

Two-Level Multigrid Method

- Basic idea of multigrid: do enough iterations on the fine grid and transfer the problem to a coarse grid.

- Obtain an equation for the errors that we can transfer to the coarser mesh ⇒ the residual equation.
  - Recall the system of equations

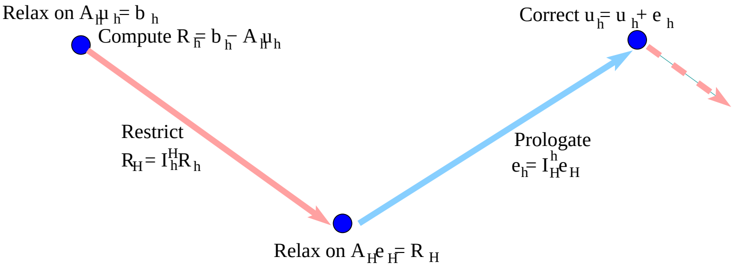
$$\begin{aligned} Au_{ex} &= b \\ Au - b + R &= 0 \qquad \text{where} \quad R = b - Au \\ Ae = R &= b - Au \qquad (e = u_{ex} - u) \end{aligned}$$

  - Relaxation on the original equation $Au_{ex} = b$ with an arbitrary initial guess u^0 is equivalent to relaxing on the residual equation $Ae = R$ with the specific initial guess $e = \mathbf{0}$.

- Based on the idea of relaxation on the error, we can initiate a multigrid approach.

Two-Level Multigrid Method

- A two-level multigrid procedure
  1. Relax ν_1 times on $A_h u_h = b$ on the fine grid to obtain an approximation u_h.
  2. Compute the residual $R_h = b - A_h u_h$ on the fine grid.
  3. Transfer the residual vector to the coarse grid, $R_H = I_h^H R_h$.
  4. Solve the residual equation $A_H e_H = R_H$ on the coarse grid to obtain an approximation to the error e_H.
  5. Interpolate the error on the coarse grid up to the fine grid and update the solution $u_h = u_h + I_H^h e_H$.
  6. Use u_h as initial guess and go back to Step 1.

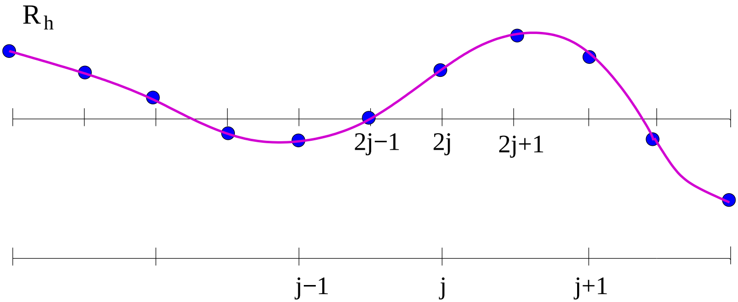


Restriction Procedure

- Restriction procedure

$$R_H = I_h^H R_h$$

  - R_H: residual on mesh with spacing $H = 2h$.
  - R_h: residual on mesh with spacing h.
  - I_h^H: restriction operator

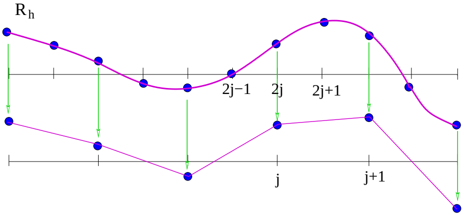


Restriction Procedure

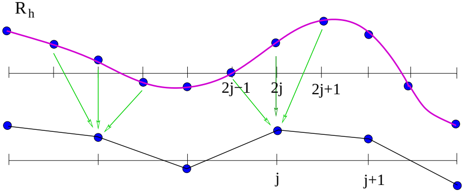
- The restriction operator can be defined in several ways.

$$R_H = I_h^H R_h$$

  - Direct injection, $R_{H_j} = R_{h_{2j}}$



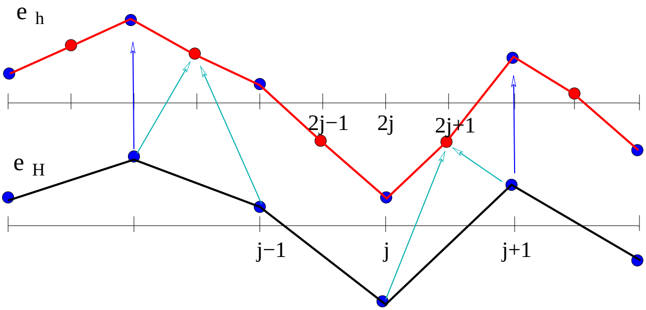
  - Full weighting, $R_{H_j} = \frac{1}{4}(R_{h_{2j-1}} + 2R_{h_{2j}} + R_{h_{2j+1}})$



Prolongation Procedure

- The prolongation is typically done using linear interpolation.

$$e_h = I_H^h e_H$$



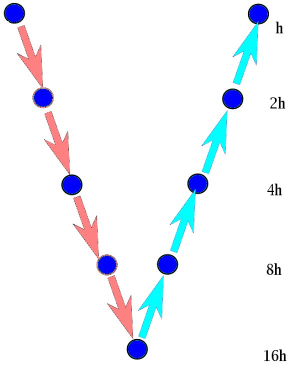
$$e_{h2j} = e_{Hj}$$

$$e_{h2j+1} = \frac{1}{2}(e_{Hj} + e_{Hj+1}) \quad 0 \le j \le \frac{n}{2} - 1$$

- A procedural question: What is the best way to solve the coarse-grid problem?
- In many occasions we do not have to solve the residual equation on the coarse mesh exactly.
- Alternatively we can apply the same multigrid procedure recursively.
- Replace the direct solve in the two-level multigrid scheme by an accurate solve using multiple cycles of multigrid.
- The two-level multigrid procedure is thus performed recursively.
- Typical multigrid cycles: V-Cycle, full Multigrid schemes and etc.

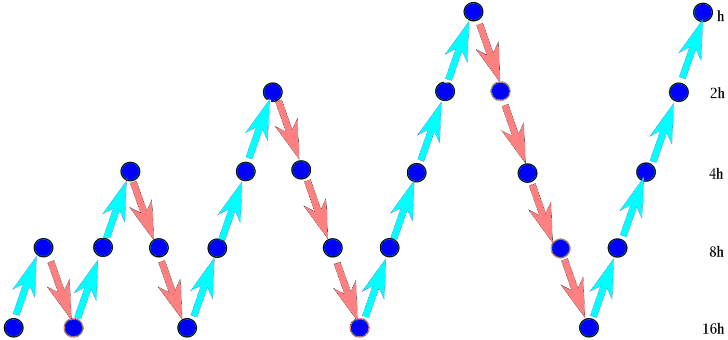
V-Cycle Multigrid Scheme

1. h: Relax v_1 times on $A_h u_h = b$ to obtain u_h.
2. h: Compute $R_h = b - A_h u_h$.
3. $h \rightarrow 2h$ Transfer $R_{2h} = I_h^{2h} R_h$.
4. $2h$: Relax v_1 times on $A_{2h} e_{2h} = R_{2h}$ with initial guess $e_{2h} = 0$
5. $2h$: Compute $R_{2h} = R_{2h} - A_{2h} e_{2h}$.
6. $2h \rightarrow 4h$: Transfer $R_{4h} = I_{2h}^{4h} R_{2h}$.
7. ...
8. $16h$: Solve $A_{16h} e_{16h} = R_{16h}$.
9. ...
10. $4h \rightarrow 2h$: Correct $e_{2h} = e_{2h} + I_{4h}^{2h} e_{4h}$.
11. $2h$: Relax v_2 times on $A_{2h} e_{2h} = R_{2h}$ with initial guess e_{2h}.
12. $2h \rightarrow h$: Correct $u_h = u_h + I_{2h}^{h} e_h$.
13. $2h$: Relax v_2 times on $A_h u_h = R_h$ with initial guess u_h.



Full Multigrid (FMG)

- Start from the relaxation problem at the coarsest grid.
- Each V-cycle is preceded by a coarse-grid V-cycle to obtain a good initial solution.
- Interpolate this initial guess on the current grid.
- Perform a V-cycle to improve the solution.



Outline (Lecture 2)

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

Multigrid Approach for Nonlinear Equations

- Full Approximation Storage (FAS) Scheme
  - A system of nonlinear algebraic equations

$$R(u) = b$$

  - u, b are vectors with dimension N.
  - The notation $R(u)$ denotes a nonlinear operator.

- The nonlinear residual equation on the fine grid can be derived

$$\begin{aligned} R_h(u_h) &= b_h \\ r_h(v_h) &= b_h - R_h(v_h) \end{aligned}$$

  - v_h is an approximation to u_h and $e_h = u_h - v_h$.
  - Subtracting the above two equations from one to the other yields

$$R_h(u_h) - R_h(v_h) = r_h(v_h)$$

- If the high-frequency errors have been previously smoothed, then this equation can be approximated on a coarser mesh as

$$R_H(u_H) = \tilde{I}_h^H r_h + R_H(I_h^H v_h)$$

$$R_H(u_H) = \tilde{I}_h^H r_h + R_H(I_h^H v_h)$$

- $\tilde{I}_h^H$ and I_h^H denote restriction operators for the residual and the solution variables.
- $I_h^H v_h$ serves as an initial approximation to the solution on the coarse mesh.
- u_H is the exact solution on mesh H.
- Full solution is computed and stored on the coarse mesh ⇒ referred to as full approximation storage.
- With the residual equation we can next use a two-level FAS scheme.

Two-Level Full Approximation Storage

1. Relax on the fine grid to obtain an approximation v_h for $R_h(u_h) = b_h$.
2. Compute the residual on the fine grid $r_h(v_h) = b_h - R_h(v_h)$.
3. Restrict the residual and the fine-grid approximation to the coarse grid as $\tilde{I}_h^H r_h$ and $I_h^H v_h$, respectively.
4. Solve the residual equation on the coarse grid $R_H(u_H) = \tilde{I}_h^H r_h + R_H(I_h^H v_h)$.
5. Compute the coarse grid correction $c_H = u_H - I_h^H v_h$.
6. Transfer the correction (i.e. error) to the fine grid, $I_H^h c_H$.
7. Correct v_h on the fine grid using the prolongated correction as $v_h = v_h + I_H^h c_H$ and go to Step 1.

Restriction and Prolongation

- Solution variables are conserved variables, such as $\rho, \rho\mathbf{u}$ and ρE

- To preserve conservativity we use a volume weighted restriction operator for the solution variables.

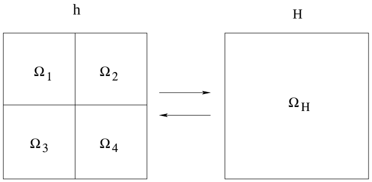
$$I_h^H v_h = \frac{\sum_k \Omega_k v_{hk}}{\sum_k \Omega_k}$$

  - The summation takes over all the fine grid cells which make up the coarse grid cell.

- Restriction operator of the residual is just a summation as

$$\tilde{I}_h^H R_h = \sum_k R_{hk}$$

- Prolongation operator of the correction is linear interpolation.

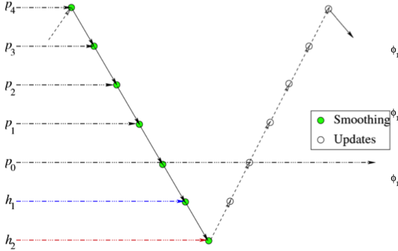


Outline (Lecture 2)

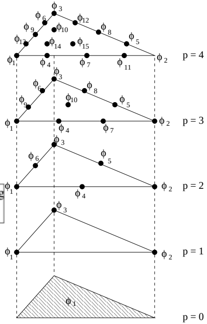
- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

hp−Multigrid Approach

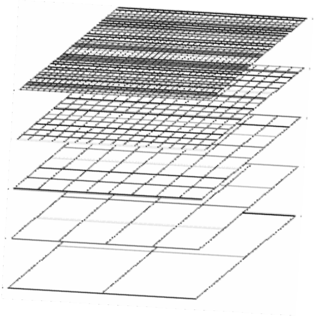
- Combination of spectral and geometric multigrid schemes (*hp*−Multigrid)
- The spectral multigrid (*p*−Multigrid) approach makes use of lower-order *p*−levels as coarser grids.
  - Alleviate the need to generate a sequence of agglomerated grids.
  - Simplify the interpolation and prolongation procedures with hierarchical functions.
  - Often more suitable for unsteady problems.



hp-Multigrid V-Cycle



Multigrid in various *p*-levels



Multigrid in various *h*-levels

hp–Multigrid Approach

SimCenter NATIONAL CENTER for COMPUTATIONAL ENGINEERING

- The same procedure in the traditional multigrid approach is applied: fine grid problem is accelerated by means of coarser grid corrections.

$$\begin{aligned} \text{Fine}: & \quad R_p(u_p) = b_p \\ \text{Coarse}: & \quad R_{p-1}(u_{p-1}) = \tilde{I}_p^{p-1} r_p + R_{p-1}(I_p^{p-1} v_p) \end{aligned}$$

- The use of the hierarchical basis functions greatly facilitates the processes of restriction and prolongation in the p–multigrid scheme.

$$\begin{array}{lllllllllll} p_1 \cdots & \phi_1 & \phi_2 & \phi_3 & & & & & & & \\ p_2 \cdots & \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & & & & \\ p_3 \cdots & \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 & \phi_9 & \phi_{10} \end{array}$$

  - Approximation spaces are nested, i.e. $\mathcal{V}_h^{p-1} \subset \mathcal{V}_h^p$

$$\phi_i^{p-1} = \sum_j \alpha_{ij}^{p-1} \phi_j^p$$

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{bmatrix}$$

- Levels between p and $p-1$ $(p-1>0)$ $\qquad \phi_i^{p-1} = \sum_j \alpha_{ij}^{p-1}\phi_j^p$
- Weighted residual restriction $\tilde{I}_p^{p-1} r_p$ and state variable restriction $I_p^{p-1} v_p$

$$\tilde{I}_p^{p-1} r_p = R(\phi_i^{p-1}, v_p) = R\left(\sum_j \alpha_{ij}^{p-1}\phi_j^p, v_p\right) = \sum_j \alpha_{ij}^{p-1} R(\phi_j^p, v_p)$$

$$I_p^{p-1} v_p = \alpha_{ij}^{p-1} v_p$$

  - Obtained by disregarding the higher order modes and transferring the values of the low order modes exactly.

- State variable prolongation $c_p = I_{p-1}^p c_{p-1}$

$$I_{p-1}^p = \left(\alpha_{ij}^{p-1}\right)^T \qquad \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

  - Obtained by setting the high order modes to zero and injecting the values of the low order coefficients exactly.

$hp-$Multigrid Approach

- Restriction and prolongation between p_1 and p_0 levels
  - Basis function is constant for $p = 0$, $\phi^{p_0} = 1$.
  - Underlying grid is the same.
  - Triangular mesh as an example:

$$I_{p_1}^{p_0} v_{p_1} = \frac{1}{3} \sum_{i=1}^{3} v_{p_1,i} \qquad I_{p_0}^{p_1} c_{p_0} = c_{p_0}$$

$$\tilde{I}_{p_1}^{p_0} R_{p_1} = \sum_{i=1}^{3} R_{p_1,i}$$

- Restriction and prolongation between h-levels operate the same manners as those for the traditional multigrid schemes.

$$I_h^H v_h = \frac{\sum_k \Omega_k v_{hk}}{\sum_k \Omega_k} \qquad \tilde{I}_h^H R_h = \sum_k R_{hk} \qquad I_H^h c_H = c_H$$

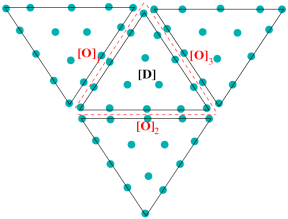
Relaxation Methods at Each Level

- Approximate Newton method for $R(u) = S$

$$\left[\frac{\partial R}{\partial u}\right]^n \Delta u^{n+1} = S - R(u^n)$$

$$u^{n+1} = u^n + \omega \Delta u^{n+1}$$

  - Decompose the Jacobian matrix as $\left[\frac{\partial R}{\partial u}\right]^n = [D^n] + [O^n]$



- Various relaxation/smoothing solvers
  - Nonlinear element Jacobi

$$\Delta u^{n+1} = [D^n]^{-1} \left(S - R(u^n)\right)$$

  - *Quasi* nonlinear element Jacobi (runs with sub-iterations, k)

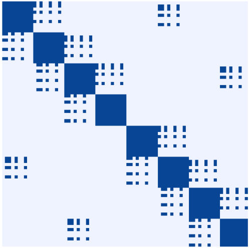
$$\Delta u^{k+1} = [D^n]^{-1} \left(S - R(u^k)\right)$$

  - Linearized element Jacobi

$$\Delta u^{k+1} = [D^n]^{-1} \left(S - R(u^n) - [O^n]\Delta u^k\right)$$

  - Linearized element Gauss-Seidel $[O^n] = [L^n] + [U^n]$

$$\Delta u^{k+1} = [(D + L)^n]^{-1} \left(S - R(u^n) - [U^n]\Delta u^k\right)$$



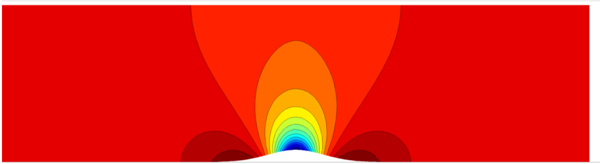
Outline (Lecture 2)

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

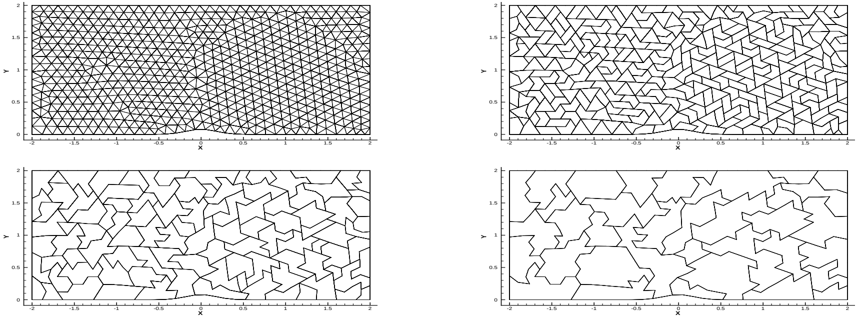
- Compressible channel flow over a Gaussian bump
- Convection of an isentropic vortex

Compressible Channel Flow over a Gaussian Bump

- $M_\infty = 0.2$ (steady-state problem)
- Inflow/Outflow boundary conditions and wall boundary conditions enforced on the top and bottom

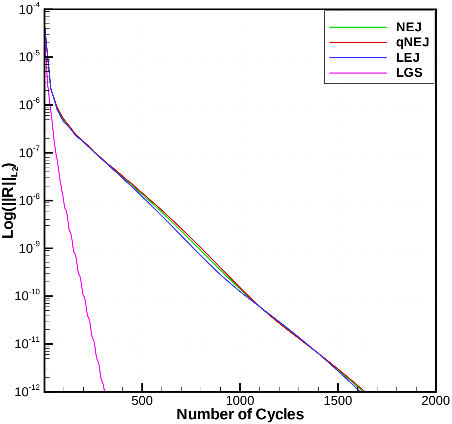


- Agglomerated coarser grids

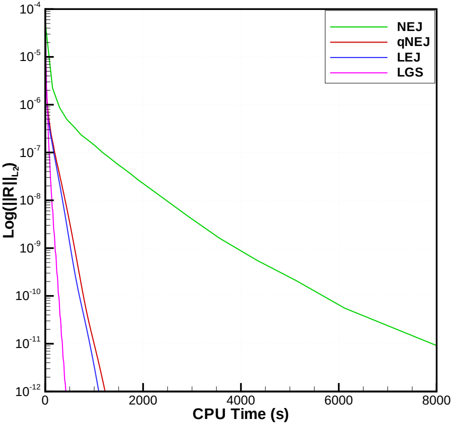


Compressible Channel Flow over a Gaussian Bump

- Comparison of convergence of non-linear element Jacobi (NEJ), *quasi*-nonlinear element Jacobi (qNEJ), linearized element Jacobi (LEJ) and linearized Gauss-Seidel (LGS) smoothers
- Mesh size $N = 1248$, DG $p = 4$ (i.e. fifth-order) scheme, 5 sub-iterations



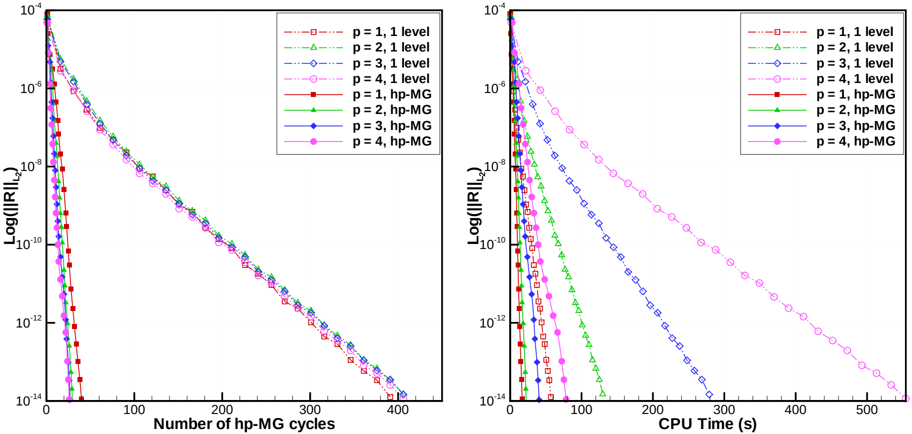
L_2-norm of residual vs. MG cycles



L_2-norm of residual vs. CPU time

Compressible Channel Flow over a Gaussian Bump

- Effect of various discretization orders on the solution convergence
- Single level method versus *hp*−multigrid approach
- Discretization orders vary from $p = 1$ to $p = 4$

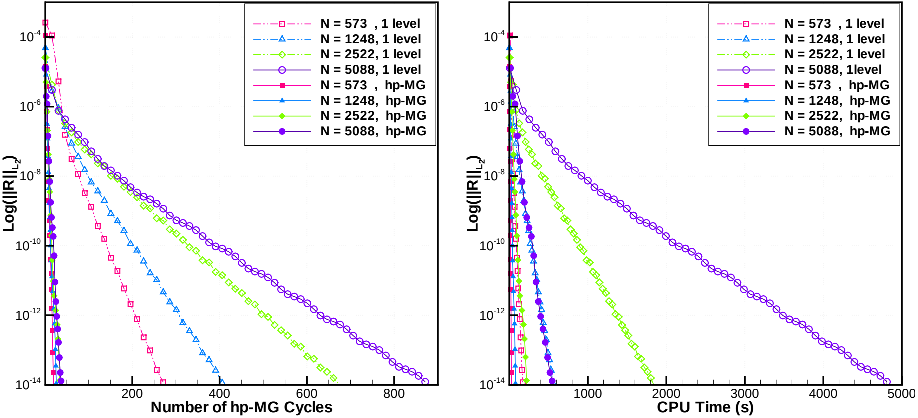


L_2-norm of residual vs. MG cycles

L_2-norm of residual vs. CPU time

Compressible Channel Flow over a Gaussian Bump

- Effect of mesh resolution on the solution convergence
- Single level method versus *hp*−multigrid approach
- Variation of mesh sizes $N = 573$, $N = 1248$, $N = 2522$ and $N = 5088$ (fixed $p = 4$)



L_2-norm of residual vs. MG cycles L_2-norm of residual vs. CPU time

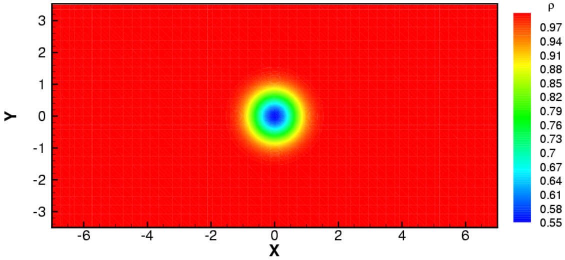
Convection of An Isentropic Vortex

- Effects of mesh sizes and time-step sizes on the solution convergence
- Uniform flow perturbed by an isentropic vortex

$$\delta u = -\frac{\sigma}{2\pi}(y - y_0)e^{\vartheta(1-r^2)}$$

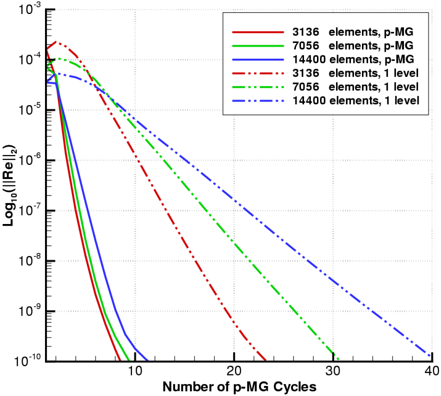
$$\delta v = \frac{\sigma}{2\pi}(x - x_0)e^{\vartheta(1-r^2)}$$

$$\delta T = -\frac{\sigma^2(\gamma - 1)}{16\vartheta\gamma\pi^2}e^{2\vartheta(1-r^2)}$$

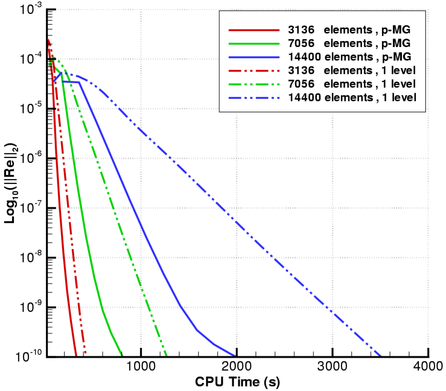


Convection of An Isentropic Vortex

- Effect of mesh sizes on the solution convergence
- Single level versus $p-$multigrid solvers
- DG $p = 4$ scheme and the BDF2 temporal scheme (fixed time-step size $\Delta t = 1.0$)
- Various mesh sizes $N = 3136$, $N = 7056$ and $N = 14400$



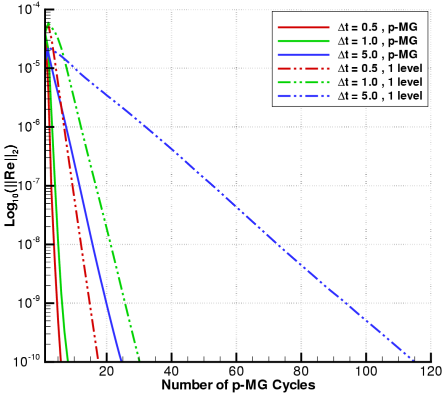
Convergence history vs. p-Multigrid cycles



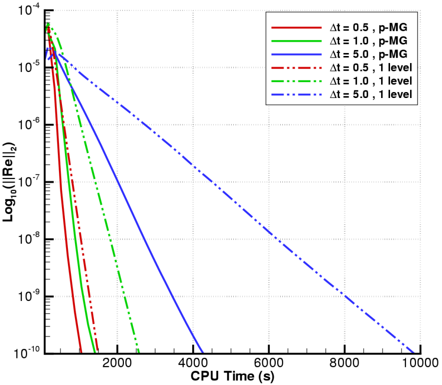
Convergence history vs. CPU time

Convection of An Isentropic Vortex

- Effect of time-step sizes on the solution convergence
- Single level versus p−multigrid solvers
- DG $p = 4$ scheme, the BDF2 temporal scheme and fixed mesh size $N = 14400$
- Various time-step sizes $\Delta t = 0.5, 1.0$ and 5.0



Convergence history vs. p-MG cycles



Convergence history vs. CPU time

Outline (Lecture 2)

SimCenter National Center for Computational Engineering

- Introduction and Basic Concepts
- Model Problem and Two-Level Multigrid Approach
- Multigrid Approach for Nonlinear Equations
- *hp*−Multigrid Strategy
- Numerical Examples
- Conclusions

Conclusions

- The multigrid method is designed to eliminate low-frequency errors on the fine mesh by transferring the fine-grid residual to a coarse grid.
- Purely spectral ($p-$) multigrid approach operates on the approximation spaces of different orders.
- The coupling of spectral and agglomerated ($hp-$) multigrid procedures increases the overall efficiency for steady-state problems, while the purely $p-$multigrid approach is more appropriate for implicit time-integration problems.
- Compared to the nonlinear Jacobi smoother, the linearized smoothers require additional storage, but generally more efficient than the former nonlinear smoother.
- The $hp-$multigrid schemes demonstrates both $h-$ and $p-$independent convergence rates, thus the efficiency benefits become more significant for finer meshes.
- For implicit time-integration problems, the $p-$multigrid strategy exhibits $h-$independent convergence rates while retaining slight dependence on time-step sizes.

References

W. Briggs, V. E. Henson and S. F. McCormick, A Multigrid Tutorial: Second Edition, ISBN: 0-89871-462-1, Society for Industrial and Applied Mathematics, 2000.

P. Wesseling, Introduction to Multigrid Methods, ICASE Report No. 95-11, 1995.

C. R. Nastase, D. J. Mavriplis, High-order Discontinuous Galerkin Methods Using an hp-multigrid Approach, Journal of Computational Physics, 213 (1), 330357, 2006.

L. Wang, Techniques for High-Order Adaptive Discontinuous Galerkin Discretizations in Fluid Dynamics, PhD dissertation, University of Wyoming, 2009.

K. J. Fidkowski, T. A. Oliver, J. Lu, D. Darmofal, p-multigrid Solution of High-Order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations, Journal Computational Physics, 207, 92113, 2005.

L. Wang, and D.J. Mavriplis, Implicit Solution of the Unsteady Euler Equations for High-order Accurate Discontinuous Galerkin Discretizations, Journal of Computational Physics, 225 (2), pp. 1994-2015, 2007.