A matrix-free implicit method for flows at all speeds

Alberto $\text{BECCANTINI}^{(a)}$, Christophe $\text{CORRE}^{(b)}$ and Thibaud $\text{KLOCZKO}^{(a,b)}$

(a) CEA Saclay / DM2S / LTMF91 191, Gif-sur-Yvette Cedex - FRANCE {beccantini}{kloczko}@cea.fr

(b) SINUMEF Lab., ENSAM 151, boulevard de l'Hôpital - 75013 Paris - FRANCE corre@paris.ensam.fr

The computation of steady flows can be considered efficient not only if a steady state is reached for a reduced CPU time but also if a low memory storage is used in this process; besides for complex industrial applications involving grids with very large numbers of points this latter requirement may become especially critical. The same comment holds of course for unsteady flows, now classically computed by a dual time-stepping approach in which a steady-state with respect to the dual time must be approximately reached at each physical timestep. The strong need for low-storage efficient implicit treatments has led to the development of so-called matrix-free methods [1][2]. In the particular applications we are dealing with here, namely the simulation of buoyant multicomponent reactive flow in a nuclear reactor containment, the methods should be versatile enough to deal with flow regimes ranging from nearly incompressible to highly compressible [3]. It has been known for more than a decade now that a proper preconditioning, so-called low-Mach preconditioning, of the equations governing compressible flows enables the application of schemes initially developed for compressible flows simulation to the incompressible regime (see [4] for a recent review). The introduction of such a preconditioning matrix in an implicit scheme is not really an issue if a standard block factorization or relaxation is used to solve the linear system associated with the implicit stage [5][6]. However, one may wonder whether it is possible to preserve a *matrix-free* implicit treatment for flows at all speeds when the implicit stage involves a *preconditioning matrix*; it is the purpose of the present paper to bring an answer to this question.

A time-accurate solution of the Navier-Stokes equations is computed for flows at all speeds by solving :

$$P\frac{\partial w}{\partial \tau} + \frac{\partial w}{\partial t} + \sum_{p=1}^{d} \frac{\partial f_p^E}{\partial x_p} = \sum_{p=1}^{d} \frac{\partial f_p^V}{\partial x_p} \tag{1}$$

where τ is a pseudo or dual-time, t is the physical time, f_p^E and f_p^V are respectively the Euler (convective) and viscous fluxes in the p^{th} space-direction (d is the dimension of the problem) and P is a preconditioning matrix which takes, in the present work, the form proposed in [7]. System (1) is solved by the following scheme :

$$P\frac{w_j^{n,m+1} - w_j^{n,m}}{\Delta \tau} + \frac{\frac{3}{2}(w_j^{n,m} - w_j^n) - \frac{1}{2}(w_j^n - w_j^{n-1})}{\Delta t} + \sum_{p=1}^d (\frac{\delta_p h^p}{\delta x_p})_j^{n,m} = 0$$
(2)

where *m* is the pseudo-iteration (on dual-time) counter, *n* is the time step counter, $j = (j_1, j_2, \ldots, j_d)$ is a multi-integer associated with a point $x_j = (j_1 \delta x_1, \ldots, j_d \delta x_d)$ of a regular Cartesian grid; $(\delta_p h^p)$ is the difference operator on a grid cell δ_p in the p^{th} space-direction applied to the numerical flux h^p approximating the physical flux $(f_p^E - f_p^V) : (\delta_p h^p)_j = h_{j+\frac{e_p}{2}}^p - h_{j-\frac{e_p}{2}}^p$, where e_p is a multi-integer with components e_{pq} equal to 0 if $q \neq p$ and to 1 if q = p. For high-resolution computation, the numerical flux depends on a left and right state, $h_{j+\frac{e_p}{2}}^p = h(w_{j+\frac{e_p}{2}}^L, w_{j+\frac{e_p}{2}}^R)$, which are reconstructed from adjacent grid point values using a now standard MUSCL approach. It is easy to check that, at steady-state on τ , scheme (2) yields a second-order time-accurate solution of the preconditioned Navier-Stokes equations. In order to speed up the convergence to the pseudo-steady state, scheme (2) is made implicit with respect to τ ; if a simple first-order space-discretization is retained to build this implicit stage, it eventually takes the form :

$$D\Delta w_{j}^{n,m} + \sum_{p=1}^{d} [\sigma_{p}(\mu_{p}A_{p}\delta_{p}\Delta w^{n,m})_{j} - (\delta_{p}(\frac{\tilde{\rho}_{p}}{2}P + \dot{\rho}_{p}^{V})\delta_{p}\Delta w^{n,m})_{j}] = \Delta w_{j}^{exp} = -\sum_{p=1}^{d} \sigma_{p}(\delta_{p}h^{p})_{j}^{n,m} - \lambda[\frac{3}{2}(w^{n,m} - w^{n}) + \frac{1}{2}(w^{n} - w^{n-1})]$$
(3)

where $\sigma_p = \Delta \tau / \delta x_p$, $\lambda = \Delta \tau / \Delta t$, $\Delta w^{n,m} = w^{n,m+1} - w^{n,m}$, μ_p is the average operator over a grid cell in the p^{th} space-direction $((\mu_p v)_j = \frac{1}{2}(v_{j-\frac{e_p}{2}} + v_{j+\frac{e_p}{2}}))$, $\dot{\rho}_p$ is the spectral radius of the preconditioned Jacobian $P^{-1}(\sigma_p \tilde{A}_p)$ (with $A_p = df_p^E / dw$) and $\dot{\rho}_p^V$ is the spectral radius of the viscous Jacobian $\frac{\sigma_p}{\delta x_p} A_p^V$. The diagonal coefficient appearing in the LHS implicit stage is defined by :

$$D = P_j + \frac{3}{2}\lambda Id + \sum_{p=1}^{d} [(\frac{\dot{\tilde{\rho}}_p}{2}P + \dot{\rho_p^V})_{j - \frac{e_p}{2}}Id + (\frac{\dot{\tilde{\rho}}_p}{2}P + \dot{\rho_p^V})_{j + \frac{e_p}{2}}Id]$$
(4)

In order to produce a matrix-free method, products $A_p \delta_p \Delta w^{n,m}$ appearing on the LHS of (3) are replaced by $\delta_p \Delta (f_p^E)^{n,m}$ and all the non-diagonal terms are relaxed to yield :

$$D\Delta w_{j}^{(l+1)} = \underbrace{[\Delta w_{j}^{exp} - \sum_{p=1}^{d} \sigma_{p} (\delta_{p} \mu_{p} \Delta (f_{p}^{E})^{(l)})_{j} + \sum_{p=1}^{d} (\delta_{p} \dot{\rho^{V}}_{p}) \delta_{p} \Delta w^{(l)})_{j}]}_{=(\Delta w_{1}^{(l)})_{j}} + \underbrace{\sum_{p=1}^{d} (\delta_{p} (\dot{\rho^{P}}_{p} P) \delta_{p} \Delta w^{(l)})_{j}}_{\approx P_{j} (\Delta w_{2}^{(l)})_{j}}$$
(5)

For standard compressible flows, in which preconditioning is not required, P = Id so that D is a diagonal matrix and the implicit scheme (5) is indeed matrix-free since D^{-1} acts as a scalar on each component of the RHS. However, when the preconditioning [7] is applied, P becomes of the form :

$$P = Id + (\frac{1}{\Phi^2} - 1)\frac{\gamma - 1}{c^2} \begin{pmatrix} q^2 = \frac{1}{2}(u^2 + v^2) & -u & -v & 1\\ uq^2 & -u^2 & -uv & u\\ vq^2 & -uv & -v^2 & v\\ Hq^2 & -uH - vH & H \end{pmatrix} = Id + (\frac{1}{\Phi^2} - 1)Q$$
(6)

where Φ is a function of the local Mach number, c is the speed of sound, u and v the velocity components and H the total enthalpy. Consequently the diagonal coefficient D given by (4) is now a full matrix, to be inverted, so that storage requirement and operations count of the implicit treatment increase for low Mach number flows. This loss of efficiency can be cured if, in a first step, D is computed with a single evaluation of matrix P at point j, resulting in a diagonal coefficient of the form :

$$D = \underbrace{(1 + \sum_{p=1}^{d} [\frac{1}{2} (\dot{\tilde{\rho}}_{p})_{j - \frac{e_{p}}{2}} + \frac{1}{2} (\dot{\tilde{\rho}}_{p})_{j + \frac{e_{p}}{2}}])}_{=a} P_{j} + \underbrace{(\frac{3}{2}\lambda + \sum_{p=1}^{d} [(\dot{\rho}_{p}^{V})_{j - \frac{e_{p}}{2}} + (\dot{\rho}_{p}^{V})_{j + \frac{e_{p}}{2}}])}_{=b} Id$$
(7)

Taking advantage of the fact that the matrix Q in (6) satisfies $Q^2 = Q$, it is easy to obtain an explicit expression for D^{-1} ; left-multiplying (5) by this expression and rearranging leads to :

$$\Delta w_j^{(l+1)} = \frac{1}{a+b} [\Delta w_1^{(l)} + \Delta w_2^{(l)}]_j + \frac{\Phi_j^2 - 1}{(a+b)(a+b\Phi_j^2)} Q_j [a\Delta w_1^{(l)} + b\Delta w_2^{(l)}]_j$$
(8)

Treatment (8) becomes wholly matrix-free if the specific expression of matrix Q given in (6) is used to express a matrix-vector such as $Q \cdot X$ under the form :

$$Q.X = \frac{\gamma - 1}{c^2} \left(\frac{q^2}{2} X^{(1)} - u X^{(2)} - v X^{(3)} + X^{(4)}\right) \cdot \left(1 \ u \ v \ H\right)^T$$

with $X^{(m)}$ the components of vector X. Note that the implicit scheme (8) is not only matrix-free now but also allows to decouple the standard matrix-free implicit treatment used for compressible flow from the added treatment specific to low-Mach number flows ($\Phi \neq 1$). Applications of the scheme (with explicit treatment of Roe, Rusanov or AUSM+ type) will be presented at the Conference for a range of low-Mach number flows, on structured and unstructured grids; assessment of the efficiency gains offered by the approach with respect to more conventional block treatments will be detailed.

References

[1] H. Luo, J.D. Baum, R. Löhner, An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids, Computers and Fuids, **30** (2001), 137-159.

[2] Y. Zhao, Computation of complex turbulent flow using matrix-free implicit dual time-stepping scheme and LRN turbulence model on unstructured grids, Computers and Fluids, **33** (2004), 119-136.

[3] A. Beccantini, F. Dabbene, S. Kudriakov et al., Simulation of hydrogen release and combustion in large scale geometries: Models and Methods, Int. Conf. on Supercomputing in Nuclear Application, Paris, France, Sept. 22-24, 2003.
[4] E. Turkel, Preconditioning techniques in CFD, Annu. Rev. Mech. 31 (1999), 385-416.

[5] W.R. Briley, L.K. Taylor, D.L. Whitfield, High-resolution viscous flow simulations at arbitrary Mach number, J. Comput. Phys. 184(2003), 79-105.

[6] S.A. Pandya, S. Venkateswaran, T.H. Pulliam, Implementation of preconditioned dual-time procedures in OVER-FLOW, AIAA Paper 2003-0072, Jan. 2003.

[7] J.M. Weiss, W.A. Smith, Preconditioning applied to variable and constant density flows, AIAA J., 33, n° 11 (1995).