Implicit Finite Volume Schemes and Preconditioned Krylov Subspace Methods for the Discretization of Hyperbolic and Parabolic Conservation Laws

Andreas Meister

UMBC, Department of Mathematics and Statistics

Content

- Aims
- Navier Stokes Equations
- Finite Volumen Scheme
- Iterative Solution Methods and Preconditioning
- Numerical Results

Aims

- **Development of a numerical method for the simulation of real life fluid dynamics problems in the project CASWING**
- **Flexible usage as a basic method**
  - Turbulent combustion
  - Chemical reacting fluid flows
  - Low Mach number fluid flows
  - Thermoregulation
  - Fluid Structur Interaction
- **Investigation of methods to accelerate the convergence**

Aims

- **Requirements**
  - Flow dependent discretization of arbitrary complex geometries
  - Taking account of moving boundaries
  - Consideration of turbulence effects
  - Minimization of computing time
    - ⋆ Implicit scheme
    - ⋆ Preconditioned Krylov subspace methods

Balance laws

Unsteady, compressible and dimesionless Navier Stokes equations

$$\partial_t \mathbf{u} + \sum_{m=1}^{2} \partial_{x_m} \mathbf{g}_m^c(\mathbf{u}) = \frac{1}{Re_\infty} \sum_{m=1}^{2} \partial_{x_m} \mathbf{f}_m^\nu(\mathbf{u})$$

Vector of conserved variables

$$\mathbf{u} = (\rho, \rho v_1, \rho v_2, \rho E)$$

Convective flux function

$$\mathbf{g}_m^c(\mathbf{u}) = \begin{pmatrix} \rho v_m \\ \rho v_m v_1 + \delta_1^m p \\ \rho v_m v_2 + \delta_2^m p \\ \rho H v_m \end{pmatrix}$$

Viscous flux function

$$\mathbf{f}_m^\nu(\mathbf{u}) = \begin{pmatrix} 0 \\ \mu S_{1m} \\ \mu S_{2m} \\ \sum_{i=1}^{3} \mu S_{im} v_i + \frac{\mu\kappa}{\mathrm{Pr}_\infty \partial_{x_m} e} \end{pmatrix}$$

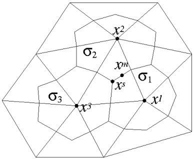
m = 1, 2

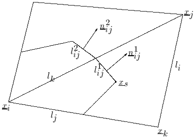
Finite Volume Method

$$\int_{\sigma(t)} \partial_t \mathbf{u}\, d\mathbf{x} + \sum_{m=1}^{2} \int_{\sigma(t)} \partial_{x_m} \mathbf{g}_m^c(\mathbf{u})\, d\mathbf{x} = \frac{1}{Re_\infty} \sum_{m=1}^{2} \int_{\sigma(t)} \partial_{x_m} \mathbf{f}_m^\nu(\mathbf{u})\, d\mathbf{x}$$

- Cell averages on a dual mesh

$$\mathbf{u}_i = \frac{1}{|\sigma_i|} \int_{\sigma_i} \mathbf{u}\, d\mathbf{x}$$





- Gauß Integral Theorem and Reynolds Transport Theorem (Evolution equation for cell averages)

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i(t)|} \left(\frac{1}{Re_\infty} \int_{\partial\sigma_i(t)} \sum_{m=1}^{2} \mathbf{f}_m^\nu(\mathbf{u}) n_m\, ds - \int_{\partial\sigma_i(t)} \sum_{m=1}^{2} \mathbf{f}_m^c(\mathbf{u}) n_m\, ds \right)$$

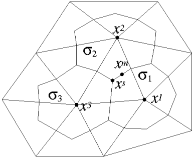
$$\mathbf{f}_m^c(u) = \mathbf{g}_m^c(u) - \nu_{netz,m}\mathbf{u}$$

Discretization of the convective fluxes

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

- Representation of the boundary integrals

$$\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds = \sum_{j\in N(i)}\sum_{k=1}^{2}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds$$



- Quadrature rule

$$\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds \approx \mathbf{H}^c(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j;\mathbf{n}_{ij}^k)|l_{ij}^k|$$

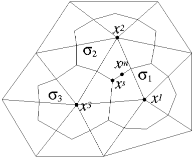
- Convective flux function $\mathbf{H}^c$ (AUSMDV, Lax-Friedrichs, ...)
- Spiatial discretization of higher order
  - TVD-scheme, WENO- und ENO-method

Discretization of the convective fluxes

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

- Representation of the boundary integrals

$$\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds = \sum_{j\in N(i)}\sum_{k=1}^{2}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds$$



- Quadrature rule

$$\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds \approx \mathbf{H}^c(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j;\mathbf{n}_{ij}^k)|l_{ij}^k|$$

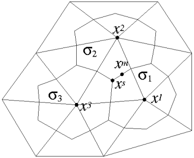
- Convective flux function $\mathbf{H}^c$ (AUSMDV, Lax-Friedrichs, ...)
- Spiatial discretization of higher order
  - TVD-scheme, WENO- und ENO-method

Discretization of the convective fluxes

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

- Representation of the boundary integrals

$$\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds = \sum_{j\in N(i)}\sum_{k=1}^{2}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds$$



- Quadrature rule

$$\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds \approx \mathbf{H}^c(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j;\mathbf{n}_{ij}^k)|l_{ij}^k|$$

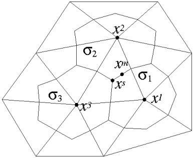
- Convective flux function $\mathbf{H}^c$ (AUSMDV, Lax-Friedrichs, ...)
- Spiatial discretization of higher order
  - TVD-scheme, WENO- und ENO-method

Discretization of the convective fluxes

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

- Representation of the boundary integrals

$$\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds = \sum_{j\in N(i)}\sum_{k=1}^{2}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds$$



- Quadrature rule

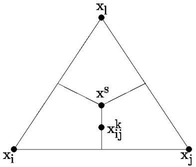
$$\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_{ij,m}^k\, ds \approx \mathbf{H}^c(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j;\mathbf{n}_{ij}^k)|l_{ij}^k|$$

- Convective flux function $\mathbf{H}^c$ (AUSMDV, Lax-Friedrichs, ...)
- Spiatial discretization of higher order
  - TVD-scheme, WENO- und ENO-method

Discretization of the viscous fluxes

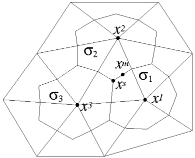
$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

$$\mathbf{f}_m^{\nu}(\mathbf{u}) = \begin{pmatrix} 0 \\ \tau_{1m} \\ \tau_{2m} \\ v_1\tau_{1m} + v_2\tau_{2m} + \frac{\kappa\gamma}{(\gamma-1)Pr}\partial_{x_m} T \end{pmatrix}$$



Values at the midpoint x_{ij}^k of the line segment l_{ij}^k:

- Velocity and viscosity
- Gradient of the velocity and the temperature



$$\frac{1}{Re}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m\, ds \approx \sum_{m=1}^{2}\mathbf{H}_m^{\nu}\left(\mathbf{v}, \mu, \nabla_{\mathbf{x}}\mathbf{v}, \nabla_{\mathbf{x}} T\right)\; n_m\, |l_{ij}^k|$$

Time stepping scheme

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\left(\frac{1}{Re}\int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{\partial\sigma_i}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}\left(\frac{1}{Re}\int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{\nu}(\mathbf{u})n_m ds - \int_{l_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^{c}(\mathbf{u})n_m ds\right)$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}\left(\sum_{m=1}^{2}\mathbf{H}_m^{\nu}(\mathbf{u})\;n_m\,|I_{ij}^k| - \int_{I_{ij}^k}\sum_{m=1}^{2}\mathbf{f}_m^c(\mathbf{u})n_m ds\right)$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\,\mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\,\mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u}_i = \frac{1}{|\sigma_i|}\sum_{j\in N(i)}\sum_{k=1}^{2}\left(\sum_{m=1}^{2}\mathbf{H}_m^{\nu}(\mathbf{u})\ n_m\,|l_{ij}^k| - \mathbf{H}^c(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j;\mathbf{n}_{ij}^k)|l_{ij}^k|\right)$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\ \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\ \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u}_i = \mathbf{F}_i(\mathbf{u}), \quad i = 1, \ldots, N$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}\left(\mathbf{u}\right), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^n\right)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^{n+1}\right)$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

  $\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^{n+1}\right)$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^{n+1})$

- Large time step size Δt
- Solution of a (non-)linear system

  $\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}\left(\mathbf{u}\right), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^n\right)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^{n+1}\right)$

- Large time step size Δt
- Solution of a (non-)linear system

$\rightarrow$ Large, sparse, badly conditioned

Time stepping scheme

$$\frac{d}{dt}\mathbf{u} = \mathbf{F}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^{4*N}$$

Numerical methods for ordinary differential equations

- Runge-Kutta-methods (Euler-scheme, DIRK, SDIRK)
- Adams-, Nyström-, Milne-Simpson-scheme
- BDF-method

Explicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}(\mathbf{u}^n)$

- Easy to implement
- Restrictive CFL-condition

Implicit ansatz $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{F}\left(\mathbf{u}^{n+1}\right)$

- Large time step size Δt
- Solution of a (non-)linear system

  $\longrightarrow$ Large, sparse, badly conditioned

Iterative solution methods

Consider $Ay = b$ with $A \in \mathbb{R}^{n \times n}$ nonsingular , $y, b \in \mathbb{R}^n$

- **Direct methods**
  - Gauß-elmination (LU-factorization)
    Cholesky-factorization (LL^T-factorization)
    QR-factorization
  - Why do we calculate the exact solution?
  - Rounding errors, Storage requirements

Iterative solution methods

Consider $Ay = b$ with $A \in \mathbb{R}^{n \times n}$ nonsingular , $y, b \in \mathbb{R}^n$

- **Iterative methods**
  - Splitting-methods
    - Jacobi-, Gauß-Seidel-, SOR-schemes
    - Easy to implement, slow convergence
  - Krylov subspace schemes
    - CG, GMRES, CGS, Bi-CGSTAB, TFQMR, QMRCGSTAB
    - Higher stability and often faster convergence
  - Condition number
    - Equivalent transformation of the linear system (Preconditioning)

Preconditioning

Equivalent transformation of the linear system

$$Ay = b$$

in

$$\begin{aligned} P_l A P_r z &= P_l b \\ y &= P_r z \end{aligned}$$

to accelerate the convergence.

- Left preconditioning: $P_l \neq I$
- Right preconditioning: $P_r \neq I$
- Both sided preconditioning: $P_r \neq I \neq P_l$

Control of the residual:

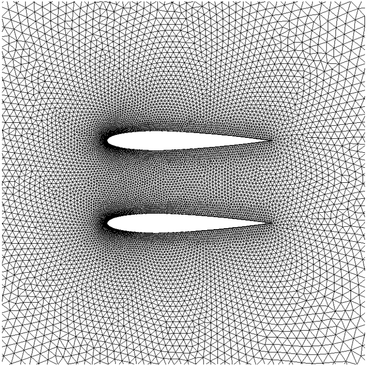
Residual right preconditioning: $r_m^r = b - AP_r z_m = b - Ay_m = r_m$

Residual left preconditioning: $r_m^l = P_l b - P_l A z_m = P_l r_m$

Preconditioning

- Incomplete LU-factorization
  - Additional storage, time consuming calculation
  - Frozen formulation, here ILU(p)
- Splitting schemes (GS, SGS)
  - No additional storage
  - No calculation
- Scaling (Jacobi et. al.)
  - Low additional storage, weak acceleration
- Characteristic approach
  - Eigenvalues $\lambda_{1/2} = v \cdot n$, $\lambda_{3/4} = v \cdot n + c$
  - Renumbering
  - Application of a splitting-based preconditioner CHA_{SGS} and CHA_{GS}

Ma = 0.55, Angle of attack 6°, inviscid
Triangulation: 26632 triangles, 13577 points



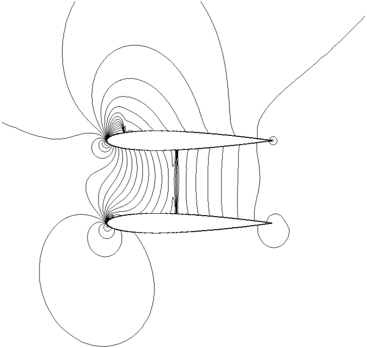
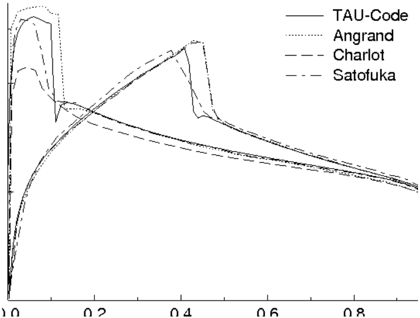


Abbildung: Partial view of the triangulation w.r.t. the Bi-NACA0012 Airfoil and density distribution

Ma = 0.55, Angle of attack 6°, inviscid
Triangulation: 26632 triangles, 13577 points



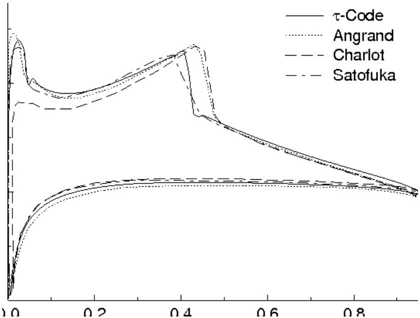


Abbildung: Distribution of the Mach number at the surface of the upper (left) and lower (right) airfoil

Combustion chamber at supersonic speed

Ma = 2.0, Angle of attack 0°, inviscid

Triangulation: 61731 triangles, 31202 points

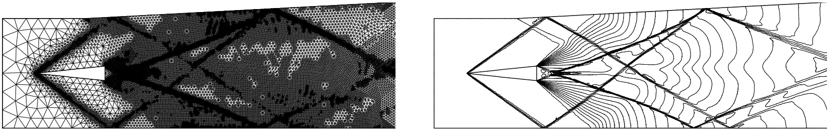


Abbildung: Adaptive triangulation and isolines of the density

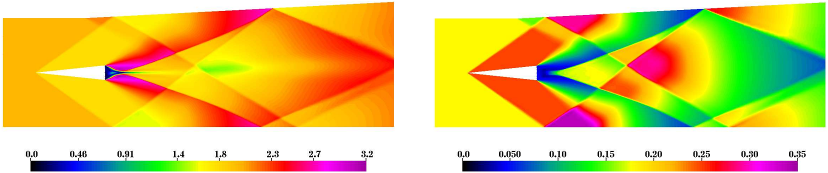
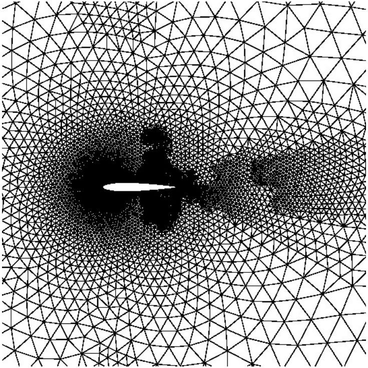


Abbildung: Distribution of the Mach number and the pressure

NACA0012-Airfoil

$$\text{Ma} = 0.85, \quad \alpha = 0^\circ, \quad \text{viscous}$$
$$\text{Re} = 500, \quad T_\infty = 273K, \quad \text{adiabatic}$$



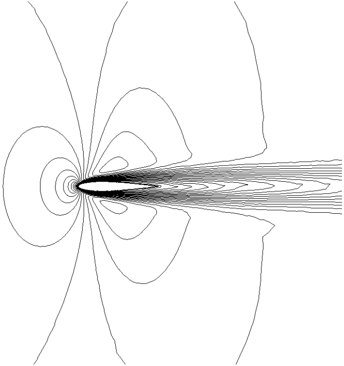


Abbildung: Partial view of the triangulation w.r.t. the NACA0012 Airfoil and isolines of the Mach number

C_p-distribution

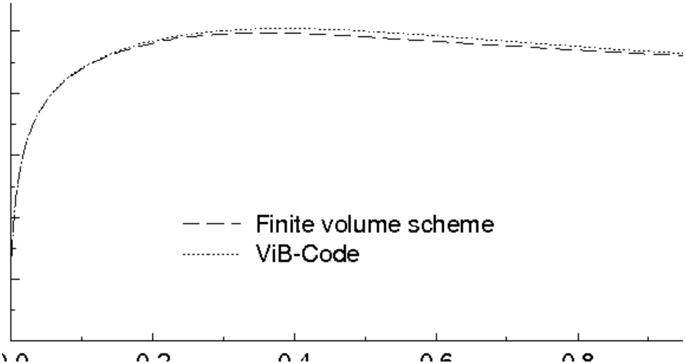
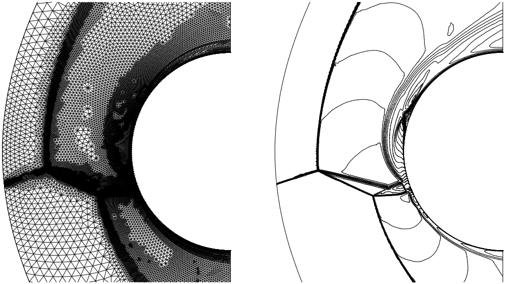
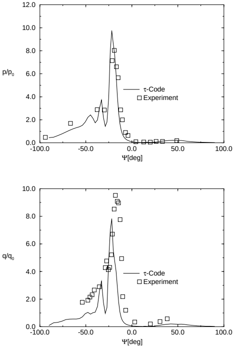


Abbildung: C_p-distribution unstructure implicit DLR- TAU-Code and structured explicit ViB-Code

Shock-Shock-Interaction

Ma = 8.03, Re = 193.750, $T_\infty = 122.1K$

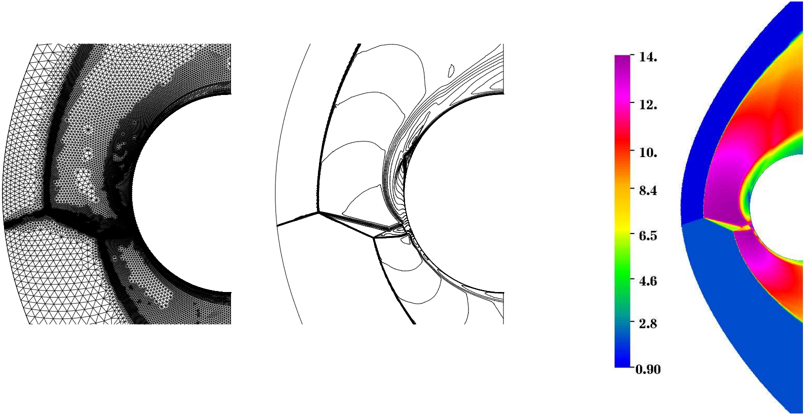




Adaptive triangulation and isolines of the Mach number distribution of the heat flux and the pressure

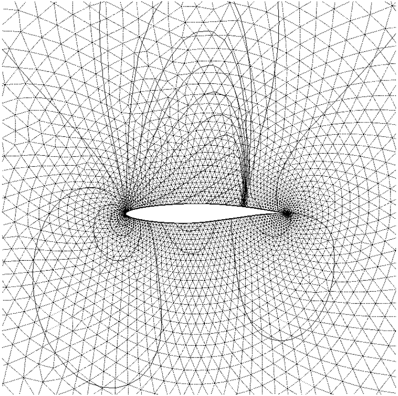
Stoß-Stoß-Wechselwirkung

Ma = 8.03, Re = 193.750, $T_\infty = 122.1K$



Adaptive triangulation, isolines of the Mach number and temperature distribution

Ma = 0.75,   Angle of attack 3°,   inviscid
Triangulation: 9974 triangles, 5071 points



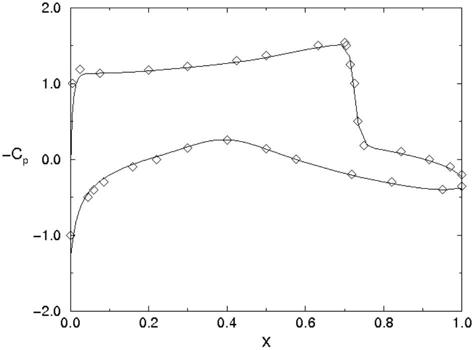


Abbildung: Distribution of the density and C_p-distribution

RAE 2822 Airfoil

Ma = 0.75, Angle of attack 3°, inviscid
Triangulation: 9974 triangles, 5071 points

Explicit scheme	Implicit scheme	
	Scaling	Incomplete LU(5)
100 %	24,36 %	2,86 %

Tabelle: Percentage comparison of the computing time

Ma = 0.65, Angle of attack 3°, inviscid
Triangulation: 46914 triangles, 23751 points

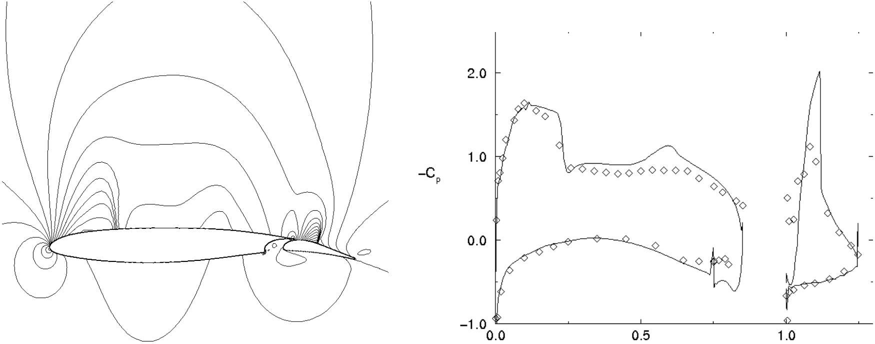


Abbildung: Distribution of the density and C_p-distribution

SKF1.1 Airfoil

Ma = 0.65, Angle of attack 3°, inviscid
Triangulation: 46914 triangles, 23751 points

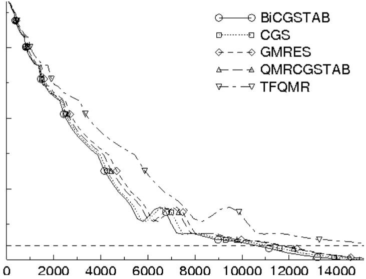
Explicit scheme	Implicit scheme	
	Scaling	Incomplete LU(5)
100 %	68,58 %	9,97 %

Tabelle: Percentage comparison of the computing time

- Convergence criterion

$$\text{Residuum} = \frac{1}{\triangle t}\sqrt{\sum_{i=1}^{\#\mathcal{N}_h} \frac{(\triangle\rho)_i^2\,|\sigma_i|}{|\Omega|}}$$

- Convergence history using ILU-preconditioned schemes



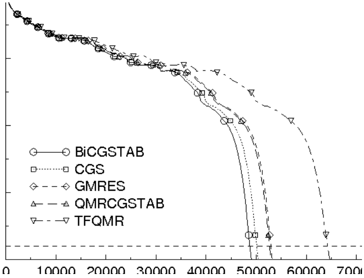


Abbildung: Bi-NACA0012 Airfoil (left) and Combustion chamber (right)

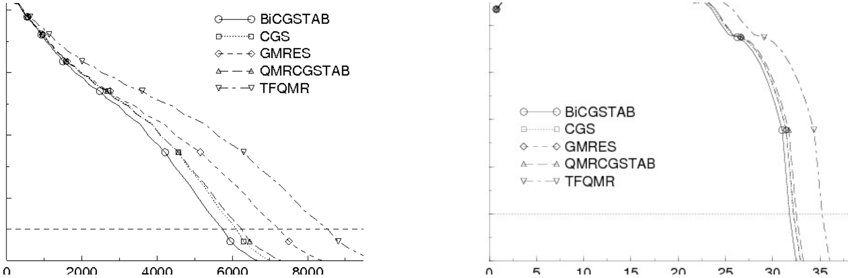
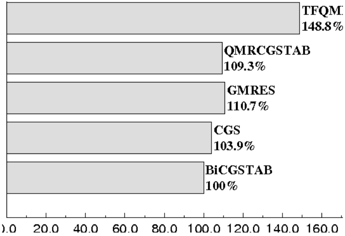
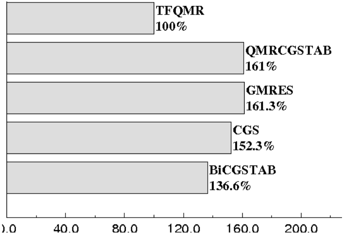


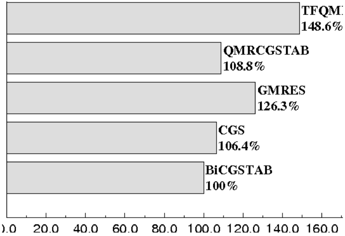
Abbildung: NACA0012 Airfoil (left) and decay of homogeneous turbulence (right)

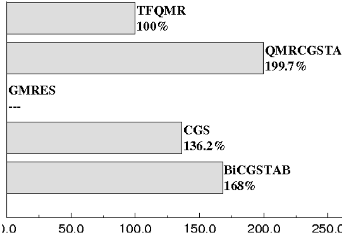
- Bi-NACA0012 Airfoil



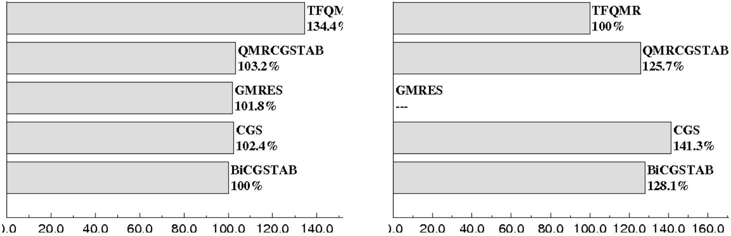


- NACA0012 Airfoil





- Laminar boundary layer of a flate plate: $Re = 6 \cdot 10^6$, $Ma = 5.0$



Preconditioned Bi-CGSTAB

Choose $z_0 \in \mathbb{R}^n$ and $\varepsilon > 0$
$r_0 = p_0 := P_l b - P_l A P_r z_0$
$\rho_0 := (r_0, r_0)_2$, $j := 0$
While $\|r_j\|_2 > \varepsilon$:

$$
\begin{aligned}
v_j &:= P_l A P_r p_j \\
\alpha_j &:= \frac{\rho_j}{(v_j,\, r_0)_2} \\
s_j &:= r_j - \alpha_j v_j \\
t_j &:= P_l A P_r s_j \\
\omega_j &:= \frac{(t_j, s_j)_2}{(t_j, t_j)_2} \\
z_{j+1} &:= z_j + \alpha_j p_j + \omega_j s_j \\
r_{j+1} &:= s_j - \omega_j t_j \\
\rho_{j+1} &:= (r_{j+1}, r_0)_2 \\
\beta_j &:= \frac{\alpha_j}{\omega_j} \frac{\rho_{j+1}}{\rho_j} \\
p_{j+1} &:= r_{j+1} + \beta_j (p_j - \omega_j v_j) \\
j &:= j + 1
\end{aligned}
$$

$y_j = P_r z_j$

Modified Bi-CGSTAB method

Choose $y_0 \in \mathbb{R}^n$ and $\varepsilon > 0$

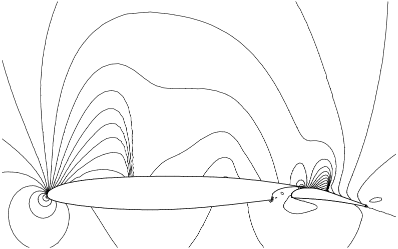
$r_0 = p_0 := b - A\,y_0\,,\ r_0^P = p_0^P := P_L\,r_0$

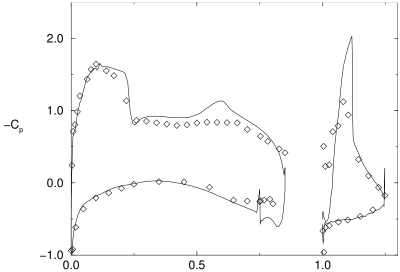
$\rho_0^P := \left(r_0^P, r_0^P\right)_2,\ j := 0$

While $\|r_j\|_2 > \varepsilon$:

$$
\begin{aligned}
\hat{v}_j &:= A\,P_R\,p_j^P\,, \quad v_j^P := P_L\,\hat{v}_j \\
\alpha_j^P &:= \frac{\rho_j^P}{(v_j^P, r_0^P)_2} \\
\hat{s}_j &:= r_j - \alpha_j^P\,\hat{v}_j\,, \quad s_j^P := P_L \hat{s}_j \\
\hat{t}_j &:= A\,P_R\,s_j^P\,, \quad t_j^P := P_L\,\hat{t}_j \\
\omega_j^P &:= \frac{(t_j^P, s_j^P)_2}{(t_j^P, t_j^P)_2} \\
y_{j+1} &:= y_j + \alpha_j^P\,P_R\,p_j^P + \omega_j^P\,P_R\,s_j^P \\
r_{j+1} &:= \hat{s}_j - \omega_j^P \hat{t}_j, \\
r_{j+1}^P &:= s_j^P - \omega_j^P\,t_j^P \\
\rho_{j+1}^P &:= (r_{j+1}^P, r_0^P)_2, \\
\beta_j^P &:= \frac{\alpha_j^P}{\omega_j^P}\,\frac{\rho_{j+1}^P}{\rho_j^P} \\
p_{j+1}^P &:= r_{j+1}^P + \beta_j^P(p_j^P - \omega_j^P\,v_j^P) \\
j &:= j + 1
\end{aligned}
$$

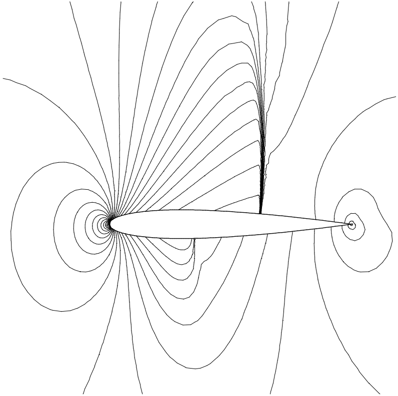
$$Ma = 0.65, \quad \alpha = 3.0^\circ$$

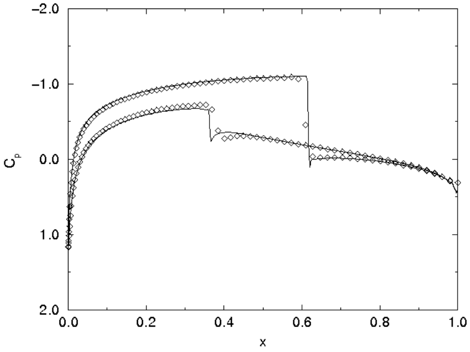




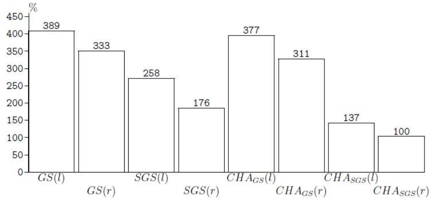
NACA0012 Airfoil

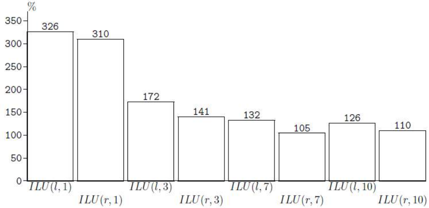
$$Ma = 0.85, \quad \alpha = 1.25^\circ$$





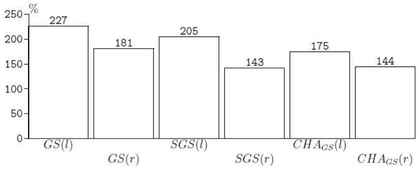
SKF 1.1 two element airfoil; $Ma = 0.65, \quad \alpha = 3.0^\circ$

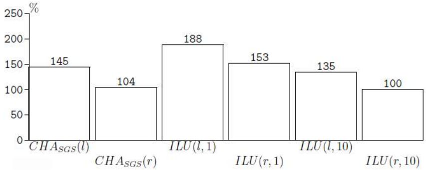




$D_1(r)$	$D_2(r)$	$D_\infty(r)$	$D_{Jac}(r)$	$D_{Jac}(l)$
1038%	953%	992%	438%	514%

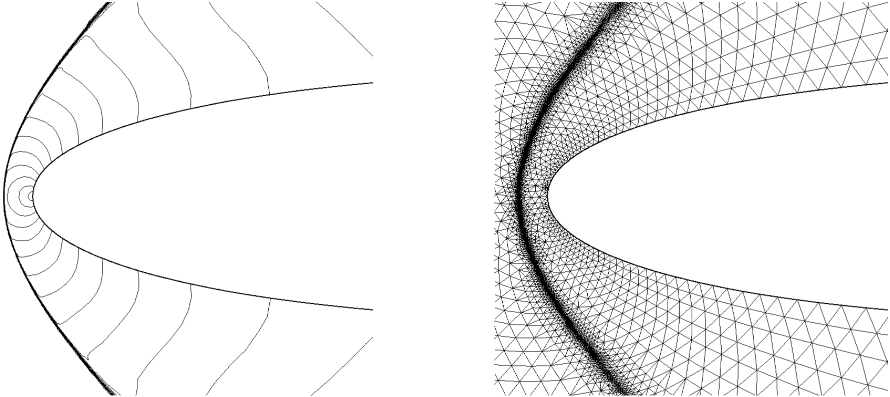
NACA0012 Airfoil $Ma = 0.85, \quad \alpha = 1.25^\circ$





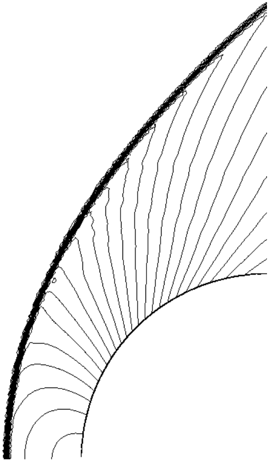
NACA0012 Airfoil

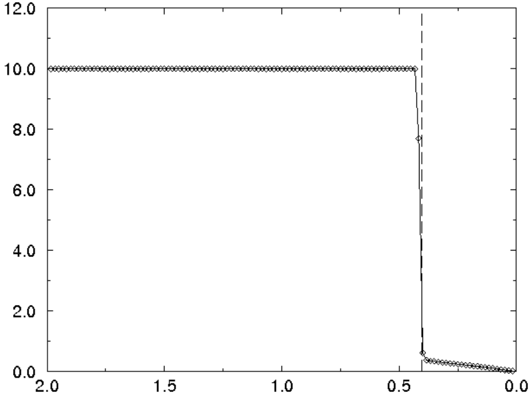
$$Ma = 3.0, \quad \alpha = 0^\circ$$



Cylinder

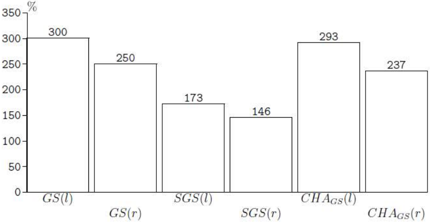
$$Ma = 10.0, \quad \alpha = 0^\circ$$

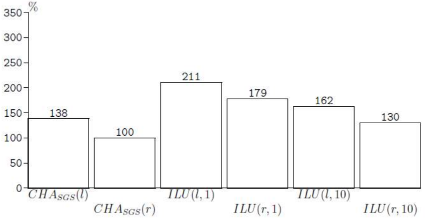




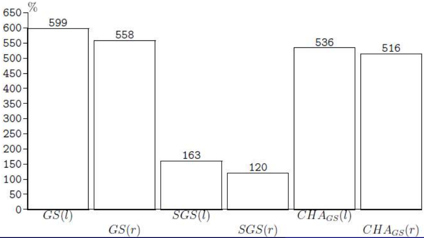
Comparison of Preconditioners

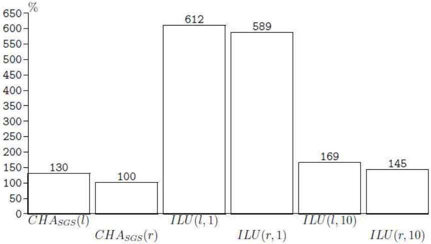
NACA0012 Airfoil $Ma = 3.0, \quad \alpha = 0°$





Cylinder $Ma = 10.0, \quad \alpha = 0°$



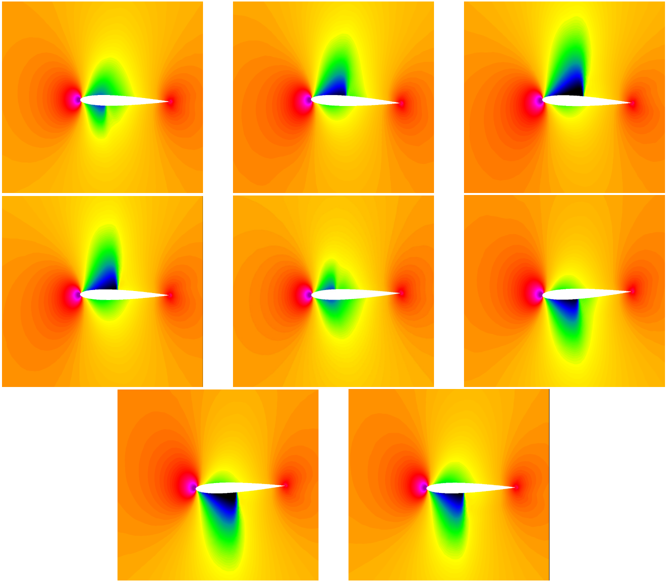


Instationär – Inviscid

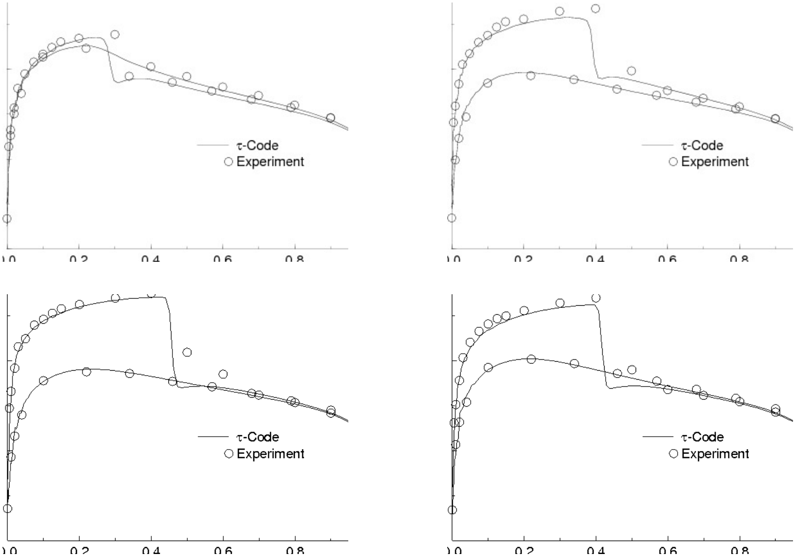
Airfoil	:	NACA0012
Freestream Mach number	:	0.755
Angle of attack in rest	:	0.016°
Amplitude	:	2.51°
Frequency	:	0.1628
Number of nodes	:	7141
Number of triangles	:	14005

Pitching NACA0012 Airfoil, Density distribution

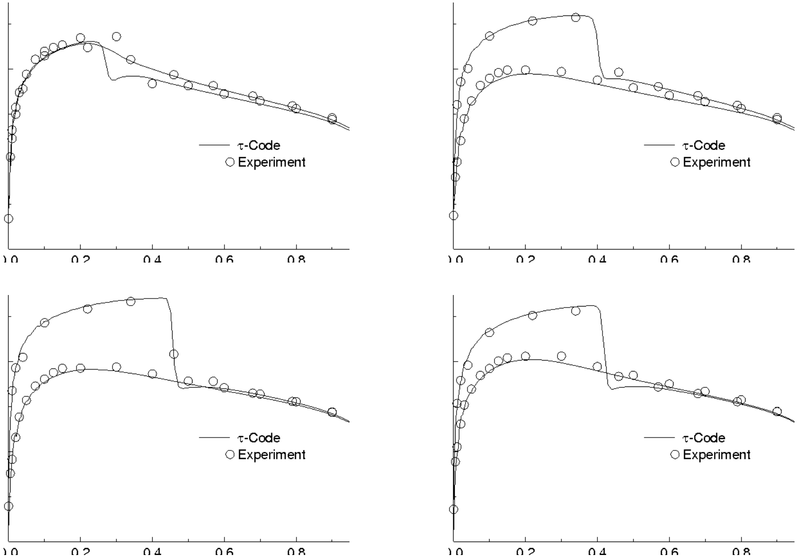
Ma = 0.755, $\alpha = 0.016^\circ + 2.51^\circ \sin kt,\ k = 0.1628,$ inviscid



Ma = 0.755, $\alpha = 0.016^\circ + 2.51^\circ \sin kt$, $k = 0.1628$, inviscid

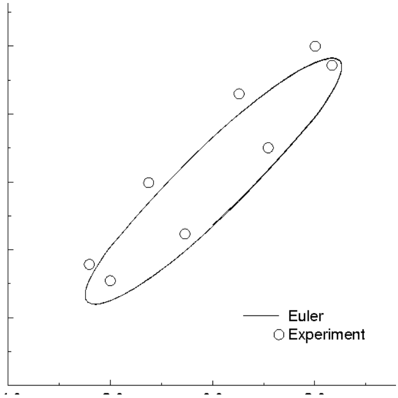


Ma = 0.755, $\alpha = 0.016^\circ + 2.51^\circ \sin kt$, $k = 0.1628$, inviscid

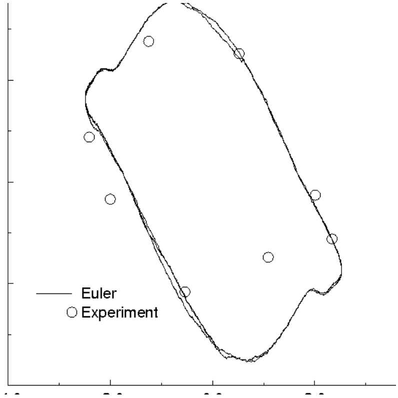


Pitching NACA0012 - Airfoil, Lift and Momentum

Lift coefficient



Momentum coefficient



Conclusion

- **Navier-Stokes Equations**
- **Finite Volumen Scheme**
- **AUSMDV-Riemannsolver for the convective fluxes**
- **Central scheme for the viscous fluxes**
- **Iterative system solver (BiCGSTAB)**
- **Preconditioner (ILU(r,p), $\mathrm{CHA}_{\mathrm{SGS}}$(r))**
- **Results**
  - very good results for inviscid test cases for both stationary and moving grids
  - very good results for viscous test cases
- **Application of implicit time stepping schemes**
  - Steady flow fields
  - Unsteady flow fields at low Mach number
  - Unsteady flow fields without fast moving shocks
  - Adaption CFL number