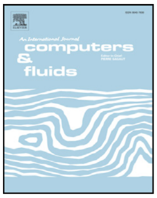
Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid



Development of a coupled matrix-free LU-SGS solver for turbulent compressible flows

Jiří Fürst

Dept. of Technical Mathematics, Faculty of Mechanical Engineering, Center of Advanced Aerospace Technology, Czech Technical University in Prague, Technicka Street 4, 16607, Prague 6, Czech Republic

ARTICLE INFO

Article history:
Received 31 August 2017
Revised 9 March 2018
Accepted 13 April 2018
Available online xxx

Keywords:
Finite volume method
CFD
Coupled solver
Implicit method
Compressible flows
Transonic flows

ABSTRACT

The paper deals with the development of the lower-upper symmetric Gauss–Seidel (LU-SGS) matrix-free finite volume solver for the simulation of compressible flows within the framework of the OpenFOAM package. The solver evaluates the convective fluxes using approximate Riemann solvers with limited piece-wise linear reconstructions whereas the viscous fluxes are approximated using a central scheme. The time evolution is realised through the backward differentiation formula of first or second order. The system of non-linear equations is then solved with the help of the matrix-free LU-SGS method. The developed solver is used to solve several flow problems and compared to a pressure-based segregated solver. Our numerical experiments indicate that the LU-SGS solver is more efficient for flows with higher Mach numbers and provides better resolution of shock waves. Moreover the LU-SGS solver benefits from the low memory footprint and does not use any problem specific setup.

1. Introduction

Computational Fluid Dynamics (CFD) is a fundamental tool for analysis and optimization of aerodynamic design. In the field of aeronautical engineering, turbomachinery or internal combustion engines, it is often possible to encounter high-speed flows with a significant compressibility effect. In these cases, non-linear phenomena such as shock waves with sudden changes in pressure, temperature, and velocity occur. These changes, based on the basic conservation laws, are tightly coupled.

Great efforts have been made in the development of numerical methods for the solution of compressible flows at various Mach numbers. For the low Mach number or incompressible flows, the so-called pressurebased methods represent an efficient solution procedure. Especially the segregated Semi-Implicit Pressure Linked Equations (SIMPLE) method [1] or its variants are commonly used for this kind of flows. Although being originally developed for incompressible or weakly compressible flows, the later developments [2–4] broadened the range of applicability of the pressure-based methods to high speed flows including transonic cases with shocks. Moreover, the development of coupled pressure-based methods [5,6] shows an important increase in the efficiency over the segregated approach at the cost of larger complexity of the code and larger memory requirements.

An alternative to the segregated pressure-based methods are the so-called coupled density-based solvers which use the concept of shock-capturing Godunov schemes using Riemann solvers, see e.g. [7]. This family of methods has been developed especially for convection dominated high speed flows and provides very robust schemes which are usually a preferred choice for transonic or hypersonic flows with shock waves, or flows with combustion.

OpenFOAM [8] is the widely used open source framework for the numerical solution of partial differential equations using the finite volume method. The C++ based library contains several CFD modules targeted mostly to the pressure-based segregated solvers. So far, there is only one coupled density-based solver using the central-upwind scheme of Kurganov and Tadmor [9] (the so-called *rhoCentralFoam*) in the standard OpenFOAM package. The FOAM-extend package (an extended fork of the OpenFOAM package) contains a specialized library developed originally under the name *densityBasedTurbo* by O. Born et al. [10]. The library provides a general framework for the development of coupled density-based solvers with several numerical fluxes including the Roe, the Rusanov, the HLLC, or the AUSM+up fluxes [7,11]. Moreover, the FOAM-extend package contains also a ready made solver called *dbnsTurbFoam* using this library. Unfortunately, both solvers, the *rhoCentralFoam* as well as the *dbnsTurbFoam*, use explicit methods for integration in time which leads to a strong stability limit on the time step and therefore both solvers become extremely inefficient in the case of flows at high Reynolds numbers with a mesh refinement in the vicinity of the wall.

E-mail address: Jiri.Furst@fs.cvut.cz

https://doi.org/10.1016/j.compfluid.2018.04.020
0045-7930/© 2018 Elsevier Ltd. All rights reserved.

The matrix-free lower-upper symmetric Gauss–Seidel (LU-SGS) scheme [12] is widely used because of its simplicity and very low memory requirements. The LU-SGS solver based on the OpenFOAM framework has been reported by Kim and Kim in [13] for the steady state case without giving enough details on the method. The article [14] describes the implementation of the LU-SGS scheme for inviscid compressible flows both for steady state and unsteady flows using the dual time stepping technique. In this article authors develop the scheme using the Steger-Warming flux splitting which needs the evaluation of the flux Jacobians and small matrix (5 by 5) operations. A similar method has been used also by Heyns et al. [15].

The current work is based on the description of the implementation of LU-SGS scheme given in the paper [14] and replaces the evaluation of the flux Jacobian by the finite difference approximation resulting in a true matrix-free method. Moreover, the method has been extended to the unsteady case assuming both the rotating reference frame or the arbitrary Lagrangian–Eulerian method.

2. The numerical method

The motion of a compressible gas is described by the following system of equations expressing conservation of mass, momentum, and energy

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{U}) &= 0,\\ \frac{\partial(\rho\mathbf{U})}{\partial t} + \nabla\cdot(\rho\mathbf{U}\otimes\mathbf{U}) + \nabla p &= \nabla\cdot\tau,\\ \frac{\partial(\rho E)}{\partial t} + \nabla\cdot[(\rho E + p)\mathbf{U}] &= \nabla\cdot(\tau\cdot\mathbf{U}) + \nabla\cdot(k\nabla T),\end{aligned}$$

where ρ is the density, $\mathbf{U}$ is the velocity vector, p is the pressure, $\boldsymbol{\tau}$ is the (effective) stress tensor, E is the specific total energy, k is the (effective) thermal conductivity, and T is the temperature. The system is closed by the equation of state for ideal gas $p/\rho = rT$ where r is the specific gas constant.

Spatial derivatives at the left hand side of the previous equations represent inviscid terms whereas the right hand side represents viscous terms. The above equations can be expressed in the integral form suitable for the finite volume approximation as follows (assuming a fixed control volume Ω at first)

$$\frac{d}{dt}\int_{\Omega}\mathbf{W}\,dV + \oint_{\partial\Omega}(\mathbb{F} - \mathbb{F}^{v})\cdot d\mathbf{S} = 0, \tag{1}$$

where $\mathbf{W} = [\rho, \rho\mathbf{U}, \rho E]$ is the vector of conservative variables and $\mathbb{F}$ and $\mathbb{F}^{v}$ represent inviscid and viscous fluxes.

The integral form of the system Eq. (1) is discretized in space using the collocated finite volume method (FVM) by taking

$$\mathbf{W}_i(t) = \frac{1}{|\Omega_i|}\iiint_{\Omega_i}\mathbf{W}(\mathbf{x}, t)\,dV,$$

where Ω_i is the mesh cell i. Hence Eq. (1) can be approximated as

$$|\Omega_i|\frac{d\mathbf{W}_i}{dt} = -\mathbf{R}(\mathbf{W})_i = -\sum_{j\in N_i}(\mathbb{F}_{ij} - \mathbb{F}^{v}_{ij})\cdot\mathbf{S}_{ij}, \tag{2}$$

where N_i is the set of indices of neighbor cells, $\mathbf{S}_{ij} = ||\mathbf{S}_{ij}||\mathbf{n}_{ij}$ is the scaled normal vector to the face shared between cells i and j oriented towards to cell j and $\mathbb{F}_{ij}$ and $\mathbb{F}^{v}_{ij}$ are the numerical fluxes. The viscous fluxes $\mathbb{F}^{v}_{ij}$ in 2 are discretized directly with the OpenFOAM built-in schemes (e.g. central scheme) and the discretization of inviscid fluxes $\mathbb{F}_{ij}$ is made using limited piece-wise linear reconstructions with the AUSM+up [11] or HLLC numerical fluxes [7].

2.1. The LU-SGS scheme for steady flows

In the case of steady state problem, the above devised system of non-linear ordinary differential equations can be solved using a local pseudo-time marching method. The time derivative is replaced by the first order backward difference using the local value of the time step

$$\begin{aligned}|\Omega_i|\frac{\mathbf{W}^{n+1}_i - \mathbf{W}^{n}_i}{\Delta t_i} &= -\mathbf{R}(\mathbf{W}^{n+1})_i \approx -\mathbf{R}(\mathbf{W}^{n})_i\\ &\quad - \sum_j \frac{\partial\mathbf{R}(\mathbf{W}^{n})_i}{\partial\mathbf{W}_j}\left(\mathbf{W}^{n+1}_j - \mathbf{W}^{n}_j\right),\end{aligned}$$

or denoting by $\Delta\mathbf{W}^{n} = \mathbf{W}^{n+1} - \mathbf{W}^{n}$

$$\sum_j\left(\frac{|\Omega_i|}{\Delta t_i}\mathbb{I} + \frac{\partial\mathbf{R}(\mathbf{W}^{n})_i}{\partial\mathbf{W}_j}\right)\Delta\mathbf{W}^{n}_j = -\mathbf{R}(\mathbf{W}^{n})_i. \tag{3}$$

This type of the implicit method was used successfully by many authors, see e.g. [12] or [16] in combination with standard linear equation solvers such as GMRES [17]. The main disadvantage of the straightforward approach is that one has to assemble the Jacobian matrix $\partial\mathbf{R}/\partial W$, which needs quite a huge amount of the memory. Together with the demands of the linear solver, the memory requirements can grow up to $50 - 100$ times the size of the array of unknowns. Such large memory requirements could be prohibitive and therefore we use the simple matrix-free lower-upper symmetric Gauss–Seidel method (LU-SGS) despite its lower performance in comparison with e.g. GMRES or multigrid methods.

In the first step, the Jacobian matrix $\partial\mathbf{R}/\partial\mathbf{W}$ is replaced by its lower order approximation calculated using thin-layer approximation for viscous terms and the first order approximation of convective terms with Rusanov flux

$$\mathbb{F}_{ij}\cdot\mathbf{S}_{ij} \approx \frac{1}{2}\left(\mathbb{F}(\mathbf{W}_i) + \mathbb{F}(\mathbf{W}_j)\right)\cdot\mathbf{S}_{ij} - \frac{\lambda_{ij}}{2}\left(\mathbf{W}_j - \mathbf{W}_i\right),$$

where λ_{ij} is the spectral radius of Jacobian of $\mathbb{F}\cdot\mathbf{S}$, i.e. $\lambda_{ij} = |\mathbf{u}_{ij}\cdot\mathbf{S}_{ij}| + a_{ij}||\mathbf{S}_{ij}||$ with $\mathbf{u}_{ij}$ being the velocity at the face between the cells i and j and a_{ij} is the sound speed. Thanks to the simplicity of the Rusanov flux the low order residual reduces to

$$\mathbf{R}(\mathbf{W})^{lo}_i = \frac{1}{2}\sum_{j\in N_i}\lambda^{*}_{ij}\mathbf{W}_i + \frac{1}{2}\sum_{j\in N_i}\left(\mathbb{F}(\mathbf{W}_j)\cdot\mathbf{S}_{ij} - \lambda^{*}_{ij}\mathbf{W}_j\right)$$

where λ^{*} includes both the spectral radii of Jacobians of convective terms and the thin-layer approximation of viscous terms, hence

$$\lambda^{*}_{ij} = \lambda_{ij} + \frac{||\mathbf{S}_{ij}||}{||\mathbf{x}_i - \mathbf{x}_j||}\max\left(\frac{4}{3\rho_{ij}}, \frac{\gamma}{\rho_{ij}}\right)\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right),$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the position vectors of centers of cells i and j, ρ_{ij} is the density at the face between cells i and j, γ is the ratio of specific heats, μ and μ_T are the molecular and turbulent viscosities, Pr is the Prandtl number, and Pr_T is the turbulent Prandtl number, see [12]. Finally the product of Jacobians with $\Delta\mathbf{W}$ is approximated with finite differences and the system of equations is solved with the LU-SGS method.

Let L and U denote the sets of cells belonging to lower and upper part of the matrix:

$$\begin{aligned}L_i &= \{j \in N_i : j < i\},\\ U_i &= \{j \in N_i : j > i\}.\end{aligned}$$

Then the matrix-free LU-SGS scheme can be written using the following two step procedure:

$$D_i\Delta\mathbf{W}^{(1)}_i = -\mathbf{R}_i - \frac{1}{2}\sum_{j\in L_i}\left[\Delta\mathbb{F}^{(1)}_j\cdot\mathbf{S}_{ij} - \lambda^{*}_{ij}\Delta\mathbf{W}^{(1)}_j\right], \tag{4}$$

Please cite this article as: J. Fürst, Development of a coupled matrix-free LU-SGS solver for turbulent compressible flows, Computers and Fluids (2018), https://doi.org/10.1016/j.compfluid.2018.04.020

$$D_i\Delta\mathbf{W}_i = D_i\Delta\mathbf{W}_i^{(1)} - \frac{1}{2}\sum_{j\in U_i}\left[\Delta\mathbb{F}_j\cdot\mathbf{S}_{ij} - \lambda^*_{ij}\Delta\mathbf{W}_j\right]. \tag{5}$$

Here

$$\begin{aligned}\Delta\mathbf{W}_i^{(1)} &= \mathbf{W}_i^{(1)} - \mathbf{W}_i^n,\\ \Delta\mathbb{F}_i^{(1)} &= \mathbb{F}(\mathbf{W}_i^{(1)}) - \mathbb{F}(\mathbf{W}_i^n),\\ \Delta\mathbb{F}_i &= \mathbb{F}(\mathbf{W}_i^{n+1}) - \mathbb{F}(\mathbf{W}_i^n),\\ D_i &= \left(\frac{|\Omega_i|}{\Delta t_i} + \frac{1}{2}\sum_{j\in N_i}\lambda^*_{ij}\right).\end{aligned}$$

Note that the scheme needs only one number per cell for storing D_i and two working arrays for $\Delta\mathbf{W}$ and $\mathbf{R}$ which is much less than the requirements of classical matrix-based implicit methods.

2.2. Dual time stepping for transient simulations

In the case of transient flows a dual time stepping procedure is used [18], i.e. instead of solving the Eq. (2) one looks in each (physical) time step t^n for a stationary solution (in the pseudo-time τ) of the following problem

$$|\Omega_i|\frac{d\mathbf{W}_i}{d\tau} = -\mathbf{R}(\mathbf{W})_i - |\Omega_i|\frac{d\mathbf{W}_i}{dt}.$$

Using the local values of the pseudo-time τ one arrives to the system of equations for pseudo-time increments similar to (3)

$$\sum_j\left(\frac{|\Omega_i|}{\Delta\tau_i}\mathbb{I} + \frac{\partial\mathbf{R}(\mathbf{W})_i}{\partial\mathbf{W}_j}\right)\Delta\mathbf{W}_j^\nu$$
$$= -\mathbf{R}(\mathbf{W}^\nu)_i - |\Omega_i|\frac{3\mathbf{W}_i^\nu - 4\mathbf{W}_i^n + \mathbf{W}_i^{n-1}}{2\Delta t},$$

where $\Delta\mathbf{W}_i^\nu = \mathbf{W}_i^{\nu+1} - \mathbf{W}_i^\nu$ with ν being a pseudo-time iteration index and the physical time derivative is approximated with the second order backward differentiation formula. One can see that the system of equations has similar form to Eq. (3) and therefore one can merge both methods for steady and unsteady cases into one by replacing $\mathbf{R}(\mathbf{W})$ in (3) or in (4) and (5) with

$$\mathbf{R}^*(\mathbf{W}^\nu)_i = \mathbf{R}(\mathbf{W}^\nu)_i + |\Omega_i|\frac{3\mathbf{W}_i^\nu - 4\mathbf{W}_i^n + \mathbf{W}_i^{n-1}}{2\Delta t}.$$

2.3. The LU-SGS method in ALE formulation

The extension to the unsteady case with moving meshes is achieved using the so-called arbitrary Lagrangian–Eulerian formulation (ALE), [12]. The spatial approximation is replaced by its ALE variant

$$\mathbf{R}(\mathbf{W})_i^{ALE} = \sum_j\left(\mathbb{F}_{ij}^{ALE} - \mathbb{F}_{ij}^\nu\right)\cdot\mathbf{S}_{ij}^{n+1},$$

where $\mathbb{F}_{ij}^{ALE}$ is the ALE numerical flux including the mesh velocity $\mathbf{v}_{ij}$ approximating the physical flux, see e.g. [12]

$$\mathbb{F}^{ALE}(\mathbf{W}) = \mathbb{F}(\mathbf{W}) - \mathbf{W}\otimes\mathbf{v}.$$

The unsteady ALE residual is then

$$\mathbf{R}^{*,ALE}(\mathbf{W}^\nu)_i = \mathbf{R}^{ALE}(\mathbf{W}^\nu)_i - \frac{3|\Omega_i^{n+1}|\mathbf{W}_i^\nu - 4|\Omega_i^n|\mathbf{W}_i^n + |\Omega_i^{n-1}|\mathbf{W}_i^{n-1}}{2\Delta t}.$$

The approximation of the Jacobian of the residual is made again using the Rusanov flux which leads to the scheme (4)–(5) with λ_{ij} replaced by $\lambda_{ij}^{ALE} = |(\mathbf{u}_{ij} - \mathbf{v}_{ij})\cdot\mathbf{S}_{ij}| + a_{ij}||\mathbf{S}_{ij}||$ and $\mathbf{R}$ by $\mathbf{R}^{*,\,ALE}$.

2.4. The LU-SGS method in rotating reference frame

A very useful approach for solving flows in turbomachinery is the use of a reference frame rotating with a constant angular velocity $\boldsymbol{\omega}$. We deal with rotating frame of reference using a modified ALE approach. It means that we are solving for the density, the momentum based on the absolute velocity, and the total energy. The modification with respect to ALE formulation consists of adding $\rho\boldsymbol{\omega}\times\mathbf{u}$ to the momentum part of the residual. The other terms (centrifugal and Coriolis forces) follow directly from the ALE formulation. Hence the steady state rotating reference frame formulation uses the mesh velocity $\mathbf{v} = \omega\times\mathbf{r}$ and uses the modified residual

$$\mathbf{R}^{*,rot}(\mathbf{W}^n)_i = \mathbf{R}^{ALE}(\mathbf{W}^n)_i + |\Omega_i|[0, \rho\omega\times\mathbf{u}, 0]^T.$$

With the above mentioned formulations we were able to develop two variants of solvers, first one for steady and transient simulations in stationary or rotating reference frame, and the second one for transient simulations using deforming meshes, using common code base.

3. Results

The following section shows some numerical results obtained with the LU-SGS method. The case of compressible inviscid flows through a channel with a bump is used for the comparison of the LU-SGS scheme with a segregated pressure-based scheme both in terms of the resolution of shock waves and the CPU time for a single core run using the Intel Xeon E3-1241v3 CPU at 3.5 GHz. Next a viscous turbulent flow over an axisymmetric bump and the unsteady flow over oscillating NACA-0012 profile is presented.

3.1. Inviscid 2D flows through a channel

The first case is an inviscid flow through a 3 m long channel of a width 1 m with 1 m long circular arc bump at the lower wall. The height of the bump is 0.1 m for subsonic and transonic cases and 0.04 m for supersonic cases. This problem has been studied by many authors in order to test the accuracy and the stability of numerical methods, see e.g. [2,19,20]. Calculations are carried out for all regimes using the LU-SGS method in combination with the AUSM+up flux and the piece-wise linear reconstructions with the Venkatakrishnan's limiter [21]. The results are compared to results obtained with the segregated pressure-based solver with a SIMPLE-type pressure-velocity coupling method with limited second order approximation of convective fluxes (the *rhoSimpleFoam* solver). The flow fields are calculated using three consecutively refined meshes with 50 × 150, 100 × 300 and 200 × 600 quadrilateral cells.

In order to quantitatively judge the results the entropy production in the domain is evaluated by calculating the difference of the entropy fluxes through the inlet and outlet, hence

$$\Delta s = \frac{\int_0^1\rho(x_2,y)u(x_2,y)s(x_2,y)\,dy - \int_0^1\rho(x_1,y)u(x_1,y)s(x_1,y)\,dy}{\int_0^1\rho(x_2,y)u(x_2,y)\,dy},$$

where x_1=0 m corresponds to the inlet boundary and x_2=3 m to the outlet boundary and $s = c_p\ln(T) - r\ln(p)$ is the specific entropy. The specific heat capacity at constant pressure c_p=1004.7 J kg^{-1}K^{-1} is used together with the specific gas constant r = 287.05 J kg^{-1}K^{-1}. The entropy production is evaluated for all three meshes giving Δs_c for the coarse mesh, Δs_m for the medium mesh, and Δs_f for the fine mesh. Using this three values of entropy production, the estimated order of accuracy is calculated as

$$q = \frac{\ln|\Delta s_m - \Delta s_c| - \ln|\Delta s_f - \Delta s_m|}{\ln(2)},$$

Please cite this article as: J. Fürst, Development of a coupled matrix-free LU-SGS solver for turbulent compressible flows, Computers and Fluids (2018), https://doi.org/10.1016/j.compfluid.2018.04.020

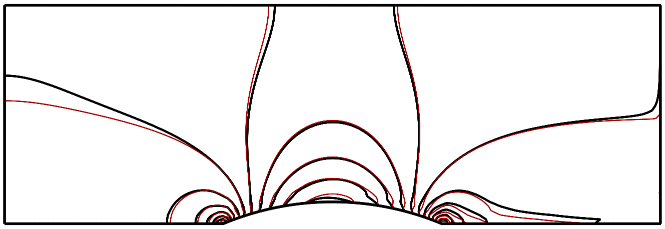


Fig. 1. Isolines of the Mach number for $M_{2i} = 0.1$, $\Delta M = 0.005$, (LU-SGS plotted by thick black line, SIMPLE by thin red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

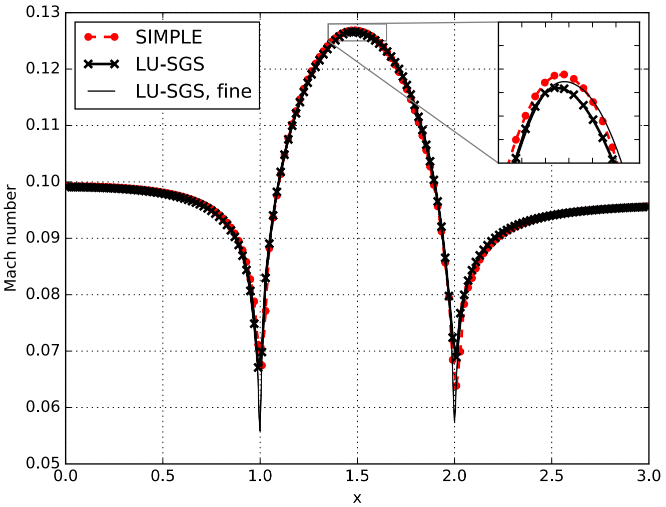


Fig. 2. Mach number distribution along the lower wall, $M_{2i} = 0.1$.

then the estimate of the entropy jump Δs^* is evaluated using the Richardson extrapolation

$$\Delta s^\star = \Delta s_f + \frac{\Delta s_f - \Delta s_m}{2^q - 1},$$

and finally the grid convergence index (GCI) is calculated

$$GCI = F_S|\Delta s^\star - \Delta s_f|$$

with $F_s = 1.25$ (see [22]). Note that the GCI is calculated using absolute values of Δs instead of more often used relative errors due to the fact that Δs should be zero in the subsonic case.

3.1.1. The low Mach number flow at $M_{2i} = 0.1$

The subsonic flow regime is characterized by the isentropic outlet Mach number M_{2i}=0.1. At the inlet the total pressure $p_{tot} = 1 \times 10^5$ Pa, total temperature $T_{tot} = 288.15$ K and flow direction parallel to x-axis is prescribed. The static pressure $p_2 = p_{tot}(1 + \frac{\gamma-1}{2}M_{2i}^2)^{\frac{\gamma}{1-\gamma}} \approx 0.993 \times 10^5$ Pa is prescribed at the outlet. The slip condition for the velocity together with homogeneous Neumann conditions for the pressure and temperature are used on both the upper and lower walls.

Figs. 1 and 2 show the isolines of the Mach number and the distribution of Mach number along the lower wall obtained with the LU-SGS method and with the SIMPLE method on the coarse mesh. Fig. 2 includes also the fine mesh solution obtained with the LU-SGS method. One can see that both methods give very similar results and that the Mach number at the lower wall is resolved very well even with the coarse mesh. The Table 1 summarizes quantitatively the calculated entropy productions. Note that in the inviscid subsonic case the flow should be isentropic and the true value of Δs should be zero. The entire entropy production is therefore due to false numerical dissipation. One can see here that the SIMPLE-based method suffers less from numerical diffusion compared to the LU-SGS method. The estimated order of accuracy is less than 2 for both methods which is probably caused by the use of limiters.

Table 1
Mesh dependence study for inviscid flow through a channel at $M_{2i} = 0:1$.

	LU-SGS	SIMPLE
Δs_c (J/kg/K)	0.0120	0.0048
Δs_m (J/kg/K)	0.0049	0.0020
Δs_f (J/kg/K)	0.0022	0.0008
Δs^* (J/kg/K)	0.0005	-7.5×10^5
Order	1.4062	1.2735
GCI (J/kg/K)	0.0020	0.0011

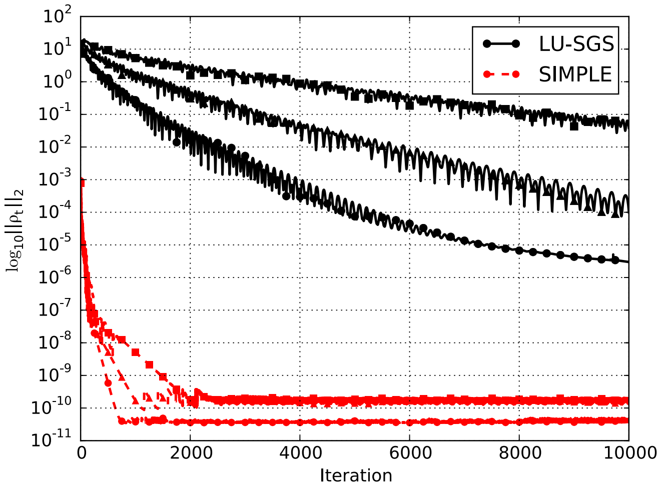


Fig. 3. Convergence history, subsonic case with $M_{2i} = 0.1$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

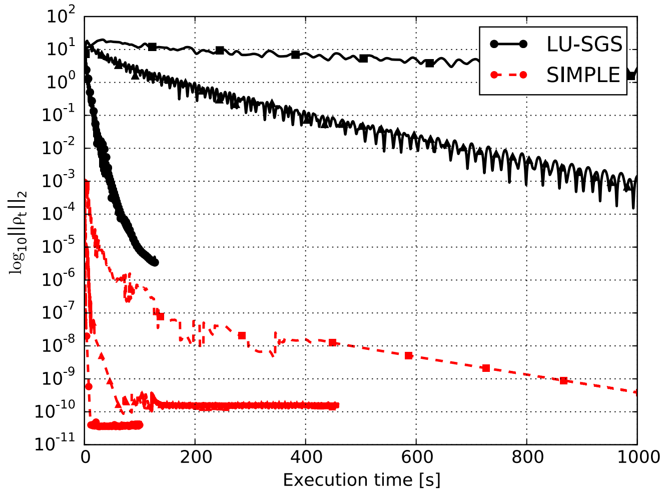


Fig. 4. Convergence history with respect to CPU time, subsonic case with $M_{2i} = 0.1$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

Figs. 3 and 4 depict the history of convergence to steady state. One can see that the SIMPLE-based solver is much more efficient in this case both in terms of number of iterations and computational time. This conclusion is supported also by the grid convergence index, see the Table 1.

3.1.2. The transonic flow at $M_{2i} = 0.675$

The next case is the flow through the same channel with the isentropic outlet Mach number set to M_{2i}=0.675 giving the static

Please cite this article as: J. Fürst, Development of a coupled matrix-free LU-SGS solver for turbulent compressible flows, Computers and Fluids (2018), https://doi.org/10.1016/j.compfluid.2018.04.020

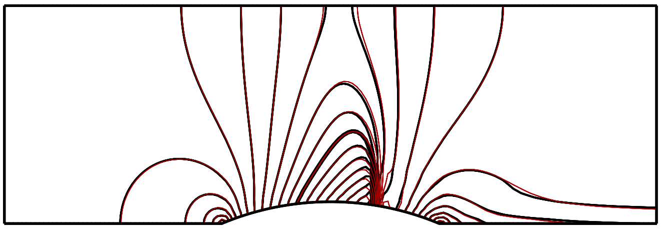


Fig. 5. Isolines of the Mach number for $M_{2i} = 0.675$, $\Delta M = 0.05$, (LU-SGS plotted by thick black line, SIMPLE by thin red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

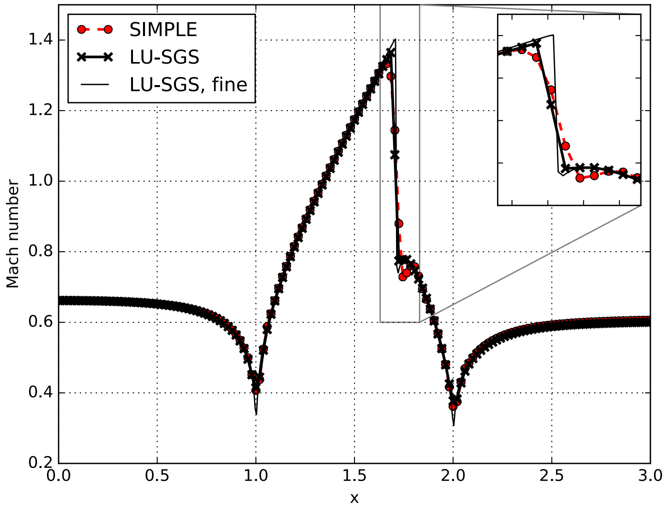


Fig. 6. Mach number distribution along the lower wall, $M_{2i} = 0.675$.

outlet pressure $p_2 = 0.737 \times 10^5$ Pa. Other boundary conditions are set identically to the previous case. The increased velocity creates a local supersonic domain closed by a shock wave. The flow is strongly influenced by the compressibility effects and therefore one has to include also the density variations in the SIMPLE method (the so-called transonic version of SIMPLE solver). That's why the SIMPLE solver needs completely different setup than in the subsonic case including specifically tuned relaxation factors. This is not the case of LU-SGS solver which uses the same setup for all regimes.

Despite the difficulties in setting-up the SIMPLE solver one can see that the results obtained with both solvers are very similar and correspond to results of other researchers e.g. [19], see Figs. 5 and 6. Nevertheless a close investigation shows that the LU-SGS solver is able to capture the shock wave within one cell whereas the SIMPLE solver typically smears the shock over 4–5 cells.

Figs. 7 and 8 show the convergence history in the transonic case. One can see that the LU-SGS scheme converges faster to the steady state both in terms of number of iterations and the CPU time. Even the quantitative comparison using GCI implies better grid convergence of the entropy production for LU-SGS method, see the Table 2.

3.1.3. The supersonic flow at $M_1 = 1.65$

The third case is the flow at the inlet Mach number M_1=1.65. The flow is supersonic in the whole domain and therefore a different set of inlet and outlet boundary conditions has to be used. All flow quantities (static pressure, static temperature, and velocity components) are prescribed at the inlet with values corresponding to the total pressure $p_{tot} = 1 \times$ Pa, the total temperature T_{tot}= 288.15 K, and the Mach number M_1=1.65.

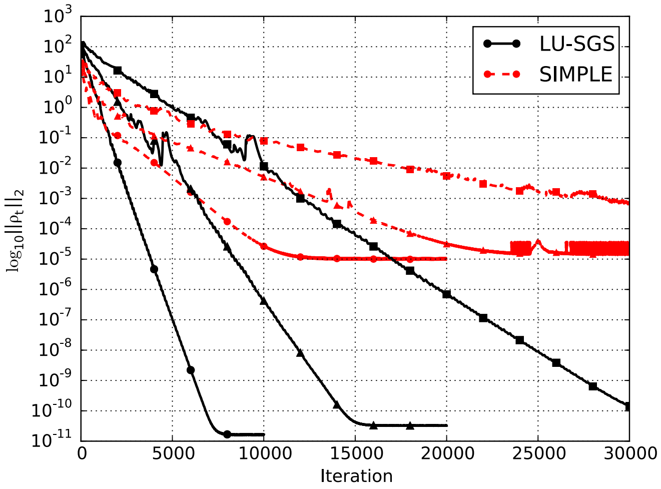


Fig. 7. Convergence history, transonic case with $M_{2i} = 0.675$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

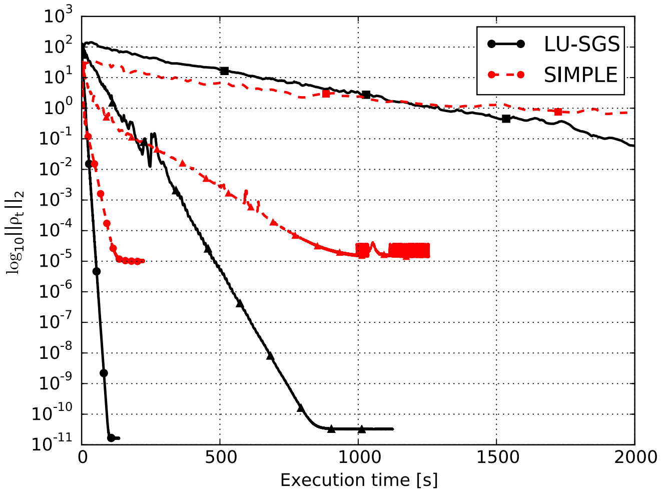


Fig. 8. Convergence history with respect to CPU time, transonic case with $M_{2i} = 0.675$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

Table 2
Mesh dependence study for inviscid flow through a channel at $M_{2i} = 0.675$.

	LU-SGS	SIMPLE
Δs_c (J/kg/K)	1.115	1.112
Δs_m (J/kg/K)	1.0224	1.0280
Δs_f (J/kg/K)	1.0031	1.0065
Δs^* (J/kg/K)	0.9981	0.9991
order	2.2673	1.9652
GCI (J/kg/K)	0.0063	0.0093

The solution obtained with the LU-SGS scheme is oscillation-free whereas the SIMPLE method produces a lot of non-physical oscillations behind the shock (see Fig. 9, 12 at $x = 1m$ and $x = 2m$). The convergence to the steady state solution (Figs. 10 and 11) shows that the LU-SGS is much faster and the SIMPLE method stalls. The absolute value of GCI is relatively large (see Table 3). Nevertheless, it is approximately within 5% of total entropy production.

3.2. Viscous turbulent 2D flows over a bump

For the assessment of the properties of the LU-SGS method for the case of complex compressible turbulent flows, the two-

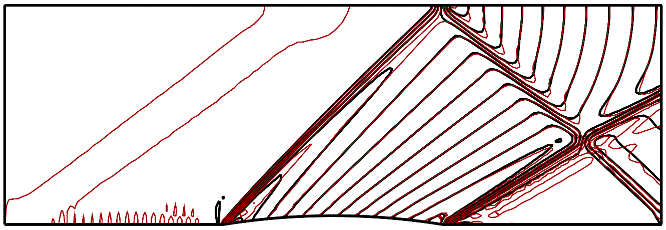


Fig. 9. Isolines of the Mach number for $M_1 = 1.65$, $\Delta M = 0.05$, (LU-SGS plotted by thick black line, SIMPLE by thin red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

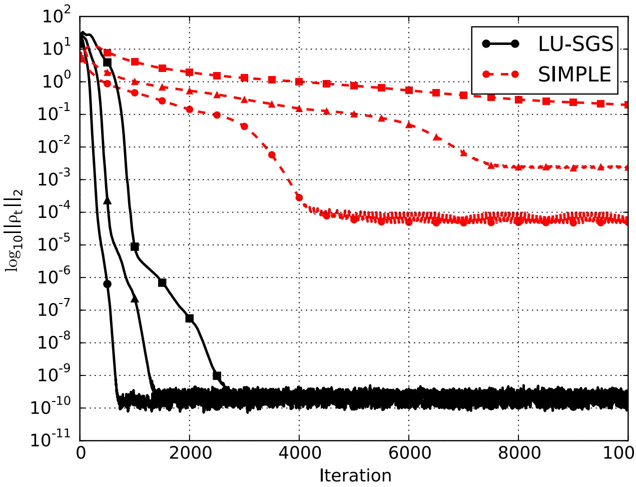


Fig. 10. Convergence history, supersonic case with $M_1 = 1.65$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

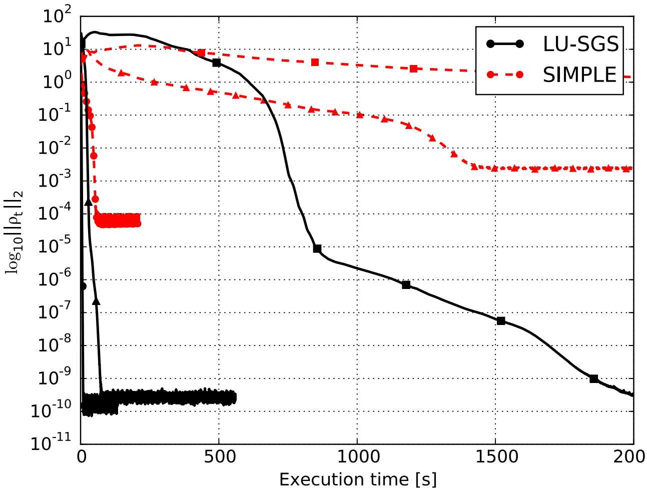


Fig. 11. Convergence history with respect to CPU time, supersonic case with $M_1 = 1.65$, (○ – coarse mesh, △ – medium mesh, □ – fine mesh).

Table 3
Mesh dependence study for inviscid flow through a channel at $M_1 = 1.65$.

	LU-SGS	SIMPLE
Δs_c (J/kg/K)	3.2163	3.1329
Δs_m (J/kg/K)	3.1329	3.0875
Δs_f (J/kg/K)	3.0764	3.0507
Δs^* (J/kg/K)	2.9438	2.8941
order	0.5194	0.3045
GCI (J/kg/K)	0.1658	0.1957

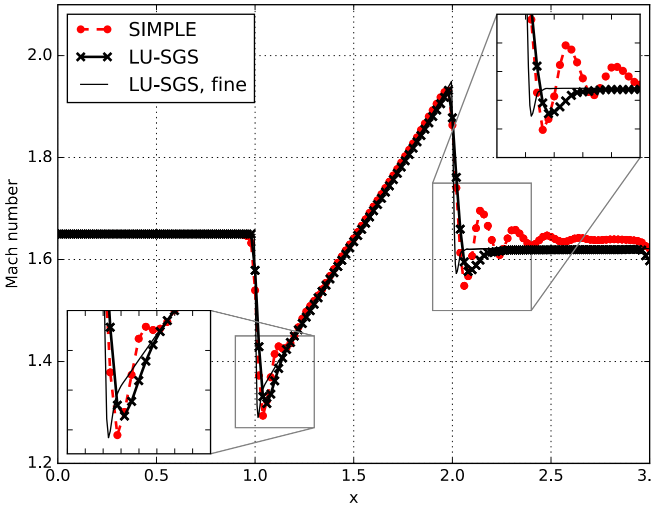


Fig. 12. Mach number distribution along the lower wall, $M_1 = 1.65$.

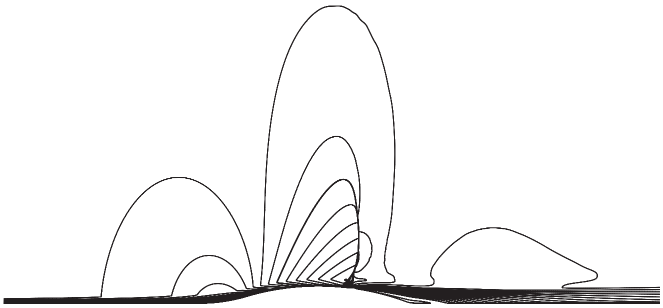


Fig. 13. Isolines of the Mach number for axisymmetric transonic bump calculation, $\Delta M = 0.05$, sonic line in bold.

dimensional flow over a bump is selected. The case was experimentally studied by Bachalo et al. [23], see also https://turbmodels.larc.nasa.gov/axibump_val.html. The domain consists of a gap between two cylinders with inner radius $r_{inner}/c = 0.375$ and outer radius $r_{outer}/c = 4.0$ where c is the chord length of the bump. The circular arc shaped bump with the height $h/c = 0.09375$ is located at the inner cylinder with a fillet at the leading edge at $x/c = 0$ and a small fillet at the trailing edge (see NASA website). The computational domain starts at $x/c = -3.2$ and ends at $x/c = 4.4$. The flow is assumed as axisymmetric and therefore the mesh has one cell in circumferential direction spanning 1%.

The flow regime is characterized by the outlet isentropic Mach number $M_{2i} = 0.875$ and the Reynolds number $Re_c = 2.763 \times 10^6$. The flow is turbulent with the inlet turbulence intensity $Tu = 0.0089\%$ and the free-stream ratio of the eddy to molecular viscosity is set to $\mu_t/\mu = 0.009$ which correspond to data used by NASA for validation of CFL3D code.

The solution is calculated using a structured mesh with 720 × 320 × 1 cells provided at the NASA website. The mesh is refined in the vicinity of the inner cylinder up to $y_1^+ \approx 0.1$ and there is also a refinement in the region where one expects the shock wave.

The solution is obtained with the LU-SGS method using AUSM+up fluxes for convective terms. The method is coupled with the Menter's SST turbulence model [24]. Fig. 13 shows the isolines of the Mach number in the vicinity of the bump. One can see the local supersonic region closed by normal shock and the flow separation due to the interaction of the shock wave with the boundary layer.

Fig. 14 shows the distribution of the pressure coefficient $c_p = 2(p - p_\infty)/(\rho_\infty U_\infty^2)$ along the inner cylinder (reference values with

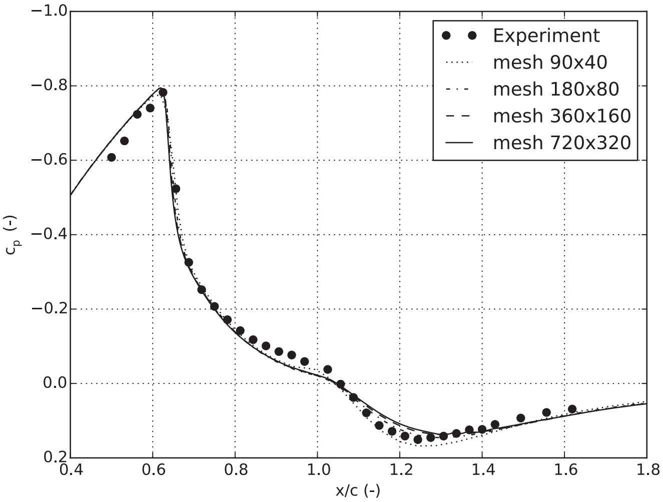


Fig. 14. Distribution of the pressure coefficient c_p along the inner cylinder of the axisymmetric transonic bump.

subscript ∞ correspond to inlet static conditions). Lines represent the solution obtained with four different meshes starting with the coarse one consisting of 90 × 40 × 1 cells up to the fine mesh with 720 × 320 × 1 cells, points represent the experimental data from [23]. One can see that the LU-SGS solver predicts correct position of the shock wave even with the coarse mesh. The most pronounced difference between the pressure fields obtained at different meshes is in the region downstream to the trailing edge ($x/c \approx 1.2 - 1.4$), nevertheless the difference decreases with mesh refinement.

Note that we were not able to find a stable setup for the *rhoSimpleFoam* solver working with all meshes and therefore we do not include the comparison with segregated solver in this case.

The results obtained with the LU-SGS method correspond very well to the results obtained with CFL3D and FUN3D code published by NASA, see [25] including the differences between simulation results and experimental data.

3.3. The unsteady transonic flow over an oscillating NACA-0012 profile

Finally, the performance of the LU-SGS solver for simulations of flows with moving mesh is examined for the case of transonic flows over an oscillating NACA-0012 profile. The case was experimentally studied by Landon, see [26], case 5. The free-stream Mach number is $M_\infty = 0.755$, the Reynolds number is $Re = 5 \times 10^6$ and the profile performs a harmonic oscillatory motion with mean incidence $\alpha_m = 0.016\%$, pitch amplitude $\alpha_o = 2.51\%$ and reduced frequency $k = \omega c/2U_\infty = 0.0814$ (here c is the profile chord) given as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t).$$

The simulation is performed using an O-type mesh with 249×99 quadrilateral cells with the free-stream boundary at the distance approximately 20 chords from the profile. The average size of the first cell in wall normal direction is approximately $y_1^+ \approx 0.7$.

Two calculations are made, the first one using the LU-SGS scheme in ALE formulation with dual time stepping and the second one using segregated pressure-based solver based on PIMPLE loop (*rhoPimpleDyMFoam*) from OpenFOAM package. Spatial terms are discretized in both cases with limited second order schemes and the time derivatives with the second order backward differentiation formula. A fixed (physical) time step is chosen as $\Delta t = T/1000$, where T is the period of the oscillation.

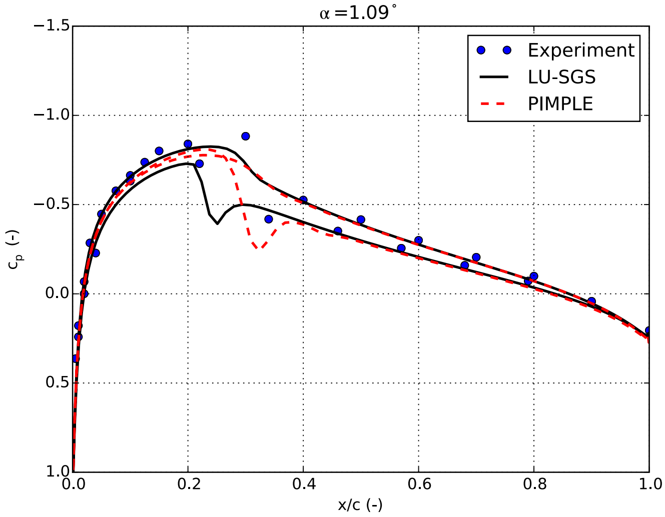


Fig. 15. Distribution of the pressure coefficient c_p for the oscillating NACA-0012 profile, angle of attack $\alpha = 1.09\%$.

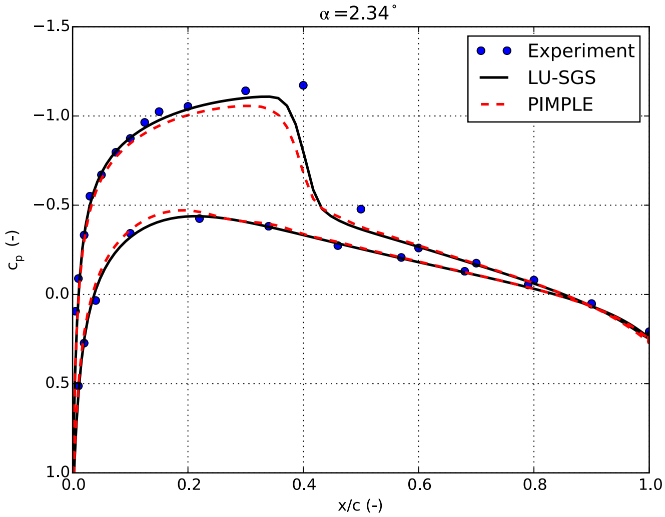


Fig. 16. Distribution of the pressure coefficient c_p for the oscillating NACA-0012 profile, angle of attack $\alpha = 2.34\%$.

Figs. 15 and 16 show the comparison of the pressure coefficient $c_p = 2(p - p_\infty)/(\rho_\infty U_\infty^2)$ at two different angles of attack with the experimental data [26]. The differences between the results (see especially Fig. 15) obtained with PIMPLE and LU-SGS method using the same mesh are caused mainly by different approximation of convective terms in both methods. Despite these differences the results provided by the two methods qualitatively agree with the experimental data. Figs. 17 and 18 show hysteresis loops of the lift and momentum coefficients c_y and c_M versus the angle of attack $\alpha(t)$. The comparison with experimental data shows again that both methods give qualitatively correct results. The LU-SGS scheme requires approximately 40 dual time iterations in each time step in order to reduce the residual by 5 orders of magnitude. The PIMPLE method requires about 70 outer correctors for the same tolerance. Expressing it in terms of single-core CPU time, the LU-SGS method requires approximately 55 min per one pitching period whereas the PIMPLE method requires 137 min.

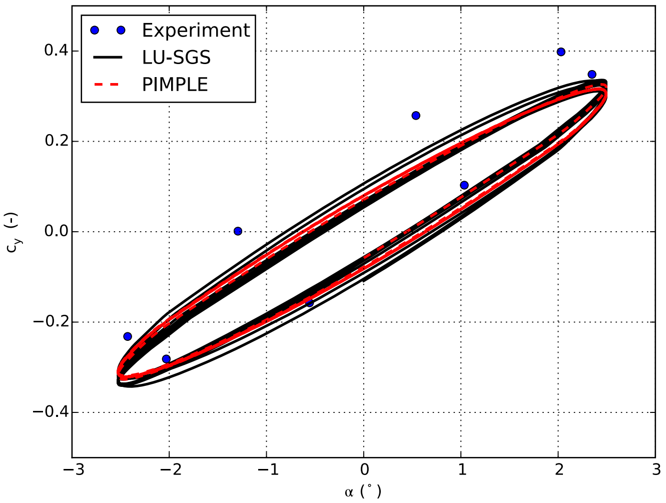


Fig. 17. The lift coefficient c_y versus angle of attack α for the case of the oscillating NACA-0012 profile.

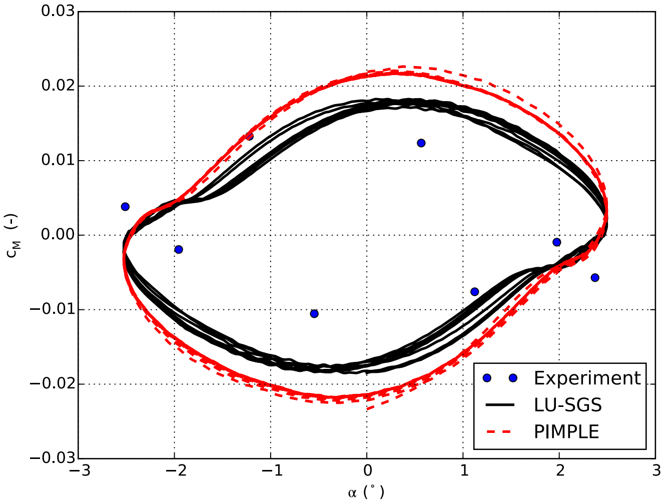


Fig. 18. The pitching momentum coefficient c_M versus the angle of attack α for the case of the oscillating NACA-0012 profile.

4. Concluding remarks

The implicit matrix-free LU-SGS density-based scheme has been successfully implemented both for stationary and transient compressible flows. Though OpenFOAM already contains pressure-based solvers which are able to solve similar problems, the LU-SGS scheme is more efficient for flows with higher Mach numbers and provides sharp and oscillation-free resolution of shock waves. Moreover, the LU-SGS scheme doesn't need any problem specific setup or relaxation.

Acknowledgments

Authors acknowledge support from the ESIF, EU Operational Programme Research, Development and Education, and from the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826), Faculty of Mechanical Engineering, Czech Technical University in Prague.

References

[1] Patankar S. Numerical heat transfer and fluid flow. CRC Press; 1980.
[2] Demirdžić I, Lilek Ž, Perić M. A collocated finite volume method for predicting flows at all speeds. Int J Numer Methods Fluids 1993;16(January):1029–50. doi:10.1002/fld.1650161202.
[3] Moukalled F, Darwish M. A high-resolution pressure-Based algorithm for fluid flow at all speeds. J Comput Phys 2001;168(1):101–30. doi:10.1006/jcph.2000.6683.
[4] Nerinckx K, Vierendeels J, Dick E. Three Mach-uniform algorithms. J Comput Appl Math 2008;215(2):521–7. doi:10.1016/j.cam.2006.03.052.
[5] Chen ZJ, Marella SV, Przekwas AJ. A finite volume method of pressure-based coupled solver for incompressible / compressible flows, In 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition (Vol. M, pp. 1–18), 2009, https://doi.org/10.2514/6.2009-600.
[6] Darwish M, Moukalled F. A fully coupled Navier–Stokes solver for fluid flow at all speeds. Numer Heat Transfer, Part B 2014;65(5):410–44. doi:10.1080/10407790.2013.869102.
[7] Toro EF. The HLL and HLLC Riemann solvers. In: Riemann solvers and numerical methods for fluid dynamics. Berlin, Heidelberg: Springer Berlin Heidelberg; 1997. p. 293–311. doi:10.1007/978-3-662-03490-3_10.
[8] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620. doi:10.1063/1.168744.
[9] Kurganov A, Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection diffusion equations. J Comput Phys 2000;160(1):241–82. doi:10.1006/jcph.2000.6459.
[10] Borm O, Jemcov A, Kau H. Density based Navier–Stokes solver for transonic flows. In: Proceedings of the 6th OpenFOAM workshop. PennStats University; 2011.
[11] Liou MS. A sequel to AUSM, Part II: AUSM+-up for all speeds. J Comput Phys 2006;214(1):137–70. doi:10.1016/j.jcp.2005.09.020.
[12] Blazek J. Computational fluid dynamics : principles and applications. Butterworth-Heinemann; 2015.
[13] Kim J, Kim K. A development and verification of density based implicit Navier–Stokes Solver using LU-SGS algorithm in OpenFOAM. In: 29th Congress of the international council of the aeronautical sciences, St. Petersburg; 2014. p. 1–9.
[14] Shen C, Xia X-l, Wang Y-z, Yu F, Jiao Z-w. Implementation of density-based implicit LU-SGS solver in the framework of OpenFOAM. Adv Eng Software 2016;91:80–8. doi:10.1016/j.advengsoft.2015.10.007.
[15] Heyns JA, Oxtoby OF, Steenkamp A. Modelling high-speed flow using a matrix-free coupled solver. 9th OpenFOAM workshop. October; 2014.
[16] Fürst J. A weighted least square scheme for compressible flows. Flow, Turbul Combust 2006;76(4):331–42. doi:10.1007/s10494-006-9021-y.
[17] Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986;7(3):856–69. doi:10.1137/0907058.
[18] Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. 10th Computational fluid dynamics conference; 1991. doi:10.2514/6.1991-1596.
[19] Xisto CM, Páscoa JC, Oliveira PJ, Nicolini DA. A hybrid pressure-density-based algorithm for the Euler equations at all Mach number regimes. Int J Numer Methods Fluids 2012;70(8):961–76. doi:10.1002/fld.2722.
[20] Hirsch C. Numerical computation of internal and external flows: the fundamentals of computational fluid dynamics; 2007. doi:10.1016/B978-0-7506-6594-0.X5037-1.
[21] Venkatakrishnan V. On the accuracy of limiters and convergence to steady state solutions. In: 31st Aerospace sciences meeting & exhibit; 1993. p. 1–10. doi:10.2514/6.1993-880.
[22] Roache PJ. Verification and validation in computational science and engineering. In: Computing in science engineering; 1998. p. 8–9.
[23] Johnson DA, Horstman CC, Bachalo WD. Comparison between experiment and prediction for a transonic turbulent separated flow. AIAA J 1982;20(6):737–44. doi:10.2514/3.51130.
[24] Menter FR, Kuntz M, Langtry R. Ten years of industrial experience with the SST turbulence model. Turbul Heat Mass Transfer 2003;4:625–32.
[25] Axisymmetric transonic bump, https://turbmodels.larc.nasa.gov/axibump_val.html Accessed: 2017-08-17.
[26] Landon RH. NACA 0012. oscillating and transient pitching, Compendium of unsteady aerodynamic measurements. AGARD R-702; 1982.