A Schur Complement Method for Compressible Navier-Stokes Equations

Thu-Huyen Dao[1,2], Michael Ndjinga[1], and Frédéric Magoulès[2]

[1] CEA–Saclay, DEN, DM2S, STMF, LMEC, F–91191 Gif–sur–Yvette, France
[2] Appl. Mat. and Syst. Lab., Ecole Centrale Paris, 92295 Châtenay–Malabry, France
thu-huyen.dao@cea.fr, michael.ndjinga@cea.fr,
frederic.magoules@hotmail.com

Summary. Domain decomposition methods were first developed for elliptic problems, taking advantage of the strong regularity of their solutions. In the last two decades, many investigations have been devoted to improve the performance of these methods for elliptic and parabolic problems. The situation is less clear for hyperbolic problems with possible singular solutions. In this paper, we will discuss a nonoverlapping domain decomposition method for nonlinear hyperbolic problems. We use the finite volume method and an implicit version of the Roe approximate Riemann solver, and propose a new interface variable inspired by Dolean and Lanteri [1]. The new variable makes the Schur complement approach simpler and allows the treatment of diffusion terms. Numerical results for the compressible Navier-Stokes equations in various 2D and 3D configurations such as the Sod shock tube problem or the lid driven cavity problem show that our method is robust and efficient. Comparisons of performances on parallel computers with up to 512 processors are also reported.

1 Introduction

When solving a nonlinear partial differential equation by an implicit scheme, one classically ends by solving a nonlinear algebraic system using a Newton method. At each step of this method we have to solve a linear system $\mathscr{A}(U^k)U^{k+1} = b(U^k)$. This task is computationally expensive in particular since the matrix $\mathscr{A}$ is usually non-symmetric and very ill-conditioned. It is therefore necessary to find an efficient preconditioner.

When the size of the system is large (as in the case of $3D$ computations), the parallel solution on multiple processors is essential to obtain reasonable computation times. Currently in the thermal hydraulic code, FLICA-OVAP (see [2]), the matrix $\mathscr{A}$ and the right hand side b are stored on multiple processors and the system is solved in parallel with a Krylov solver (classical incomplete factorization). Unfortunately, the parallel preconditioners of FLICA-OVAP only perform well on a few processors. In contrast, if we want to increase the number of processors these parallel preconditioners perform poorly. Tests were run on different test cases and led

R. Bank et al. (eds.), *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering 91,
DOI 10.1007/978-3-642-35275-1_64,

us to conclude that it is often better not to use these parallel preconditioners, especially for $3D$ problems. This strategy does not make an optimal use of the available computational power. Hence we seek for more efficient methods to distribute the computations. We study and use a domain decomposition method as an alternative to the classical distribution.

The paper is organized as follows. In Sects. 2 and 3, we present the mathematical model and its numerical schemes. In Sect. 4, we first review the domain decomposition method proposed by Dolean and Lanteri [1] based on a Schwarz algorithm. We then introduce a new interface variable which makes the Schur complement approach simpler and allows for the treatment of diffusion terms. Section 5 presents a set of numerical experiments to validate our method, compares it with that of [1] concerning the robustness and efficiency and presents the scalability and the performance of different preconditioners.

2 Mathematical Model

The simplest model of FLICA-OVAP consists of the following three balance laws for the mass, the momentum and the energy:

$$\begin{cases} \frac{\partial \rho}{\partial t} & + \nabla \cdot \mathbf{q} & & = 0 \\ \frac{\partial \mathbf{q}}{\partial t} & + \nabla \cdot \left(\mathbf{q} \otimes \frac{\mathbf{q}}{\rho} + p \mathbb{I}_d \right) & - \nu \Delta (\frac{\mathbf{q}}{\rho}) & = 0 \\ \frac{\partial (\rho E)}{\partial t} & + \nabla \cdot \left[(\rho E + p) \frac{\mathbf{q}}{\rho} \right] & - \lambda \Delta T & = 0 \end{cases} \tag{1}$$

where ρ is the density, $\mathbf{v}$ the velocity, $\mathbf{q} = \rho \mathbf{v}$ the momentum, p the pressure, ρe the internal energy, $\rho E = \rho e + \frac{\|\mathbf{q}\|^2}{2\rho}$ the total energy, T the absolute temperature, ν the viscosity and λ the thermal conductivity. We close the system (1) by the ideal gas law $p = (\gamma - 1)\rho e$. For the sake of simplicity, we consider constant viscosity and conductivity, and neglect the contribution of viscous forces in the energy equation. By denoting $U = (\rho, \mathbf{q}, \rho E)^t$ the vector of conserved variables, the Navier–Stokes system (1) can be written as a nonlinear system of conservation laws:

$$\frac{\partial U}{\partial t} + \nabla \cdot (\mathscr{F}^{conv}(U)) + \nabla \cdot \left(\mathscr{F}^{diff}(U) \right) = 0, \tag{2}$$

where $\mathscr{F}^{conv}(U) = \begin{pmatrix} \mathbf{q} \\ \mathbf{q} \otimes \frac{\mathbf{q}}{\rho} + p \mathbb{I}_d \\ (\rho E + p) \frac{\mathbf{q}}{\rho} \end{pmatrix}$, $\mathscr{F}^{diff}(U) = \begin{pmatrix} 0 \\ -\nu \nabla (\frac{\mathbf{q}}{\rho}) \\ -\lambda \nabla T \end{pmatrix}$.

3 Numerical Method

The conservation form (2) allows for the definition of weak solutions, which can be discontinuous ones. Discontinuous solutions such as shock waves are of great

importance in transient calculations. In order to correctly capture shock waves, one needs a robust, low diffusive conservative scheme. The finite volume framework is the most appropriate setup to write discrete equations that express the conservation laws at each cell (see [3]).

We decompose the computational domain into N disjoint cells C_i with volume v_i. Two neighboring cells C_i and C_j have a common boundary ∂C_{ij} with area s_{ij}. We denote $N(i)$ the set of neighbors of a given cell C_i and $\mathbf{n}_{ij}$ the exterior unit normal vector of ∂C_{ij}. Integrating the system (2) over C_i and setting $U_i(t) = \frac{1}{v_i}\int_{C_i} U(x,t)dx$ and $U_i^n = U_i(n\Delta t)$, the discretized equations can be written:

$$\frac{U_i^{n+1}-U_i^n}{\Delta t} + \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left(\overrightarrow{\Phi}_{ij}^{conv} + \overrightarrow{\Phi}_{ij}^{diff}\right) = 0. \tag{3}$$

with: $\overrightarrow{\Phi}_{ij}^{conv} = \frac{1}{s_{ij}}\int_{\partial C_{ij}} \mathscr{F}^{conv}(U^{n+1})\cdot\mathbf{n}_{ij}ds$, $\overrightarrow{\Phi}_{ij}^{diff} = \frac{1}{s_{ij}}\int_{\partial C_{ij}} \mathscr{F}^{diff}(U^{n+1})\cdot\mathbf{n}_{ij}ds$.

To approximate the convection numerical flux $\overrightarrow{\Phi}_{ij}^{conv}$ we solve an approximate Riemann problem at the interface ∂C_{ij}. Using the Roe local linearisation of the fluxes [4], we obtain the following formula:

$$\overrightarrow{\Phi}_{ij}^{conv} = \frac{\mathscr{F}^{conv}(U_i^{n+1})+\mathscr{F}^{conv}(U_j^{n+1})}{2}\cdot\mathbf{n}_{ij} - \mathscr{D}(U_i^{n+1},U_j^{n+1})\frac{U_j^{n+1}-U_i^{n+1}}{2} \tag{4}$$

$$= \mathscr{F}^{conv}(U_i^{n+1})\mathbf{n}_{ij} + A^-(U_i^{n+1},U_j^{n+1})(U_j^{n+1}-U_i^{n+1}), \tag{5}$$

where $\mathscr{D}$ is an upwinding matrix, $A(U_i^{n+1},U_j^{n+1})$ the Roe matrix and $A^{\pm} = \frac{A\pm\mathscr{D}}{2}$. The choice $\mathscr{D}=0$ gives the centered scheme, whereas $\mathscr{D}=|A|$ gives the upwind scheme. For the Euler equations, we can build $A(U_i^{n+1},U_j^{n+1})$ explicitly using the Roe averaged state (see [3]).

The diffusion numerical flux $\overrightarrow{\Phi}_{ij}^{diff}$ is approximated on structured meshes using the formula:

$$\overrightarrow{\Phi}_{ij}^{diff} = D\left(\frac{U_i^{n+1}+U_j^{n+1}}{2}\right)(U_j^{n+1}-U_i^{n+1}) \tag{6}$$

with the matrix $D(U) = \begin{pmatrix} 0 & \mathbf{0} & 0 \\ \frac{\nu\mathbf{q}}{\rho^2} & \frac{-\nu}{\rho}\mathbb{I}_d & 0 \\ \frac{\lambda}{c_v}\left(\frac{c_vT}{\rho}-\frac{||\mathbf{q}||^2}{2\rho^3}\right) & \frac{\mathbf{q}^t\lambda}{\rho^2 c_v} & -\frac{\lambda}{c_v\rho} \end{pmatrix}$, where c_v is the heat capacity at constant volume.

3.1 Newton Scheme

Finally, since $\sum_{j\in N(i)}\mathscr{F}^{conv}(U_i^{n+1}).\mathbf{n}_{ij} = 0$, using (5) and (6) the Eq. (3) of the numerical scheme becomes:

$$\frac{U_i^{n+1}-U_i^n}{\Delta t} + \sum_{j\in N(i)}\frac{s_{ij}}{v_i}\{(A^- + D)(U_i^{n+1},U_j^{n+1})\}(U_j^{n+1}-U_i^{n+1}) = 0. \tag{7}$$

The system (7) is nonlinear, hence we use the following Newton iterative method to obtain the required solutions:

$$\frac{\delta U_i^{k+1}}{\Delta t} + \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(U_i^k, U_j^k)\right]\left(\delta U_j^{k+1} - \delta U_i^{k+1}\right) = -\frac{U_i^k - U_i^n}{\Delta t} - \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(U_i^k, U_j^k)\right](U_j^k - U_i^k), \tag{8}$$

where $\delta U_i^{k+1} = U_i^{k+1} - U_i^k$ is the variation of the k-th iterate that approximates the solution at time $n+1$.

4 Domain Decomposition Method

The principle of the domain decomposition method by Schur complement is to decompose the global problem into independent subproblems solved on each processor. More precisely, if we want to solve the problem:

$$\begin{cases} \frac{\partial U}{\partial t} + \nabla \cdot \mathscr{F}(U) = 0 \text{ in } \Omega \\ BU = g \qquad \text{on } \partial\Omega \end{cases} \tag{9}$$

on a partition of the original domain $\Omega = \cup_{I=1}^{K} \Omega_I$, defining U_I as the restriction of the solution U in the subdomain Ω_I, the algorithm of the domain decomposition method is then written as:

$$\begin{cases} \dfrac{\partial U_I}{\partial t} + \nabla \cdot \mathscr{F}(U_I) = 0 \text{ in } \Omega \\ BU_I = g \qquad \text{on } \partial\Omega \cap \partial\Omega_I \\ C_I U_I = C_I U_J \qquad \text{on } \partial\Omega_I \cap \partial\Omega_j \end{cases} \tag{10}$$

where C_I is an interface operator which we will clarify later.

4.1 Dolean and Lanteri Interface Variable

In the article [1], in order to make the subsystem (10) solution independent, Dolean et al introduced a redundant variable Φ_{ij}^{DL} at the domain interface between two cells i and j : $\Phi_{ij}^{DL} = A^+_{Roe,\mathbf{n}_{i,j}} U_i - A^-_{Roe,\mathbf{n}_{i,j}} U_j$ and then defined the orthogonal projectors $P^{\pm}$ on the eigenvectors subspaces such that

$$P^-(U_i,U_j)\delta\phi_{ij}^{Do} = A^-_{Roe,\mathbf{n}_{i,j}}\delta U_j^{k+1},\ P^+(U_i,U_j)\delta\phi_{ij}^{Do} = -A^+_{Roe,\mathbf{n}_{i,j}}\delta U_i^{k+1}$$

This strategy can only be applied to the Euler equations (Eq. (2) with no viscosity and heat conductivity terms) using the upwind scheme. In order to include diffusion terms in the model and to use various schemes, we introduce a new interface variable Φ_{ij} at the domain interface between two cells i and j:

$$\Phi_{ij} = U_j - U_i \tag{11}$$

4.2 A New Interface Variable

In the case where the cell i of the subdomain I is at the boundary and has to communicate with the neighboring subdomains, we can rewrite the system (8) as:

$$\begin{aligned}\frac{\delta U_i^{k+1}}{\Delta t} \quad &+ \quad \sum_{j\in I, j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(U_i^k, U_j^k)\right]\left(\delta U_j^{k+1} - \delta U_i^{k+1}\right) \\ &= \quad -\frac{U_i^k - U_i^n}{\Delta t} - \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(U_i^k, U_j^k)\right](U_j^k - U_i^k) \\ &\quad - \sum_{j\notin I, j\in N(i)} \left[(A^- + D)(U_i^k, U_j^k)\right]\delta\phi_{ij}\end{aligned}$$

By defining $\mathscr{U}_I = (U_1, \ldots, U_m)^t$ the unknown vector of the subdomain I and

$$\delta\phi_{IJ} = (\delta\phi_{ij})_{i\in I, j\in J, j\in N(i)} \tag{12}$$

and by denoting $P = A^- + D$, we can write the linear system as:

$$\mathscr{A}(\mathscr{U}_I^k)\delta\mathscr{U}_I^{k+1} = b_I(\mathscr{U}^n, \mathscr{U}^k) - \sum_{J\in N(I)} P(\mathscr{U}_I^k, \mathscr{U}_J^k)\delta\phi_{IJ} \tag{13}$$

By taking into account Eqs. (11)–(13), we can build an extended system that distinguishes the internal unknowns from the interface ones:

$$\left(\begin{array}{cccc|c} \mathscr{A}_1 & 0 & \ldots & \ldots & P_1 \\ 0 & \mathscr{A}_2 & 0 & \ldots & P_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \mathscr{A}_N & P_N \\ \hline M_1 & \ldots & \ldots & M_N & \mathbb{I} \end{array}\right)\left(\begin{array}{c} \delta\mathscr{U}_1 \\ \delta\mathscr{U}_2 \\ \ldots \\ \delta\mathscr{U}_N \\ \delta\Phi \end{array}\right) = \left(\begin{array}{c} b_1 \\ b_2 \\ \ldots \\ b_N \\ b_\phi \end{array}\right) \tag{14}$$

where $\mathscr{A}_I$ is the matrix that couples the unknowns associated with internal cells of Ω_I whereas M_I enables us to build $\delta\Phi$, the interface unknown on all coupling subdomain interfaces, from the δU_I. The internal unknowns can be eliminated in favor of the interface ones to yield the following interface system:

$$S\delta\phi = b_\phi \tag{15}$$

with

$$\begin{aligned}(S\delta\phi)_{IJ} &= \delta\phi_{IJ} + M_{IJ}\mathscr{A}_I^{-1}\sum_{K\in N(I)} P_{IK}\delta\phi_{IK} + M_{JI}\mathscr{A}_J^{-1}\sum_{K\in N(J)} P_{JK}\delta\phi_{JK} \\ (b_\phi)_{IJ} &= M_{IJ}{A_I}^{-1}b_I + M_{JI}{A_J}^{-1}b_J\end{aligned}$$

The Eq. (15) can be solved by, e.g., GMRES, BICGStab, or the Richardson methods.

5 Numerical Results

5.1 Validation

Figures 1 and 2 present the profile of the pressure after 10 time steps using the upwind scheme with CFL = 10 for the Euler equations. Our initial state is a pressurized ball at the center of a closed box and for $t > 0$ there are waves which propagate and reflect all over the box. The gas expands in the box and we can see the shock waves and the rarefaction waves. The solution is solved on a cartesian mesh of 200×200 cells.

Figures 3 and 4 show the streamlines of the steady state obtained using centered scheme to solve a lid driven cavity flow at Reynolds number 400 on a cartesian 50×50 mesh. The lid speed is 1 m/s, the maximum Mach number of the flow is 0.008. According to these results, we obtain the same solutions by using single or multiple domains.

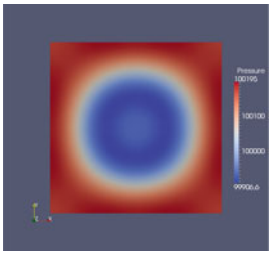

Fig. 1. Profile of the pressure at time step 10 on one processor

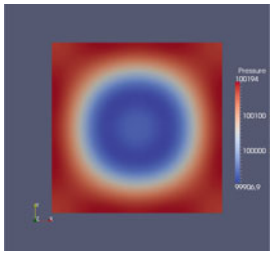

Fig. 2. Profile of the pressure at time step 10 on four processors

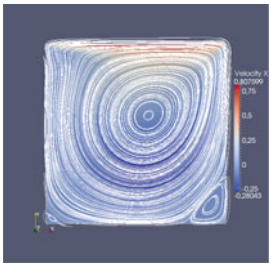

Fig. 3. Streamlines of V_x on one processor

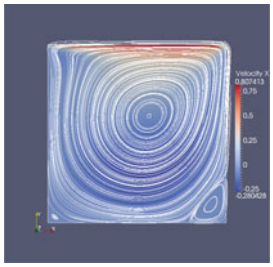

Fig. 4. Streamlines of V_x on four processors

5.2 Scalability

We now study the robustness and the scalability of our numerical method using the same test as presented in Sect. 5.1. In Figs. 5 and 6, we compare the parallel efficiency

of different preconditioners on 2*D* and 3*D* computations and with two and four processors. We see that without the preconditioner the solver is scalable. However, when

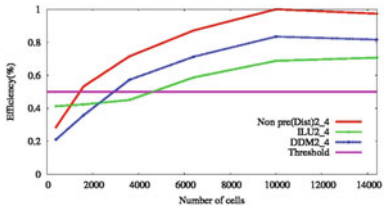

Fig. 5. Parallel efficiency for 2D Lid driven cavity

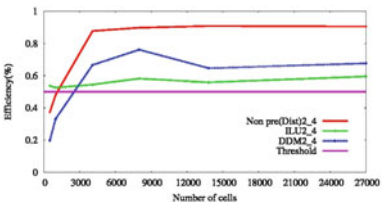

Fig. 6. Parallel efficiency for 3D Lid driven cavity

we use the Incomplete LU preconditioner, the scalability is not optimal especially for 3*D* problems. Our method proves better than ILU when we increase the number of cells in each subdomain. In Fig. 7, we compare the robustness of different methods

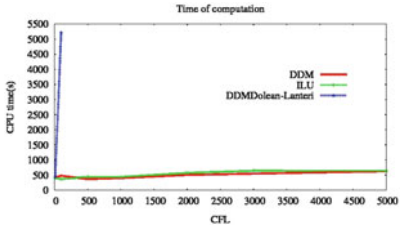

Fig. 7. Comparisons of parallelism in 3D Detonation, global mesh = $50 \times 50 \times 50$

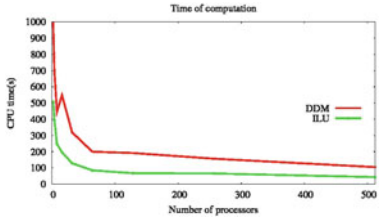

Fig. 8. Time of computation, 1 time step, global mesh = $96 \times 96 \times 96$

using the detonation problem. This problem is solved on a catersian $50 \times 50 \times 50$ cell mesh on two processors. The computation time of Dolean and Lanteri method increases rapidly because it needs many Newton iterations for convergence at each time step. In Fig. 8, we compare the scalability of the ILU preconditioner and of our method using the lid driven cavity problem solved on a global catersian $96 \times 96 \times 96$ cell mesh. The computation time of the domain decomposition method is higher than that of the ILU preconditioner due to the large number of Schur complement iterations.

6 Conclusion

We have presented a new interface variable which allows for the treatment of diffusion terms and the use of various numerical schemes. We also compared the efficiency and the scalability of our method with the classical distributed computations

and the method of Dolean and al. Our approach seems promising but we still need to find an efficient preconditioner for the Schur complement in order to reduce its computational time.

Bibliography

[1] V. Dolean and S. Lanteri. A domain decomposition approach to finite volume solution of the Euler equations on unstructured triangular meshes. *Int. J. Numer. Meth. Fluids*, 37(6), 2001.

[2] P. Fillion, A. Chanoine, S. Dellacherie, and A. Kumbaro. FLICA-OVAP: a new platform for core thermal-hydraulic studies. In *NURETH-13*, 2009.

[3] E. Godlewski and P.A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer Verlag, 1996.

[4] P.L Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, 43, 1981.