HIGH-ORDER LINEAR AND NON-LINEAR RESIDUAL DISTRIBUTION SCHEMES FOR THE SIMULATION OF COMPRESSIBLE VISCOUS FLOWS

Dante De Santis

Stanford University Aeronautics and Astronautics

AMS Seminar
March 26, 2015
BACKGROUND & MOTIVATION

CFD as a common tool in engineering

TODAY INDUSTRIAL PRACTICE

- FV schemes (theoretical) 2nd order accuracy
- Not accurate enough for realistic applications
- Very fine meshes required: CPU expensive calculations
- Simulations of large problems still expensive for fast design & optimization
BACKGROUND & MOTIVATION

CFD AS A COMMON TOOL IN ENGINEERING

TODAY INDUSTRIAL PRACTICE

- FV schemes (theoretical) 2nd order accuracy
- Not accurate enough for realistic applications
- Very fine meshes required: CPU expensive calculations
- Simulations of large problems still expensive for fast design & optimization

APPEAL OF HIGH-ORDER METHODS

- Faster reduction of the discretization error with the DoFs
- Accurate solutions at acceptable costs
- High-fidelity simulations make possible to check deficits of physical modeling
**BACKGROUND & MOTIVATION**

**HO methods for real-life applications**

- Several high-order schemes developed
  - ENO/WENO
  - Continuous finite elements
  - Discontinuous Galerkin
  - Spectral volume, Spectral difference, Flux reconstruction, ...
**Background & Motivation**

**HO methods for real-life applications**

- Several high-order schemes developed
  - ENO/WENO
  - Continuous finite elements
  - Discontinuous Galerkin
  - Spectral volume, Spectral difference, Flux reconstruction, ...

**Active Research Field**

- Reduce the computational costs
- Reliable discretization of discontinuous solutions
- **RD: A possible solution to these limitations**
Presentation outline

1. RD Advection-Diffusion Problems
2. Discretization of NS equations
3. Discretization of RANS equations
4. Conclusions and Perspectives
Residual distribution schemes
The main idea for steady problems

- Solve $\nabla \cdot f(u) = 0$, on $\Omega \subset \mathbb{R}^d$
Residual distribution schemes
The main idea for steady problems

- Solve $\nabla \cdot f(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of $k$-th order: $u \simeq u_h(x)$

$$u_h = \sum_{i \in N_h} \psi_i(x) u_i,$$
Residual distribution schemes

The main idea for steady problems

- Solve $\nabla \cdot f(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of $k$-th order: $u \simeq u_h(x)$
- The approximated solution will give birth to a residual: $\Phi_e(u_h)$

$$\Phi_e(u_h) = \int_{\Omega_e} \nabla \cdot f(u_h) \, d\Omega = \int_{\partial \Omega_e} f(u_h) \cdot \hat{n} \, d\Omega$$
Residual distribution schemes
The main idea for steady problems

- Solve $\nabla \cdot f(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of $k$-th order: $u \simeq u_h(x)$
- The approximated solution will give birth to a residual: $\Phi^e(u_h)$
- Distribution to the DoFs of the element residual

$$\Phi^e_i = \beta^e_i \Phi^e$$

$$\sum_{i \in N^e_h} \Phi^e_i = \Phi^e, \text{ for conservation}$$
Residual distribution schemes

The main idea for steady problems

- Solve $\nabla \cdot f(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of $k$-th order: $u \simeq u_h(x)$
- The approximated solution will give birth to a residual: $\Phi^e(u_h)$
- Distribution to the DoFs of the element residual

$$\Phi^e_i = \beta^e_i \Phi^e$$

$$\sum_{i \in N^e_h} \Phi^e_i = \Phi^e, \text{ for conservation}$$

- Gather the residuals: $\sum_{i \in N^i_h} \Phi^e_i = 0, \forall i$

Change of the solution driven by non-zero element residual

$$\frac{u_i^{n+1} - u_i^n}{\Delta t^n} + \sum_{i \in N^i_h} \Phi^e_i = 0, \ n \to \infty$$
Residual Distributions schemes
Some approach for advection-diffusion problems

- Well established methodology for advection problems
- What about the discretization of advection-diffusion problems?

\[ \nabla \cdot f(u) = \nabla \cdot (\nu \nabla u), \quad \text{Pe} = \|a\| h/\nu \]
Residual Distributions schemes
Some approach for advection-diffusion problems

- Well established methodology for advection problems
- What about the discretization of advection-diffusion problems?

$$\nabla \cdot f(u) = \nabla \cdot (\nu \nabla u), \quad Pe = ||a|| h/\nu$$

- Old approach: mixed RD (for advection) and Galerkin (for diffusion)
  - Error analysis reveals that the approach is 1st order accurate when $Pe \sim 1$
  - Proper scaling of the RD upwind stabilization with $Pe$ (Ricchiuto et al.)

- New principle [Roe, Nishikawa, Caraeni, ...] one distribution process for the residual of whole equation (advection+diffusion) to get an uniform order of accuracy on entire spectrum of $Pe$
Discretization of Advection-Diffusion problems

Calculation of the total residual

1. Total residual of the advection-diffusion problem

\[ \Phi^e = \int_{\Omega_e} \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu \nabla u_h) \right) \, d\Omega. \]
Discretization of Advection-Diffusion problems

Calculation of the total residual

- Total residual of the advection-diffusion problem

\[ \Phi^e = \int_{\Omega_e} \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu \nabla u_h) \right) d\Omega. \]

- Using the divergence theorem

\[ \Phi^e = \oint_{\partial \Omega_e} \left( f(u_h) - \nu \nabla \bar{u}_h \right) \cdot \hat{n} \, d\partial \Omega, \]

- For a piece-wise polynomial interpolation, \( \nabla u^h \cdot n \) is discontinuous at the elements face. The scheme requires a continuous gradient \( \bar{\nabla} u^h \).
RD: Distributions process
Central linear schemes (inspired by Lax-Wendroff)

- Originally proposed for multidimensional upwinding
  - Roe, Deconinck, Abgrall ...
  - Formulation on simplexes & difficult extension to HO
- Central schemes: HO & general elements
RD: Distributions process
Central linear schemes (inspired by Lax-Wendroff)

- Originally proposed for multidimensional upwinding
  - Roe, Deconinck, Abgrall ...
  - Formulation on simplexes & difficult extension to HO
- **Central schemes**: HO & general elements

**Linear-scheme**

\[
\Phi_i^{e, LW} = \frac{\Phi^e}{N^e_{dof}} + \int_{\Omega_e} a \cdot \nabla \psi_i \tau \left( a \cdot \nabla u_h - \nabla \cdot (\nu \nabla \bar{u}^h) \right) d\Omega 
\]

\[
\tau = \frac{1}{2} \sum_{j \in N^e_h} \max(k_j, 0) + \nu, \quad \text{with} \quad k_j = \frac{1}{2} \bar{a} \cdot \bar{n}_j,
\]

- Reconstructed gradient used in the stabilization term
- High-order preserving: \( u_h \in P^k \Rightarrow \) scheme \( O(h^{k+1}) \) (always?)
- Linear scheme: not monotone on shocks
RD: Distributions process

Construction of a non-linear RD scheme

- First order monotone scheme, e.g. Rusanov's scheme

\[
\Phi_i^e = \frac{\Phi^e}{N_{dof}} + \alpha \sum_{j \in N_h^e} (u_i - u_j), \quad \alpha > 0
\]
**RD: DISTRIBUTIONS PROCESS**

**CONSTRUCTION OF A NON-LINEAR RD SCHEME**

- First order monotone scheme, e.g. Rusanov's scheme

\[
\Phi_i^e = \frac{\Phi^e}{N^e_{dof}} + \alpha \sum_{j \in N^e_h} (u_i - u_j), \quad \alpha > 0
\]

- \(\beta_i^e = \frac{\Phi_i^e}{\Phi^e}\) unbounded. Apply the limiting map \(\beta_i^e \mapsto \hat{\beta}_i^e(u_h)\)

\[
\hat{\beta}_i^e(u_h) = \frac{\text{max}(\beta_i^e, 0)}{\sum_{j \in N^e_h} \text{max}(\beta_j^e, 0)} \quad \Rightarrow \quad \hat{\beta}_i^e \in [0, 1] \quad \& \quad \sum_{i \in N^e_h} \hat{\beta}_i^e = 1
\]
**RD: Distributions process**

**Construction of a non-linear RD scheme**

- First order monotone scheme, e.g. Rusanov’s scheme

\[
\Phi_i^e = \frac{\Phi^e}{N_{dof}^e} + \alpha \sum_{j \in N_h^e} (u_i - u_j), \quad \alpha > 0
\]

- \( \beta_i^e = \frac{\Phi_i^e}{\Phi_e} \) unbounded. Apply the limiting map \( \beta_i^e \mapsto \hat{\beta}_i^e(u_h) \)

\[
\hat{\beta}_i^e(u_h) = \frac{\max(\beta_i^e, 0)}{\sum_{j \in N_h^e} \max(\beta_j^e, 0)} \quad \Rightarrow \quad \hat{\beta}_i^e \in [0, 1] \quad \& \quad \sum_{i \in N_h^e} \hat{\beta}_i^e = 1
\]

- Compute the high order distributed residual: \( \hat{\Phi}_i^e = \hat{\beta}_i^e(u_h^h)\Phi^e \)
RD: Distributions process
Construction of a non-linear RD scheme

- First order monotone scheme, e.g. Rusanov’s scheme
  \[ \Phi_i^e = \frac{\Phi_e}{N_{dof}^e} + \alpha \sum_{j \in N_h^e} (u_i - u_j), \quad \alpha > 0 \]

- \( \beta_i^e = \frac{\Phi_i^e}{\Phi^e} \) unbounded. Apply the limiting map \( \beta_i^e \mapsto \hat{\beta}_i^e(u_h) \)
  \[ \hat{\beta}_i^e(u_h) = \frac{\max(\beta_i^e, 0)}{\sum_{j \in N_h^e} \max(\beta_j^e, 0)} \Rightarrow \hat{\beta}_i^e \in [0, 1] \quad \& \quad \sum_{i \in N_h^e} \hat{\beta}_i^e = 1 \]

- Compute the high order distributed residual: \( \hat{\Phi}_i^e = \hat{\beta}_i^e(u_h^h)\Phi_e^e \)
- Limiting enforces monotonicity but no upwinding included
- The solution consists in adding a filtering term
  \[ \hat{\Phi}_i^e = \hat{\beta}_i^e(u_h^h)\Phi_e^e + \epsilon_h \int_{\Omega_e} \left( a \cdot \nabla \psi_i - \nabla \cdot (\nu \nabla \psi_i) \right) \tau \left( a \cdot \nabla u_h^h - \nabla \cdot (\nu \nabla u_h^h) \right) d\Omega \]
SOMETHING IS STILL MISSING
IMPROVEMENT OF VISCOUS TERM DISCRETIZATION. INSPIRED BY H. NISHIKAWA

- Improve the discretization of diffusion dominated problems
SOMETHING IS STILL MISSING

IMPROVEMENT OF VISCOSITY TERM DISCRETIZATION. INSPIRED BY H. NISHIKAWA

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system

\[
\begin{cases}
\nabla \cdot f(u) - \nabla \cdot (\nu q) = 0 \\
q - \nabla u = 0
\end{cases}
\]
Something is still missing
Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system

\[
\begin{align*}
\nabla \cdot f(u) - \nabla \cdot (\nu q) &= 0 \\
q - \nabla u &= 0
\end{align*}
\]

- Discretize the FOS with a central scheme + a streamline stabilization

\[
\int_{\Omega_e} \psi_i \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) + \int_{\Omega_e} A \cdot \nabla \psi_i \tau \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) = 0
\]
Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system
  \[
  \begin{align*}
  \nabla \cdot f(u) - \nabla \cdot (\nu q) &= 0 \\
  q - \nabla u &= 0
  \end{align*}
  \]
- Discretize the FOS with a central scheme + a streamline stabilization
  \[
  \int_{\Omega_e} \psi_i \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) + \int_{\Omega_e} A \cdot \nabla \psi_i \tau \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) = 0
  \]
- Replace \( q_h \) with \( \nabla u_h \), the second line is discarded and the problem reads
  \[
  \int_{\Omega_e} \psi_i \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu \nabla u_h) \right) + \int_{\Omega_e} a \cdot \nabla \psi_i \tau_a \left( a \cdot \nabla u_h - \nabla \cdot (\nu \nabla u_h) \right) + \int_{\Omega_e} \nu \nabla \psi_i \cdot (\tau_d \left( \nabla u_h - \nabla u_h \right)) = 0
  \]
**Final form of the RD space discretization**

**Linear scheme**

\[
\Phi_{i}^{e,\text{LW}} = \frac{\Phi_{i}^{e}}{N_{\text{dof}}} + \int_{\Omega_{e}} a \cdot \nabla \psi_{i} \tau \left( a \cdot \nabla u_{h} - \nabla \cdot (\nu \nabla u_{h}) \right) d\Omega \\
+ \int_{\Omega_{e}} \nu \nabla \psi_{i} \cdot \left( \nabla u_{h} - \nabla \tilde{u}_{h} \right) d\Omega,
\]

**Non-linear scheme**

\[
\hat{\Phi}_{i}^{e,\text{Rv}} = \hat{\beta}_{i}^{e,\text{Rv}}(u_{h}) \Phi_{i}^{e}(u_{h}) \\
+ \epsilon_{h}^{e}(u_{h}) \int_{\Omega_{e}} \left( a \cdot \nabla \psi_{i} - \nabla \cdot (\nu \nabla \psi_{i}) \right) \tau \left( a \cdot \nabla u_{h} - \nabla \cdot (\nu \nabla u_{h}) \right) d\Omega \\
+ \int_{\Omega_{e}} \nu \nabla \psi_{i} \cdot \left( \nabla u_{h} - \nabla \tilde{u}_{h} \right) d\Omega,
\]
The problem of the gradient reconstruction

- The “internal” gradient of the solution is replaced by a continuous approximation
  \[ \nabla u_h \rightarrow \nabla \hat{u}_h = \sum_{i \in N^e_h} \psi_i \nabla \hat{u}_i \]

- \( \nabla \hat{u}_i \) is the reconstructed gradient of the solution at the DoF \( i \)
THE PROBLEM OF THE GRADIENT RECONSTRUCTION

- The “internal” gradient of the solution is replaced by a continuous approximation
  \[ \nabla u_h \longrightarrow \tilde{\nabla} u_h = \sum_{i \in \mathcal{N}_h^e} \psi_i \tilde{\nabla} u_i \]

- \( \tilde{\nabla} u_i \) is the reconstructed gradient of the solution at the DoF \( i \)

- High-order preserving scheme
  - With \( u_h \in P^k \Rightarrow (k+1) \)-th accurate scheme for advection problems
  - How accurate the gradient reconstruction must be for advection-diffusion problems? As accurate as the solution
The problem of the gradient reconstruction

The “internal” gradient of the solution is replaced by a continuous approximation

\[ \nabla u_h \rightarrow \widetilde{\nabla} u_h = \sum_{i \in \mathcal{N}_h^e} \psi_i \widetilde{\nabla} u_i \]

\( \widetilde{\nabla} u_i \) is the reconstructed gradient of the solution at the DoF \( i \)

High-order preserving scheme

- With \( u_h \in \mathbb{P}^k \Rightarrow (k+1) \)-th accurate scheme for advection problems
- How accurate the gradient reconstruction must be for advection-diffusion problems? As accurate as the solution

Classical gradient reconstruction methods

- Green-Gauss (area-weighted)
- \( L^2 \) projection
- Least-square

With these approaches \( \widetilde{\nabla} u_h = O(h^k) \). One order less than the solution
Super-convergent Patch Recovery (Zienkiewicz & Zhu, 92)

The main idea

- Idea coming from mechanical structure: compute stresses with the same accuracy of the displacements. Main idea: use it for CFD
- Gradients at certain points is more accurate than in others
- For structured grids theory identifies Gauss-Legendre points. No formal theory for unstructured grids
- Polynomial interpolation of degree $k$: with a least square fitting to the sampled HO values within a patch of elements.
Super-convergent Patch Recovery method

How it works

- For each vertex \( i \) of the grid, the components of the reconstructed gradient are written in polynomial form

\[
\frac{\partial u^h}{\partial x} \bigg|_i = p^T a_x, \quad \text{where} \quad p^T = (1, x, y, x^2, \ldots, y^k)
\]

\[
a_x = (a_{x_1}, a_{x_2}, \ldots, a_{x_m})
\]
Super-convergent Patch Recovery method

How it works

- For each vertex $i$ of the grid, the components of the reconstructed gradient are written in polynomial form

$$\frac{\partial u^h}{\partial x} \bigg|_i = p^T a_x, \quad \text{where} \quad p^T = (1, x, y, x^2, \ldots, y^k)$$
$$a_x = (a_{x_1}, a_{x_2}, \ldots, a_{x_m})$$

- Minimize respect to $a_x$ the function

$$F_x = \sum_{j=1}^{N_s} \left( \frac{\partial u^h}{\partial x}(x_j) - p(x_j)^T a_x \right)^2$$
**Super-convergent Patch Recovery method**

**How it works**

- For each vertex \( i \) of the grid, the components of the of the reconstructed gradient are written in polynomial form

\[
\frac{\partial u^h}{\partial x} \bigg|_i = p^T a_x, \quad \text{where} \quad p^T = (1, x, y, x^2, \ldots, y^k) \\
a_x = (a_{x1}, a_{x2}, \ldots, a_{xm})
\]

- Minimize respect to \( a_x \) the function

\[
F_x = \sum_{j=1}^{N_s} \left( \frac{\partial u^h}{\partial x}(x_j) - p(x_j)^T a_x \right)^2
\]

- Solve (in a least square sense) the linear system

\[
A a_x = b_x^h, \quad \text{for} \quad a_x
\]

\[
A = \begin{pmatrix}
1 & x_1 & y_1 & \cdots & y_{1}^k \\
1 & x_2 & y_2 & \cdots & y_{2}^k \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{N_s} & y_{N_s} & \cdots & y_{N_s}^k
\end{pmatrix}, \quad b_x^h = \begin{pmatrix}
\partial u^h / \partial x(x_1) \\
\partial u^h / \partial x(x_2) \\
\vdots \\
\partial u^h / \partial x(x_{N_s})
\end{pmatrix}
\]
ZZ Super-convergent Patch Recovery method

**Important Remarks**

- The method is very flexible: 2D/3D, hybrid grids
- $A \in \mathbb{R}^{N_s \times m}$, $N_s \geq m$ ($N_s$ sampling points, $m$ polynomial coefficients)
- On the boundary use the interior patch of the nearest vertices
- How to handle high order elements?

![Diagram](image)

Gradient is reconstructed by evaluating on the patch, at the coordinates of the nodes, the polynomial function constructed for the nearest grid vertex
**ZZ Super-convergent Patch Recovery method**

**Accuracy test**

Gradient recovery of a smooth solution on unstructured grids
**ZZ Super-convergent Patch Recovery method**

**Accuracy test: triangular grids**

ZZ-SPR gives gradients with the same order of accuracy of the solution.
ZZ Super-convergent Patch Recovery method

Accuracy test: hybrid grids

ZZ-SPR gives gradients with the same order of accuracy of the solution
Solve: $\mathbf{a} \cdot \nabla u = \nu \nabla \cdot \nabla u$, with $\nu = 0.01$ ($\text{Pe} \sim 1$)

Linear scheme: effect of the gradient reconstruction

- Triangular grids and $\mathbb{P}^1$ elements

2nd order for solution and gradient with all gradient reconstruction
Solve: \( \mathbf{a} \cdot \nabla u = \nu \nabla \cdot \nabla u, \) with \( \nu = 0.01 \) (Pe \( \sim 1 \))

**Linear scheme: effect of the gradient reconstruction**

- Triangular grids and \( \mathbb{P}^2 \) elements

\[
\text{error } u_h
\]

\[
\text{error } \tilde{\nabla} u_h
\]

3rd order for solution and gradient only with ZPR-ZZ gradient reconstruction
Solve: \( a \cdot \nabla u = \nu \nabla \cdot \nabla u \), with \( \nu = 0.01 \) (Pe \( \sim 1 \))

Linear scheme: Benefit of high-order approximation

For \( u^h \) error \( \simeq 10^{-5} \)
- \( P^2 \): \( N_{\text{dof}} \simeq 12000 \), and CPU time \( \simeq 25\text{min} \)
- \( P^1 \): \( N_{\text{dof}} \simeq 31000 \), and CPU time \( \simeq 5\text{h} \)
**DISCONTINUOUS SOLUTION: LINEAR AND NON-LINEAR SCHEMES**

$$\mathbf{a} \cdot \nabla u = \nu \nabla \cdot \nabla u, \quad \mathbf{a} = (1/2, \sqrt{3}/2)^T, \quad \nu = 10^{-3}$$

- Linear scheme $\mathbb{P}_1$
- Linear scheme $\mathbb{P}_2$
- Non-linear scheme $\mathbb{P}_1$
- Non-linear scheme $\mathbb{P}_2$
Governing Equations
Compressible Navier-Stokes Equations

\[ \nabla \cdot \mathbf{f}^a(u) - \nabla \cdot \mathbf{f}^v(u, \nabla u) = 0 \]

\[ u = \begin{pmatrix} \rho \\ m \\ E^t \end{pmatrix}, \quad \mathbf{f}^a = \begin{pmatrix} m \\ \frac{m \otimes m}{\rho} + P^I \\ \frac{m}{E^t + P} \end{pmatrix}, \quad \mathbf{f}^v = \begin{pmatrix} 0 \\ \mathbb{S} \\ \mathbb{S} \cdot \frac{m}{\rho} + \kappa \nabla T \end{pmatrix} \]

- Viscous flux function homogeneous with the respect to the gradient of the conservative variables: \( \mathbf{f}^v(u, \nabla u) = \mathbb{K}(u) \nabla u \)
- Straightforward extension of the numerical schemes to system of equations
- Additional features respect to scalar equations
  - Boundary conditions
  - Implicit scheme
**Boundary conditions**

**Boundary representation**

- Imposition of the boundary conditions

\[
\sum_{e \in \mathcal{E}_{h,i}} \Phi^e_i + \sum_{f \in \mathcal{F}_{h,i}} \Phi^{e,\partial}_i = 0, \quad \forall i \in \mathcal{N}_h,
\]

- High-order schemes requires high-order boundary representation
- Here isoparametric formulation used. Same order for solution and geometry
- Piecewise polynomial approximation of the geometry
**Boundary conditions**

**Boundary representation**

- Imposition of the boundary conditions

\[
\sum_{e \in \mathcal{E}_{h,i}} \Phi^e_i + \sum_{f \in \mathcal{F}_{h,i}} \Phi^{e,\partial}_i = 0, \quad \forall i \in \mathcal{N}_h,
\]

- High-order schemes requires high-order boundary representation
- Here isoparametric formulation used. Same order for solution and geometry
- Piecewise polynomial approximation of the geometry
Navier-Stokes manufactured solutions
Definition of the exact solution

\[ \nabla \cdot f^a (u^{MS}) - \nabla \cdot f^v (u^{MS}, \nabla u^{MS}) = S(u^{MS}) \]
**Navier-Stokes Manufactured Solutions**

**Linear and Non-linear schemes: solution and gradient order of accuracy**

![Graphs showing linear and non-linear schemes](image-url)
Laminar flow around a delta wing

\[ M = 0.5, \alpha = 12.5^\circ, \text{Re} = 4000 \]

- Separated steady flow at high angle of attack
- Linear scheme with SPR-ZZ gradient reconstruction
- Three levels of nested grids. Parallel simulations: 8, 16, 32 processors
- Boundary conditions: no-slip adiabatic wall, symmetry plane and far-field
- Residual drop \( \sim 10^{-10} \), respect to the initial value
LAMINAR FLOW AROUND A DELTA WING
EXAMPLE OF SOLUTION AND RESIDUAL ON THE FINEST GRID

(a) Mach number and streamlines

(b) Residual $P^1$ and $P^2$
LAMINAR FLOW AROUND A DELTA WING
FORCE COEFFICIENTS CONVERGENCE

(a) Error CD
- Reference values: extrapolated from Higher-order DG resulted
- Effect on the CD of the singularity at the leading edge
- Benefit of high-order approximation on CL

(b) Error CL
Shock-Wave/Laminar Boundary Layer Interaction
Problem specifications

- $M_\infty = 2.15$, $\theta_s = 30.8^\circ$, $Re_{LS} = 10^5$
- Non-linear scheme with SPR-ZZ gradient reconstruction
- Grid: $N_x = 90$ (uniform), $N_y = 85$ (clustered to the wall)
- Second and third order simulations
- Residual drop at least $\sim 10^{-8}$ with respect to the initial value
**SHOCK-WAVE/LAMINAR BOUNDARY LAYER INTERACTION**

**Problem specifications**

- $M_\infty = 2.15$, $\theta_s = 30.8^\circ$, $\text{Re}_{LS} = 10^5$
- Non-linear scheme with SPR-ZZ gradient reconstruction
- Grid: $N_x = 90$ (uniform), $N_y = 85$ (clustered to the wall)
- Second and third order simulations
- Residual drop at least $\sim 10^{-8}$ with respect to the initial value
- Example of third order solution

![Pressure contours](image1.png)

![Recirculation bubble](image2.png)

(a) Pressure contours  
(b) Recirculation bubble
Shock-Wave/Laminar Boundary Layer Interaction
Comparison between second and third order accurate results

Profiles at \( y = 0.29 \)

(a) Pressure

(b) Mach number
Governing equations

RANS equations with Spalart-Allmaras model

\[ \nabla \cdot \mathbf{f}^a(\mathbf{u}) - \nabla \cdot \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \]

\[ \mathbf{u} = \begin{pmatrix} \rho \\ m \\ E^t \\ \mu_t^* \end{pmatrix}, \quad \mathbf{f}^a = \begin{pmatrix} \frac{m}{\rho} \\ \frac{m \otimes m}{m} + P^I \\ \frac{m}{\rho} \mu_t^* \\ \frac{m}{\rho} \mu_t^* \end{pmatrix}, \quad \mathbf{f}^v = \begin{pmatrix} 0 \\ \mathbf{S} \cdot \frac{m}{\rho} + \kappa \nabla T \\ \frac{\mu + \mu_t^*}{\sigma} \nabla \left( \frac{\mu_t^*}{\rho} \right) \end{pmatrix} \]

- \( \mathbf{S} = (0, 0, 0, S_{sa})^T \), \( S_{sa} = P_{sa} + E_{sa} - D_{sa} \) Spalart-Allmaras source term
- Fully coupled approach: augmented advective and diffusive flux function
- Discretization with linear and non-linear schemes, similar to Navier-Stokes
- Additional work
  - Modifications to the original SA equation
  - Implicit solver
**Spalart-Allmaras model**

**Improvement of the robustness**

- Negative values of the turbulent working variable in the outer part of the boundary layer and wakes (insufficient mesh resolution)
- Clipping the eddy viscosity produce a physical valid model

\[
\mu_t = \begin{cases} 
\mu_t^* f v_1, & \mu_t^* > 0 \\
0, & \mu_t^* \leq 0 
\end{cases}
\]

but robustness issues in the numerical solver are still present

- Several modification to the original SA model proposed
  - Not completely satisfying for HO methods
SPALART-ALLMARAS MODEL

IMPROVEMENT OF THE ROBUSTNESS (PERAIRE ET AL., 2011)

- A robust implementation of the SA equation requires to remove the adverse effects of negative values of $\mu^*_t$ on the turbulence model

$$\mu^*_t = \mu \frac{\mu^*_t}{\mu} = \mu \chi$$

- The source term and flux functions of the SA equation tend to zero for $\mu^*_t < 0$

- Differentiability of the equation
Implicit scheme for turbulent flows


- Jacobian-free approach non robust for HO RANS
- Solve with the symmetric variation of the Gauss-Seidel method with multiple sweeps \( (k = 1, \ldots, k_{\text{max}}) \)

\[
\left[ \frac{I}{\Delta t^n} + \frac{\partial R_i}{\partial u_i} \right] \Delta u_i^{(k+1)} = -R_i(u^n) - \sum_{j \in \Omega_i, j \neq i} \frac{\partial R_i}{\partial u_j} \Delta u_j^{(*)}
\]

\( \Delta u_j^{(*)} \) the most recently updated solution

- Linearization of the residual at \( u^{(*)} = u^n + \Delta u^{(*)} \)

\[
R_i(u^{(*)}) \approx R_i(u^n) + \sum_{j \in \Omega_i} \frac{\partial R_i}{\partial u_j} \Delta u_j^{(*)}
\]

\[
= R_i(u^n) + \sum_{j \in \Omega_i, j \neq i} \frac{\partial R_i}{\partial u_j} \Delta u_j^{(*)} + \frac{\partial R_i}{\partial u_i} \Delta u_i^{(*)}
\]
**Implicit Scheme for Turbulent Flows**

**Non-linear LU-SGS:** Y. Sun, Z.J. Wang Y. Liu. 2009

- Substituting in the RHS of the SGS scheme

\[
\left[ \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R_i}{\partial u_i} \right] \left( \Delta u^{(k+1)}_i - \Delta u^*_i \right) = -R_i(u^*) + \frac{\Delta u^*_i}{\Delta t^n},
\]

which is solved with the forward and backward sweeps.

- The RHS is nothing but the residual evaluated at the latest available solutions. (So the algorithm is called non-linear)

- At the beginning of each step the diagonal block of the Jacobian in the LHS is inverted using LU decomposition.

- Only diagonal blocks of the approximated Jacobian used.
Subsonic and transonic flow over a RAE2822 airfoil

- Re = $6.5 \times 10^6$, $\alpha = 2.79^\circ$
  - M = 0.4 (subsonic)
  - M = 0.734 (transonic)
- SPR-ZZ gradient reconstruction
  - Linear scheme (subsonic)
  - Non-linear scheme (transonic)
- Non-linear LU-SGS implicit scheme
  - Residual drop $\sim 10^{-10}$
Subsonic flow over a RAE2822 airfoil
Example of third order results on a fine grid

(a) Mach number contours

(b) Residual history
Subsonic flow over a RAE2822 airfoil
Force coefficient convergence

(a) Cl

(b) Cd
TRANSONIC FLOW OVER A RAE2822 AIRFOIL
EXAMPLE OF THIRD ORDER RESULTS ON A FINE GRID

(a) Mach number contours

(b) Residual history
Transonic flow over a RAE2822 airfoil
Comparison second and third order results ($N_{dof} = 32784$)

Experiments

<table>
<thead>
<tr>
<th></th>
<th>Cl</th>
<th>Cd</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD 3rd ord.</td>
<td>0.7978</td>
<td>0.0192</td>
</tr>
<tr>
<td>RD 2nd ord.</td>
<td>0.7712</td>
<td>0.0197</td>
</tr>
</tbody>
</table>

DG (UMich) 0.798 0.0191
Exp 0.803 0.0168
Transonic flow over a RAE2822 airfoil
Force coefficient convergence: DoFs

(a) $C_l$

(b) $C_d$

Dante De Santis (Stanford University)
Transonic flow over a RAE2822 airfoil

Force coefficient convergence: CPU time

(a) $C_L$

(b) $C_d$
L1T2 high-lift multi-element airfoil

$M = 0.197$, $Re = 3.52 \times 10^6$, $\alpha = 20.18^\circ$

- Unstructured grid of triangles: 33338 elements
- 2nd and 3rd order computations: linear scheme + SPR-ZZ
- Convergence criterion: normalized L2 residual $\sim 1 \times 10^{-10}$
L1T2 high-lift multi-element airfoil
P1 & P2 Mach number contours
**L1T2 high-lift multi-element airfoil**

**P1 & P2 Cp and experimental data**

---

![Slat](image1)

![Main](image2)

![Flap](image3)

---

Dante De Santis (Stanford University)
Subsonic flow over the NASA 65° sweep delta wing

\( M = 0.4, \alpha = 13.3°, \text{Re} = 3 \times 10^6 \)

- Linear scheme with SPR-ZZ
- Grid of 1,145,797 tetrahedra
- Residual drop \( \sim 10^{-5} \)
- Parallel simulations: 96 processors

(a) Surface grid

(b) \( C_p \) and \( \mu_t \) contours (third order)
**Subsonic flow over the NASA 65° sweep delta wing**

**Third order results: Cp on the wing at different spanwise sections**
## Conclusions & Perspectives

### Numerical Method
- Idea developed first for advection-diffusion problems
- Based on accurate reconstruction of the gradient
- Possibility to get **monotone & accurate** solutions

### Numerical Results
- Extensive evaluation of the numerical solver
- Extension to N-S and RANS equations
- Efficient and robust implicit scheme
- Capabilities of HO approach in solving challenging applications
- Extension to complex EOS and hypersonic flows
Conclusions & Perspectives

Numerical Method
- Idea developed first for advection-diffusion problems
- Based on accurate reconstruction of the gradient
- Possibility to get monotone & accurate solutions

Numerical Results
- Extensive evaluation of the numerical solver
- Extension to N-S and RANS equations
- Efficient and robust implicit scheme
- Capabilities of HO approach in solving challenging applications
- Extension to complex EOS and hypersonic flows

Big Challenge
- Unsteady problems: not clear how to get HO
Acknowledgment

- Prof. R. Abgrall (University of Zurich)
- Funded by the European FP7 STREP IDIHOM
Backup slides
Write the original advection-diffusion problem as a first order system

\[
\begin{align*}
\nabla \cdot f(u) - \nabla \cdot (\nu q) &= 0 \\
q - \nabla u &= 0
\end{align*}
\]

Discretize the f.o.s with a central scheme + a streamline stabilization

\[
\int_{\Omega_e} \psi_i \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) + \int_{\Omega_e} A \cdot \nabla \psi_i \tau \left( \nabla \cdot f(u_h) - \nabla \cdot (\nu q_h) \right) = 0
\]

where

\[
A \cdot \nabla \psi_i = \begin{pmatrix}
a \cdot \nabla \psi_i & -\nu \frac{\partial \psi_i}{\partial x} & -\nu \frac{\partial \psi_i}{\partial y} \\
-\frac{\partial \psi_i}{\partial x} & 0 & 0 \\
-\frac{\partial \psi_i}{\partial y} & 0 & 0
\end{pmatrix}
\]

and

\[
\tau = \begin{pmatrix}
\tau_a & 0 & 0 \\
0 & \tau_a & 0 \\
0 & 0 & \tau_a
\end{pmatrix}
\]
\[
\frac{\partial u}{\partial t} + a \cdot \nabla u = \nu \left( \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} \right)
\]

\[
\frac{\partial p}{\partial t} = \frac{1}{T_r} \left( \frac{\partial u}{\partial x} - p \right)
\]

\[
\frac{\partial q}{\partial t} = \frac{1}{T_r} \left( \frac{\partial u}{\partial y} - q \right)
\]

\[
\frac{\partial u}{\partial t} + A \cdot \nabla u = S,
\]

with

\[
u = \begin{pmatrix} u \\ p \\ q \end{pmatrix}, \quad A_x = \begin{pmatrix} a_x & -\nu & 0 \\ \frac{1}{T_r} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_y = \begin{pmatrix} a_y & 0 & -\nu \\ 0 & 0 & 0 \\ \frac{1}{T_r} & 0 & 0 \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ -\frac{p}{T_r} \\ -\frac{q}{T_r} \end{pmatrix}
\]
**Solve:** $a \cdot \nabla u = \nu \nabla \cdot \nabla u$, with $\nu = 0.01$ (Pe $\sim 1$)

**Linear scheme:** Benefit of high-order approximation

- Similar results with the non-linear scheme
- Similar results with grids of quadrangles and hybrid elements
- What is the effect of the grid regularity?
Solve: $\mathbf{a} \cdot \nabla u = \nu \nabla \cdot \nabla u$, with $\nu = 0.01$ (Pe $\sim 1$)

Linear scheme and non-linear scheme on perturbed grid
**Solve:** \( \mathbf{a} \cdot \nabla u = \nu \nabla \cdot \nabla u \), with \( \nu = 10^{-6} \) (Pe \( \gg 1 \))

**Linear scheme and non-linear scheme unstructured grid**

---

Dante De Santis (Stanford University)
Anisotropic diffusion problem
Linear scheme and non-linear scheme unstructured grid

\[- \nabla \cdot \mathbf{K} \nabla u^h = 0 \]

\[ \mathbf{K} = \begin{pmatrix} 1 & 0 \\ 0 & \delta \end{pmatrix}, \delta = 10^3 \]
RD discretization of system of equations

- Calculation of the total residual

\[ \Phi^e = \int_{\partial \Omega_e} \left( f^a(u_h) - K(u_h) \tilde{\nabla} u_h \right) \cdot \hat{n} \]

\[ \tilde{\nabla} u^h: \text{gradient reconstruction similarly to scalar case} \]

**Linear scheme**

\[ \Phi_i^e = \frac{\Phi^e}{N_{\text{dof}}} + \int_{\Omega_e} A \cdot \nabla \psi_i \tau \left( A \cdot \nabla u_h - \nabla \cdot (K \tilde{\nabla} u_h) \right) \]

\[ + \int_{\Omega_e} K \cdot \nabla \psi_i \left( (\nabla u_h - \tilde{\nabla} u_h) \right) \]

\[ \tau = \frac{|\Omega_e|}{N_{\text{dim}}} \left( \sum_{i \in e} R_{n_i} \Lambda_{n_i}^+ L_{n_i} + K_{jj} \right)^{-1} \]
RD discretization of system of equations

- Calculation of the total residual

\[ \Phi^e = \int_{\partial \Omega_e} \left( f^a(u_h) - K(u_h) \tilde{\nabla} u_h \right) \cdot \hat{n} \]

\( \tilde{\nabla} u^h \): gradient reconstruction similarly to scalar case

Non-linear scheme

\[ \hat{\Phi}^e_{i,Rv} = \Phi^e_i + \varepsilon^e_h(u_h) \int_{\Omega_e} \left( A \cdot \nabla \psi_i - K \nabla \psi_i \right) \Xi \left( A \cdot \nabla u_h - \nabla \cdot \left( K \tilde{\nabla} u_h \right) \right) d\Omega \]

\[ + \int_{\Omega_e} K \tilde{\nabla} \psi_i \cdot \left( \nabla u_h - \tilde{\nabla} u_h \right) d\Omega \]

\[ \Xi = \frac{1}{2} |\Omega_e| \left( \sum_{i \in N^e_h} R_{n_i}(\bar{u}) \Lambda^+_n(\bar{u}) L_{n_i}(\bar{u}) + \sum_{j=1}^{N_{\text{dim}}} K_{jj}(\bar{u}) \right)^{-1} \]
Boundary conditions
How to impose boundary conditions

- Imposition of weak boundary conditions

\[ \sum_{e \in E_{h,i}} \Phi^e_{i} + \sum_{f \in F_{h,i}} \Phi^e_{i,\partial} = 0, \quad \forall i \in N_h, \]

- Boundary residual contribution

\[ \Phi^e_{i,\partial} = \int_{\partial \Omega_i \cap \partial \Omega} \psi_i (f(u^\partial) - f(u_h)) \cdot n \ d\partial \Omega \]

- Correction flux: \((f(u^\partial) - f(u_h)) \cdot n\)

  - Slip wall: \( (f^a(u_{wall}^\partial) - f^a(u_h)) \cdot \hat{n} = -v_n (\rho, \rho v, E^t + P)^T \)

  - In/Out flow: \( (f^a(u_{in/out}^\partial) - f^a(u_h)) \cdot \hat{n} = A_n^- (u_h^\partial)(u_{in/out}^\partial - u_h) \)

  - Adiabatic wall: \( v = 0 \) (strong) and \( f^v(u_{wall}^\partial) = (0, 0, 0, -\kappa \nabla T \cdot n)^T \)
Implicit Euler scheme with linearization: 

$$A(u_h^n) \Delta u_h^n = -R(u_h^n)$$

$$\left[ \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right] \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n$$

Implicit time integration
Inexact Newton-Krylov methods
Implicit Euler scheme with linearization: \( A(u^n_h) \Delta u^n_h = -R(u^n_h) \)

\[
\left( \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \right) \Delta u^n = -R(u^n_h), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

Approximated solution \( \|R(u^n_h) + A(u^n_h)\Delta u^n_h\| \leq \eta^h \|R(u^n_h)\| \) with GMRES
Implicit time integration
Inexact Newton-Krylov methods

- Implicit Euler scheme with linearization: \( A(u^n_h)\Delta u^n_h = -R(u^n_h) \)

\[
\begin{bmatrix}
\frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h)
\end{bmatrix}
\Delta u^n = -R(u^n_h), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

- Approximated solution \( \|R(u^n_h) + A(u^n_h)\Delta u^n_h\| \leq \eta^n_h \|R(u^n_h)\| \) with GMRES

- Impossible to compute the analytical Jacobian: poor iterative convergence
Implicit time integration
Inexact Newton-Krylov methods

- Implicit Euler scheme with linearization: \( A(u^n_h) \Delta u^n_h = -R(u^n_h) \)

\[
\left[ \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \right] \Delta u^n = -R(u^n_h), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

- Approximated solution \( \|R(u^n_h) + A(u^n_h)\Delta u^n_h\| \leq \eta^n_h \|R(u^n_h)\| \) with GMRES

**Jacobian-free**

\[
Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \right) w
\]
Implicit time integration
Inexact Newton-Krylov methods

- Implicit Euler scheme with linearization: 
  \[ A(u_h^n) \Delta u_h^n = -R(u_h^n) \]

  \[
  \left[ \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right] \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n
  \]

- Approximated solution \[ \|R(u_h^n) + A(u_h^n)\Delta u_h^n\| \leq \eta_h^n \|R(u_h^n)\| \] with GMRES

Jacobian-free

\[
Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right) w
\]

\[
\approx \frac{I}{\Delta t^n} w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}
\]
Implicit Euler scheme with linearization:

\[ A(u_h^n) \Delta u_h^n = -R(u_h^n) \]
\[
\begin{bmatrix}
\frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n)
\end{bmatrix} \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

Approximated solution \( \| R(u_h^n) + A(u_h^n) \Delta u_h^n \| \leq \eta_h^n \| R(u_h^n) \| \) with GMRES

Jacobian-free

\[
Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right) w
\]
\[
\simeq \frac{I}{\Delta t^n} w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}
\]
\[
\epsilon = \frac{1 + \| u \|_{L_2}}{\| w \|_{L_2}} \epsilon_{\text{rel}}, \quad \epsilon_{\text{rel}} = 10^{-8}
\]
Implicit time integration
Inexact Newton-Krylov methods

- Implicit Euler scheme with linearization:
  \[ A(u^*_h)\Delta u^n_h = -R(u^n_h) \]

\[
\frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \Delta u^n = -R(u^n_h), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

- Approximated solution

\[ \|R(u^n_h) + A(u^n_h)\Delta u^n_h\| \leq \eta^n_h \|R(u^n_h)\| \text{ with GMRES} \]

Jacobian-free

\[ Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \right) w \]

\[ \simeq \frac{I}{\Delta t^n} w + \frac{R(u^n_h + \epsilon w) - R(u^n_h)}{\epsilon} \]

\[ \epsilon = \sqrt{1 + \|u\|_{L_2}} \|w\|_{L_2} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8} \]

Preconditioning: \[ AP^{-1}Px = b \]

\[ AP^{-1}w = b \quad \text{and} \quad x = P^{-1}w \]
Implicit Euler scheme with linearization: \( A(u_h^n) \Delta u_h^n = -R(u_h^n) \)

\[
\left[ \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right] \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

Approximated solution \( \| R(u_h^n) + A(u_h^n) \Delta u_h^n \| \leq \eta_h^n \| R(u_h^n) \| \) with GMRES

**Jacobian-free**

\[
Aw = \left( \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right) w
\]

\[
\approx \frac{\mathbb{I}}{\Delta t^n} w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}
\]

\[
\epsilon = \frac{\sqrt{1+\|u\|_{L_2}^2}}{\|w\|_{L_2}} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8}
\]

**Preconditioning:** \( AP^{-1} Px = b \)

\[
AP^{-1} w = b \quad \text{and} \quad x = P^{-1} w
\]

LU-SGS Preconditioner:

\[
(D + L)D^{-1}(D + U) x = b + (LD^{-1}U)x
\]
**Implicit time integration**

**Inexact Newton-Krylov methods**

- Implicit Euler scheme with linearization: \( A(u^n_h) \Delta u^n_h = -R(u^n_h) \)

\[
\begin{bmatrix}
I \\
\Delta t^n
\end{bmatrix}
+ \frac{\partial R}{\partial u}(u^n_h) \Delta u^n = -R(u^n_h), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

- Approximated solution \( \|R(u^n_h) + A(u^n_h)\Delta u^n_h\| \leq \eta^n_h \|R(u^n_h)\| \) with GMRES

---

**Jacobian-free**

\[
Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u^n_h) \right) w
\]

\[
\approx \frac{I}{\Delta t^n} w + \frac{R(u^n_h + \epsilon w) - R(u^n_h)}{\epsilon}
\]

\[
\epsilon = \frac{\sqrt{1+\|u\|_2}}{\|w\|_2} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8}
\]

---

**Preconditioning:** \( AP^{-1}Px = b \)

\[
AP^{-1}w = b \quad \text{and} \quad x = P^{-1}w
\]

LU-SGS Preconditioner:

\[
(D + L)D^{-1}(D + U)x = b + \begin{bmatrix} LD_{L^{-1}}U \end{bmatrix} x
\]
Implicit time integration
Inexact Newton-Krylov methods

• Implicit Euler scheme with linearization: \( A(u_h^n) \Delta u_h^n = -R(u_h^n) \)

\[
\begin{bmatrix}
\frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n)
\end{bmatrix} \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n
\]

• Approximated solution \( \|R(u_h^n) + A(u_h^n)\Delta u_h^n\| \leq \eta_h^n \|R(u_h^n)\| \) with GMRES

Jacobian-free

\[
Aw = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right)w
\]

\[
\approx \frac{I}{\Delta t^n}w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}
\]

\[
\epsilon = \sqrt{1 + \|u\|_{L_2}} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8}
\]

Preconditioning: \( AP^{-1}Px = b \)

\[
AP^{-1}w = b \quad \text{and} \quad x = P^{-1}w
\]

LU-SGS Preconditioner:

\[
(D + L)D^{-1}(D + U)x = b + \frac{(LD^{-1}U)x}{P}
\]
Implicit time integration
Inexact Newton-Krylov methods

- Implicit Euler scheme with linearization:
  \[ A(u_h^n) \Delta u_h^n = -R(u_h^n) \]

  \[
  \left[ \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right] \Delta u^n = -R(u_h^n), \quad \Delta u^n \equiv u^{n+1} - u^n
  \]

- Approximated solution
  \[ \| R(u_h^n) + A(u_h^n) \Delta u_h^n \| \leq \eta_h^n \| R(u_h^n) \| \text{ with GMRES} \]

Jacobian-Free

\[ A w = \left( \frac{I}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right) w \]

\[
\approx \frac{I}{\Delta t^n} w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}
\]

\[
\epsilon = \sqrt{1 + \| u \|_{L_2}} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8}
\]

Preconditioning:

\[ AP^{-1}Px = b \]

\[ AP^{-1}w = b \quad \text{and} \quad x = P^{-1}w \]

LU-SGS Preconditioner:

\[ P = (D + L)D^{-1}(D + U) \]
Implicit Euler scheme with linearization: $A(u_h^n) \Delta u_h^n = -R(u_h^n)$

$$\left[ \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right] \Delta u^n = -R(u_h^n), \quad \Delta u^n = u^{n+1} - u^n$$

Approximated solution $\| R(u_h^n) + A(u_h^n) \Delta u_h^n \| \leq \eta_h^n \| R(u_h^n) \| \text{ with GMRES}$

**Jacobian-free**

$$Aw = \left( \frac{\mathbb{I}}{\Delta t^n} + \frac{\partial R}{\partial u}(u_h^n) \right) w$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n} w + \frac{R(u_h^n + \epsilon w) - R(u_h^n)}{\epsilon}$$

$$\epsilon = \sqrt{1 + \| u \|_{L_2}} \epsilon_{rel}, \quad \epsilon_{rel} = 10^{-8}$$

**Preconditioning:** $AP^{-1} Px = b$

$$AP^{-1}w = b \quad \text{and} \quad x = P^{-1}w$$

LU-SGS Preconditioner:

$$P = (D + L)D^{-1}(D + U)$$

$$x^*_i = D_i^{-1} \left( w_i - \sum_{j<i} w_j x_j^* \right), \quad i = 1, \ldots, N_{dof}$$

$$x_i = x_i^* - D_i^{-1} \sum_{j>i} w_j x_j, \quad i = N_{dof}, \ldots, 1$$
Laminar NACA-0012
M = 0.5, α = 0, Re = 5000

- 4216 P2 elements (8564 DOFs)
- Linear scheme with ZZ-SPR
- Residual down to zero machine
Laminar NACA-0012

$M = 0.5, \alpha = 0, Re = 5000$

3rd order

2nd order (Equivalent-DOF)
Laminar NACA-0012

M = 0.5, α = 0, Re = 5000
Profiles at $y = 0.29$

(a) Pressure

(b) Mach number
Profiles at $y = 0.15$

(a) Pressure

(b) Mach number
Shock-Wave/Laminar Boundary Layer Interaction
Comparison between second and third order accurate results

Profiles along the wall

(a) Pressure

(b) Friction coefficient
Turbulent flow over a flat plate

$M = 0.2, \text{Re}_{L=1} = 5 \times 10^6$

- Linear scheme with SPR-ZZ gradient reconstruction
- Jacobian-free with LU-SGS preconditioner (Residual drop $\sim 10^{-10}$)
- Nested grids
- Value of $y_1^+$ (at $x = 0.97$)

<table>
<thead>
<tr>
<th>Grid</th>
<th>$y_{1P_1}$</th>
<th>$y_{1P_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$35 \times 25$</td>
<td>1.500</td>
<td>0.765</td>
</tr>
<tr>
<td>$69 \times 49$</td>
<td>0.722</td>
<td>0.372</td>
</tr>
<tr>
<td>$137 \times 97$</td>
<td>0.359</td>
<td>0.184</td>
</tr>
<tr>
<td>$273 \times 193$</td>
<td>0.182</td>
<td>–</td>
</tr>
</tbody>
</table>
Turbulent flow over a flat plate
$M = 0.2, Re_{L=1} = 5 \times 10^6$

- Linear scheme with SPR-ZZ gradient reconstruction
- Jacobian-free with LU-SGS preconditioner (Residual drop $\sim 10^{-10}$)
- Nested grids
- Velocity profiles (at $x = 0.97$)

(a) Second order
(b) Third order
Turbulent flow over a flat plate
Friction coefficient along the plate

(a) Second order
(b) Third order
Turbulent flow over a flat plate
Drag coefficient values

(a) $C_d$

(b) $C_d$ error