EasyChair Preprint

№ 1007

Two implicit schemes for compressible and turbulent fluid flow's modeling

Fayssal Benkahldoun, Jaroslav Fořt and Jean-Baptiste Montavon

May 25, 2019

Two implicit schemes for compressible and turbulent fluid flow's modeling

CAISAM Conference

Belkhandoun, F., Fort, J., Montavon, J.B.

April 2019

1 Introduction

We propose to verify the efficiency of an implicit scheme, based on the finite volume method, applied to Navier-Stokes equations. Two approaches for building such an algorithm are exposed with the perspective to compare them and find out the most efficient. Therefore, in the last part, different test cases are proposed where main results and the computational time are given as criteria of comparison.

2 Governing equations

The evolution equations we deal with are described by one or several sets of conservation with the expression given in (1).

$$\frac{\partial \vec{W}}{\partial t} + \vec{\nabla}.(\vec{f_c}(\vec{W}) + \vec{f_d}(\vec{W})) = \vec{s}(\vec{W}) \tag{1}$$

The vector of unknowns is denoted by $\vec{W}(x,t)$, whose size m corresponds to the number of conservative variables. It is defined on the domain $\Omega \subset \mathcal{R}^n \times \mathcal{R}^+$, with $n, m \in \mathcal{N}$.

The convective flux f_c is a density of macroscopic flux and the diffusive flux f_d is a density of microscopic flux, both are specified in the descritpion of Navier-Stokes equations.
The related functions $\vec{f}$ and $\vec{s}$ are defined as $\vec{f} : \mathcal{R}^m \to \mathcal{R}^{m,n}$ and as $\vec{s} : \mathcal{R}^m \to \mathcal{R}^m$

3 Numerical method

In this section, we briefly present the numerical algorithm used in the code and expose the construction of the matrix built with the implicit mehtod. We

distinguish the construction of the matrix with large stencil and the one with reduced stencil which roughly corresponds to first and second order.

3.1 FVM approximation

The finite volume method is used for the discretization of the equations (1): The domain Ω is divided into distinct elements T_i satisfying, $\Omega \equiv \bigcup_i T_i$ on which we integrate the conservative equations (1). With the Green's formula, we obtain :

$$\int_{T_i} \frac{\partial \vec{W}_i}{\partial t} dV + \oint_{\partial T_i} (\vec{f_c} + \vec{f_d}) \vec{n} dS = \int_{T_i} s(\vec{W}_i) dV. \tag{2}$$

The approximation of equation (2), in sense of finite volume method, leads on each T_i to

$$\begin{aligned} \frac{\partial \vec{W}_i}{\partial t} &= -\frac{1}{\mu_i} \sum_{j=1}^{N_j} (f_{c_{ij}} \vec{+} f_{d_{ij}}).\vec{n_{ij}} |\sigma_{ij}| + s_i(\vec{W}_i) = \\ &= -R_i(\vec{W}.\vec{\ell}) \end{aligned} \tag{3}$$

$\vec{\ell} = \{\ell_1, \ell_2, \cdots, \ell_s\}$ is a set containing the unknowns indices used for the evaluation of the residual.

Next, N_j is the number of faces of the sub-domain T_i, μ_i is volume of the sub-domain T_i, $\vec{f_{c_{ij}}}$, $\vec{f_{d_{ij}}}$ are the numerical fluxes, $s_i(\vec{W}_i)$ is a numerical approximation of the source term, $\vec{n}_{ij}$ is the unit normal vector of the face σ_{ij} (face between volumes T_i and T_j) and $|\sigma_{ij}|$ is the face's mesure.

Both numerical convective and diffusive flux require specific scheme, not given here.

3.1.1 Implicit scheme

We apply a backward differentiation formula (BDF), method of second order in time, to the time derivative term of equation (3).

$$\frac{3W_i^{k+1} - 4W_i^k + W_i^{k-1}}{2\Delta t} = -R_i(W_\ell^{k+1}). \tag{4}$$

We introduce the notation, $\Delta W^k = W^{k+1} - W^k$. After a linearization of the residual, we obtain the following linear system:

$$\frac{3}{2}\Delta W_i^k + \Delta t \sum_\ell \frac{\partial R_i^k}{\partial W_\ell} \Delta W_\ell^k = -\Delta t.R_i(W_\ell^k) + \frac{1}{2}\Delta W_i^{k-1}. \tag{5}$$

The time step Δt is chosen as the time step computed from a stability criterion for explicit scheme multiplied by a number called CFL.

In the right hand side of equation (5), the convective flux is computed with a second order (linear reconstruction, with Bart-Jespersen limiter) upwind scheme. The dissipative term comprises the computation of a gradient on the faces $\{i, j\}$ which is evaluated on a diamond cell (also second order).

In the left hand side, the derivative of the residual, $\partial R_i^k / \partial W_\ell$ consists of the derivative of convective and diffusive part of the residual. In the following sections, we present the computations of these derivatives with large or reduced stencil. In every case, we use the notation i for the current cell and j for a face neighbor.

3.1.2 Large stencil for convection

The scheme for convection computes the flux, on a face i, j, with two variables $\{W_i^r, W_j^r\}$ (r stands for reconstructed value). The derivative of that flux, on a face i, j, is written in equation (6) only in respect to W_i^r.

$$\frac{\partial \boldsymbol{F_C(W_i^r, W_j^r)}}{\partial \boldsymbol{W_i^r}} \boldsymbol{\Delta W_i^r} \tag{6}$$

With F_c, the numerical flux function. Note that ℓ is equal to j which means that the sum in equation (5) is done on all face neighbors.

The reconstruction for W_i^r method is a function whose entries are a certain amount of neighbors. We have:

$$W_i^r(W_{i_0}, ..., W_{i_k}, ..., W_{i_n}).$$

k is the index of any neighbour, n is their total amount.
The expression of the increment $\boldsymbol{\Delta W_i^r}$ is as follow:

$$\boldsymbol{\Delta W_i^r} = \sum_k^n \frac{\partial \boldsymbol{W_i^r}}{\partial \boldsymbol{W_{i_k}}} \boldsymbol{\Delta W_{i_k}}$$

By inserting this relation in equation (6), the Jacobian flux times the increment is given by the following equation:

$$\frac{\partial \boldsymbol{F_C(W_i^r, W_j^r)}}{\partial \boldsymbol{W_i^r}} \boldsymbol{\Delta W_i^r} = \frac{\partial \boldsymbol{F_C}}{\partial \boldsymbol{W_i^r}} \cdot \sum_k^n \frac{\partial \boldsymbol{W_i^r}}{\partial \boldsymbol{W_{i_k}}} \boldsymbol{\Delta W_{i_k}} \tag{7}$$

It means that for one face i, j, the derivative in respect to W_i^r gives an amount $m^2 \times i_n$ values which are added in the matrix of the implicit linear system.

3.1.3 Reduced stencil for convection

We approximate the previous equation (7) as follow:

$$\sum_{k=1}^n \frac{\partial \boldsymbol{W_i^r}}{\partial \boldsymbol{W_{i_k}}} = 0 \quad if i_k \neq i$$

So that the derivation of convective flux becomes:

$$\frac{\partial \boldsymbol{F_C}(\boldsymbol{W_i^r}, \boldsymbol{W_j^r})}{\partial \boldsymbol{W_i^r}} \Delta \boldsymbol{W_i^r} = \frac{\partial \boldsymbol{F_C}}{\partial \boldsymbol{W_i^r}} . \frac{\partial \boldsymbol{W_i^r}}{\partial \boldsymbol{W_i}} \Delta \boldsymbol{W_i} \tag{8}$$

In this case, for one face i, j, the derivative in respect to W_i^r gives an amount m^2 values which are added in the matrix of the implicit linear system.

3.1.4 Large stencil for diffusion

The stencil of the diffusive flux is determined by the algorithm used for the derivative of the diffusive variable (Velocity, Temperature ...). The diffusive coefficient, if not constant, has a simple stencil of computation, contained in the previous one, therefore, out of interest in this case.
The figure (1) is a diamond cell, built between two cells i, j, the interface of which is described with the nodes C_l.

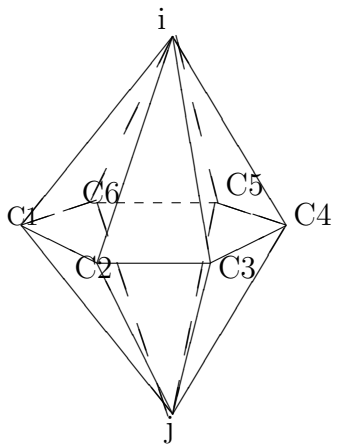


Figure 1:

The derivative is performed thanks to that cells (1). The values at the node C_l are computed with least square method. Finally, any derivative q on that cell has the following expression:

$$(\nabla q)_{ij} = \sum_k \boldsymbol{a_k} q_k + \boldsymbol{a_i} . q_i + \boldsymbol{a_j} . q_j$$

The values q_k are the centered value of cell k, $\boldsymbol{a_k}$ are geometrical coefficients stemming from both diamond scheme and LSM.

We also can write:

$$(\nabla q)_{ij} = \sum_k^{N_n} \boldsymbol{\alpha_\ell} q_\ell$$

N_n is the number of all the unknowns gathered from both method LSM and Diamond scheme.

The derivative of the diffusive flux $D_{ij}(\nabla q)_{ij}$ (D_{ij} the diffusive coefficient, here taken as a constant),on the face i, j, in respect to W_ℓ is as follow:

$$\frac{\partial \boldsymbol{F_D(W_1, \ldots, W_{N_n})}}{\partial \boldsymbol{W_\ell}} \Delta \boldsymbol{W_\ell} = D_{ij} \sum_k^{N_n} \boldsymbol{\alpha_\ell} \frac{f(W_\ell)}{W_\ell} \Delta \boldsymbol{W_\ell} \tag{9}$$

With $f(W_\ell) = q_\ell$, the diffusive variable.

In this case, $m^2 \times N_n$ values are added to the matrix.

3.1.5 Reduce stencil for diffusion

In the computation of $\nabla q)_{ij}$, the influence of the cells indices save i, j is neglected, so that the derivative of q is reduced to $\nabla q)_{ij} = \boldsymbol{a_i}.q_i + \boldsymbol{a_j}.q_j$ and the derivative of F_D in respect to W_i:

$$\frac{\partial \boldsymbol{F_D(W_i, W_j)}}{\partial \boldsymbol{W_i}} \Delta \boldsymbol{W_i} = D_{ij} \boldsymbol{\alpha_i} \frac{f(W_i)}{W_i} \Delta \boldsymbol{W_i} \tag{10}$$

In this case $2 \times m^2$ elements are added to the matrix.

3.1.6 Boundaries in the implicit scheme

In the expression given in the previous sections, some of the increment ΔW_ℓ, appear to be a ghost cell, which can't be considered as an unknown.
Yet, we know that any ghost value $\boldsymbol{W_\ell}$ is a function of the adjacent inner cell $\boldsymbol{W_{\ell_{ad}}}$, $\boldsymbol{W_\ell} = f(\boldsymbol{W_{\ell_{ad}}})$
Therefore, any ghost increment is modified as follow:

$$\Delta \boldsymbol{W_\ell} = \frac{\partial \boldsymbol{W_\ell}}{\partial \boldsymbol{W_{\ell_{ad}}}} \Delta \boldsymbol{W_{\ell_{ad}}}$$

4 Navier-Stokes with turbulence

We apply the implicit method to the Navier-Stokes equations:

$$\begin{cases} \rho_t + \nabla.(\rho.\boldsymbol{u}) = 0 \\ (\rho.\boldsymbol{u})_t + \nabla.(\rho.\boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p - \nabla.\bar{\bar{\tau}} = 0 \\ (\rho.e)_t + \nabla.(\rho.h.\boldsymbol{u}) - \nabla.(\bar{\bar{\tau}}.\boldsymbol{u} - \boldsymbol{q}) = 0 \end{cases}$$

With $\boldsymbol{u}$ the velocity, ρ the density, e the total energy equal to $u_i + \frac{\|\boldsymbol{u}\|^2}{2}$.
u_i the internal energy.
h the total enthalpy equal to $e + \frac{p}{\rho}$.
$\boldsymbol{q}$ the heat flux modeled by Fourier's law:$\boldsymbol{q} = -k\nabla T$ (k the thermal conductivity).

The medium is a Newtonian fluid, we have: $\bar{\bar{\tau}} = 2\mu\bar{\bar{D}} + \lambda\nabla.\boldsymbol{u}.\mathbb{1}$.
μ the dynamic viscosity, $\lambda = -\frac{2}{3}\mu$, the second dynamic viscosity.
The strain rate tensor is $\bar{\bar{D}} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^t)$

Two additional conservative equations are added to catch turbulence phenomena. A SST model is used in this case.

$$\begin{cases} (\rho k)_t + \nabla.(\rho.\boldsymbol{u}k) = \nabla.((\mu_L + \sigma_k.\mu_T)\nabla k) + \tau^F.D - \beta^*\rho\omega k \\ (\rho\omega)_t + \nabla.(\rho.\boldsymbol{u}\omega) = \nabla.((\mu_L + \sigma_\omega.\mu_T)\nabla\omega) + \frac{C_\omega\rho}{\mu_T}\tau^F.D - \beta\rho\omega^2 + 2(1-f_1)\frac{\rho\sigma_{\omega_2}}{\omega}\nabla\omega\nabla k \end{cases}$$

With k the turbulent energy, $\omega = \frac{\epsilon}{\beta^* k}$ the specific dissipation, f_1 damping function,μ_T turbulence viscosity and $\beta, beta^*, C_\omega, \sigma_k, \sigma_\omega$ constants of SST model.

4.1 Computational algorithm

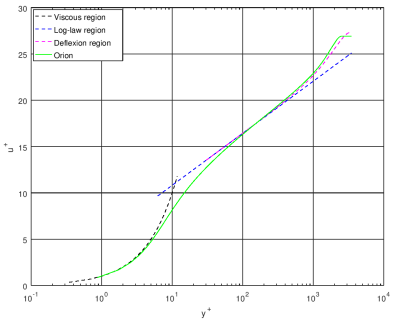
We use the following computational algorithm:

1. Firstly, we compute the compressible variables $\vec{W}^{k+1}$ from Navier-Stokes equations (1) which yields to a new time level (t^{k+1}).
2. Secondly, we compute the turbulent variables considering the compressible variable as constant of the time step (t^{k+1}).
3. Finally, we calculate the turbulent viscosity from turbulent equation used in Navier-Stokes equations.

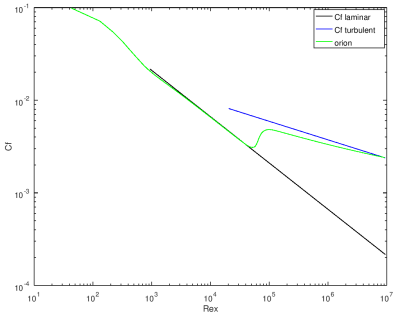
5 Results

5.1 2D flat plate

A free laminar stream flows over a flat plate in the close region of which the velocity, reduced to zero at the surface, develops into a viscous boundary layer for small Reynolds number and turn into a turbulence boundary layer further on the plate. Figure (5.1) shows the evolution of the skin friction along the plate, following the development of the boundary layer. The velocity profile in first graph of this figure is drawn when the turbulent boundary layer is sufficently developed.



Velocity profile at the abscissa $x = 0.8$



Skin friction

With $u^+ = \frac{u}{u_\tau}$, normalized velocity, $y^+ = \frac{yu_\tau}{\nu}$ normalized ordinate (u horizontal velocity, $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$, τ_w the viscous stress at the wall).
$Re_x = \frac{\rho u x}{\mu}$ is the Reynolds number along the plate and $Cf = \frac{\tau_w}{\frac{1}{2}\rho u_\infty^2}$ skin friction.

The green line represents the results given by the numerical computation, others lines are theoretial asymptotes (see in [2]) toward which the solution converges according to the physical configuration.

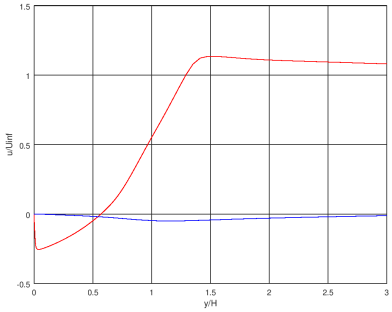
We give in the table (5.1), the CFL number (see in section 3.1.1), the time step corresponding to physical time and the computational time.

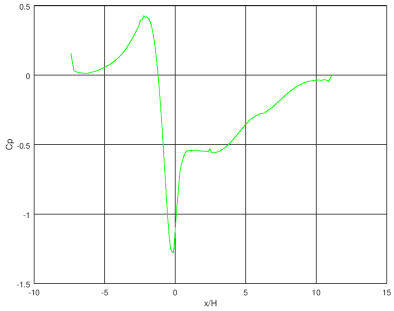
	CFL	Time step	Time CPU
Reduced stencil	35000	0.93 s	40 min
Large stencil	35000	0.93 s	215 min

Grid size : 32000 elements.

5.2 2D hill

In this case, a horizontal free stream flows over a hill. After the obstacle, a separation in the flow appears, which we can observe on the figure (5.2) with negative values of the horizontal velocity and a plateau in the pressure coefficient profile.





Velocity profile at the abscissa $\frac{x}{H} = 0.35$ Pressure coefficient

With $H = 0.04m$, eight of the hill and the pressure coefficient $C_p = \frac{p-p_\infty}{\frac{1}{2}\rho u_\infty^2}$.

	CFL	Time step	Time CPU
Reduce stencil	50000	1.9e-2 s	250 min
Large stencil	50000	1.9e-2 s	2240 min

Grid size : 35000 elements.

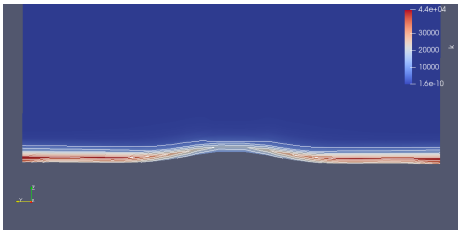
Both profiles are in accordance to the experimantal data, detailed in the paper [3]

5.3 3D bump

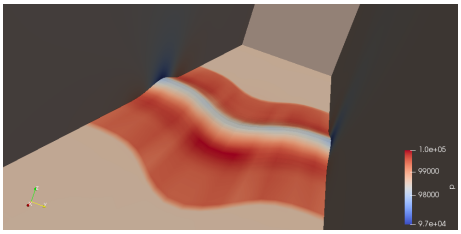
As in the previous case, a horizontal free stream flows over a 3D bump.

The computation is performed with the following parameters:
$R = 287.15$, $\mu = 2.69 \times 10^{-5}$, $Pr = 0.72$.

$p_0 = 101325$, $\rho_0 = 1.177$, $Re = 3 \times 10^6$, $U_\infty = 69$
$k_\infty = 0.675\frac{u_\infty^2}{Re}$, $\omega_\infty = 75\frac{u_\infty}{L}$



turbulent energy on plane $x = 1.2$



Pressure on the surface

	CFL	Time step	Time CPU
Reduce stencil	10000	2.68e-2 s	310 min
Reduce stencil	10000	2.68e-2 s	a lot!

This results are computed on a rough grid ($40 \times 20 \times 20$), hence the approximative results. We display nonetheless the shape of turbulent energy and pressure which corresponds to the results given on the following web page:

Turbulence Modeling Resource

6 Conclusion

The accuracy of results is the same for both methods, only in the stationnary phase. Transitionary phenomena are not treated in this article, such studies yet can be found in the article [1].

The serie of test cases has shown that both type of implicit schemes, with reduce and large stencil, allow to increase the time step 100-50000 times higher, according to the test case, than an explicit scheme. The CPU times of an explicit schme are not relayed in that article; some preliminary computation with an ideal flow has yet shown that the ratio between CPU for explicit and implicit has the ordre of the CFL number.

The CPU times differ importantly between both schemes. For the large stencil, the CPU times is in averaged 8 times bigger than in case of reduce stencil which means that this last scheme is the most efficient.

References

[1] Fort, J. · Karel, J. · Trdlicka, D. · Benkhaldoun, F. · Kissami, I. · Montavon, J.-B. · Hassouni, K. · Mezei, J. Zs. Finite volume methods for numerical simulation of the discharge motion described by different physical models ACOM 2018

[2] Schlichting, H., Gersten, K. (2016). Boundary-layer theory. Springer.

[3] T. Houra · Y. Nagano Turbulent Heat and Fluid Flow over a Two-Dimensional Hill Springer Science + Business Media B.V. 2009