Overlapping Domain Decomposition Applied to the Navier-Stokes Equations

Oana Ciobanu, Laurence Halpern, Xavier Juvigny, Juliette Ryan

Overlapping Domain Decomposition Applied to the Navier–Stokes Equations

Oana Ciobanu, Laurence Halpern, Xavier Juvigny, and Juliette Ryan

1 Introduction

This article focuses on the research field of laminar flow of an ideal gas, on the resolution of aerodynamic multi-scale problems that are costly and difficult to solve in their original form. In order to solve these large data systems several techniques of parallel computing have been developed but some convergence problems may occur for large number of sub-domains.

Robust and fast methods are now available, which combine non-linear and linear solvers requiring less memory capacity. In the context of long term simulations, global implicit approaches have proven their superiority as they are able to simulate a quasi-steady-state behaviour without being restricted to short time steps to ensure convergence. Implementing these approaches on GPUs can certainly improve the efficiency versus a simple CPU implementation, as will be shown below, but by combining this implementation with domain decomposition another scale of efficiency could be achieved. In this paper, we propose an improved parallel time-space method for steady/unsteady problems modelled by Euler and Navier-Stokes equations for a direct numerical simulation.

Domain decomposition methods split large problems into smaller sub-problems that can be solved in parallel. Usually, only space domain decomposition method is used to provide high-performing algorithms in many fields of numerical applications. To achieve full performance on large clusters with up to 100,000 nodes (such

O. Ciobanu (✉) • X. Juvigny • J. Ryan
ONERA, BP72-29 avenue de la Divison Leclerc, 92322 Chatillon, France
e-mail: oana.ciobanu@onera.fr

L. Halpern
Université Paris 13, LAGA, 93430 Villetaneuse, France
e-mail: halpern@math.univ-paris13.fr

as recently the IBM Sequoia, or GPUs) the time dimension has to be taken into account. An essential gain to be obtained from time-space domain decomposition is the ability to apply different time-space discretisation on sub-domains thus improving efficiency and convergence of implicit schemes.

In practice, we are often working with large computational domains where only a small part is highly interactive and a wide region of the domain is close to equilibrium state. What is usually done is that the sub-domains are balanced in space so that each processor finishes the simulation at the same moment and the computation is done, on each sub-domain with the same time step, the global one. The time step depends on the CFL condition, the value of the flow velocity and the space step. This means that the part of the simulation domain which is not dominated by strong non-linearities is solved with a much higher precision than is needed. Some sub-domains are over-solved. The sub-domain close to the equilibrium state converges in fewer iterations and it is less costly, but it has to wait for the high reactive sub-domain to end in order to continue the simulation. To avoid this loss of efficiency and optimize the computational cost, the time step should be computed locally and the distribution of flow in sub-domains should take into consideration several factors: closeness to equilibrium region, strong non-linearities region and time step influence.

Our work focuses on the improvement of the Schwarz waveform relaxation (SWR) Method introduced under this name by Gander [4] at the 10th Domain Decomposition Conference to solve parabolic equations. It was previously presented by Gander and Stuart [5] as a multi-splitting formulation on overlapping sub-domains [9] combined with a waveform relaxation algorithm [12] in space-time for the heat equation. The purpose is to solve the space-time partial differential equation in each sub-domain in parallel, and to transmit domain boundary information to the neighbours at the end of the time interval. Originally applied to linear PDEs, the SWR algorithm was extended and optimised to the non-linear reactive transport equations by Haeberlein [6] and Haeberlein and Halpern [7]. With the SWR method different time-space discretisation can be applied on sub-domains thus improving efficiency and convergence of the schemes.

2 Navier–Stokes Solvers

The Navier–Stokes equations are given by three conservation laws.

- Mass conservation: $\frac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{u}) = 0$
- Momentum conservation: $\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla.(\mathbf{u} \otimes (\rho \mathbf{u})) + \nabla.pI - \nabla.\tau = 0$
- Energy conservation: $\frac{\partial \rho E}{\partial t} + \nabla.(\mathbf{u}(\rho E + p)) - \nabla.(\tau \mathbf{u} - q) = 0$

where $\rho, \mathbf{u}, E, \tau, q$ are, respectively density, velocity, energy, viscous tensor and heat flux. Three algorithms are presented. They are all based on the same time discretisation (second order implicit Backward Differentiation Formula), the non-linear problem is solved with the Newton method and linear problems are solved directly ($(L + D)D^{-1}(D + U)$ factorisation). The first method is a classical non-linear domain decomposition method [10, 11] which consists in semi-discretising uniformly in time the system, in applying a global Newton linearisation, then dividing the linear system in several local overlapping subsystems that we can solve in parallel. This algorithm is referred to as the Newton-Schwarz algorithm.

Newton-Schwarz Algorithm:
- Semi-discretisation in time
- Linearisation (Newton)
- Space Schwarz DDM
  - –Solve the local linear system

In some cases, one Schwarz iteration is sufficient to achieve convergence of Newton to the solution of the problem. Space decomposition and linearisation are independent. The next idea is to first do the decomposition and then solve in each sub-domain the non-linear system. This algorithm is the same as the one introduced by Cai and Keyes [1], but using a different linear solver.

Schwarz-Newton Algorithm:
- Semi-discretisation in time
- Space Schwarz DDM
  - –Solve the local non-linear system

To achieve full speed-up performance, a SWR method is used, as it allows local space and time stepping. The whole time interval of study is split into sub-intervals or time windows, then space is decomposed into sub-domains. For each time window the space-time Navier-Stokes equations are solved in each sub-domain in parallel. Boundary conditions are transmitted at the end of the time window.

SWR Algorithm:
- Schwarz DDM over time windows
- For each sub-domain:
  - –Semi-discretisation in time
  - –Solve the local non-linear system

SWR uses time windowing techniques that doesn't degrade the solution and exchanges less information between processors. After each iteration we proceed to the improvement of the interface condition in each sub-domain. This can lead to a completely different time step to satisfy either a stability criteria (for explicit schemes) or an accuracy bound, both based on the CFL number, thus the necessity to locally recompute the time step which is an improvement of the classical SWR algorithm. In this paper we propose, within the SWR iterative process, an adaptive time stepping technique to improve the scheme consistency, thus different time steps in each sub-domain and inside each time window. In the following we shall test the scalability of these three algorithms.

3 Numerical Results

Space discretisation is achieved with finite volumes on cartesian non-conforming grids. The Euler fluxes are computed using the MUSCL-Hancock (Monotone Upstream centred Scheme for Conservative Laws) second order scheme combined with the AUSM$^+$-UP (Advection Upstream Splitting Method) scheme. The advantage of the AUSM$^+$-UP developed by Liou [13] is that it was conceived to be uniformly valid for all speed regimes. The viscous fluxes are computed with a second order Finite Difference scheme. First, we solve the global domain for a simple configuration on CPU and we compare the results with those found using exactly the same second order algorithm, but on GPUs. Then, performances of the different parallel computing strategies (using OpenMP, MPI) are compared on the inviscid and viscous motion of a 2D isolated vortex in an uniform free-stream based on [15] and on the case of the mixing layer. The sub-domains overlap region has the stencil size. We use a second order projection method to exchange data in time and in space. All implicit algorithms are second order in time and space.

3.1 GPU Versus CPU for Euler Equations

First, these algorithms can be accelerated using GPUs. GPUs are used to solve a global problem or a local one using a massive parallel architecture. We start by solving the global problem on a GPU (NVidia Corporation GF110 [Geforce GTX 580] Compute Capability 2.0) with CUDA [3] launched from a CPU and compare its computational cost with one running on a CPU (7.8 GB, 2 Cores at 3.33 GHz) with OpenMP. The computational domain is a rectangular one with an imposed inflow velocity at each time step. On Table 1 is shown the ratio of the computation on a CPU with OpenMP with the computation on CPU-GPU. As can be seen there is a definite gain to be obtained on the CPU-GPU configuration with one domain, and the greater the number of points the better is the ratio. GPU code is portable on any NVidia GPUs using CUDA programming model, though, it should be noted that performances on GPUs vary a great deal depending on the GPU specifications.

Table 1 CPU-OPENMP time cost/CPU-GPU time cost

Grid size	Time step	2D fluxes	Update step	Boundary update	Total
130 × 130	43.08	1.63	8.62	0.31	3.72
260 × 260	109.26	1.71	15.90	1.58	4.65
525 × 525	164.83	2.81	40.37	1.38	6.88
1050 × 1050	392.72	2.58	321.21	2.39	7.80

3.2 2D Isentropic Vortex for Euler Equations

We present results on a convective vortex with $(u_\infty, v_\infty) = (1, 1)$ for a perfect gas: $\gamma = 1.4$, $\frac{p}{\rho^\gamma} = 1$. The computational domain is $[-5., 5.] \times [-5., 5.]$. The initial condition equals the mean flow field plus an isentropic vortex with no perturbation in entropy. We use periodic boundary conditions and Dirichlet transmission conditions. This test is interesting as the isentropic vortex is an exact solution of the Euler equations. At the end of each cycle that lasts 10 s the vortex equals the initial solution.

$$
\begin{aligned}
\rho &= (T_\infty + \delta T)^{\frac{1}{\gamma-1}} = (1 - \tfrac{(\gamma-1)\beta^2}{8\gamma\pi} e^{1-(x^2+y^2)})^{\frac{1}{\gamma-1}} \\
\rho u &= \rho(u_\infty + \delta u) = \rho(1 - \tfrac{\beta}{2\pi} e^{\frac{1-(x^2+y^2)}{2}}) \\
\rho v &= \rho(v_\infty + \delta v) = \rho(1 + \tfrac{\beta}{2\pi} e^{\frac{1-(x^2+y^2)}{2}}) \\
p &= \rho^\gamma \\
e &= \tfrac{p}{\gamma-1} + \tfrac{1}{2}\rho(u^2 + v^2)
\end{aligned}
$$

3.2.1 Accuracy Study

Let us begin with an accuracy study of the Euler equations computing L^2 and L^∞ slopes of errors in the case of non adaptive time steps. First, let us fix the number of sub-domains to 2×2 and a common time step. We increase the global number of space cells from 40×40 cells to 60×60 cells and 80×80 cells (the time step varies in the same ratio as the space step) We consider that we have converged when we reach an error less than a tolerance equal to $1.e - 6$ for both Newton and Schwarz stopping criteria.

As can be seen in Fig. 1, all presented methods are second order in time and close to second order in space, depending on the Van Albada limiter chosen in MUSCL scheme. Velocity, pressure and energy errors behave similarly for all presented methods. The method denoted as Newton in Fig. 1 is the Newton-Schwarz method using only one Schwarz iteration, it only has order one accuracy showing that Schwarz is a good preconditioner for our scheme.

3.2.2 Computational Cost

To evaluate the cost (machine independent), a good indicator is the number of local linear solves, given by the product between the number of Newton iterations and the number of Schwarz iterations. This cost is a linear function of the number of cells. On Table 2 are shown the average number of local linear solves per time step for the Newton-Schwarz (NS) method, the Schwarz-Newton (SN) method and for the

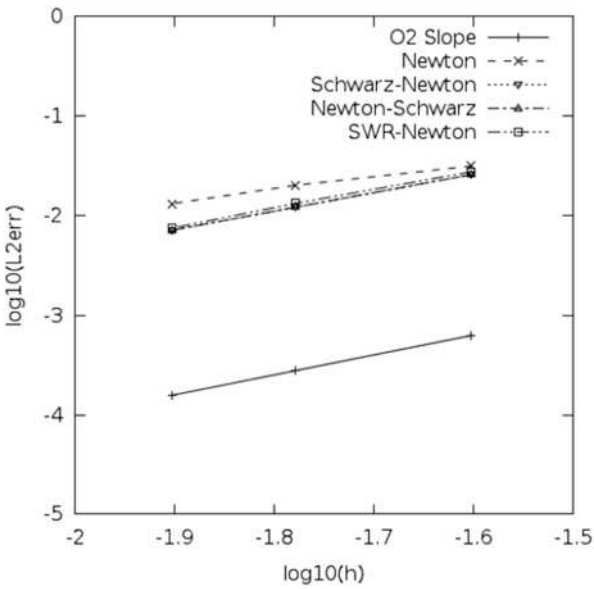

Fig. 1 L^2 error over density field

SWR-Newton (SWR) method for an increasing number of sub-domains with a fixed number of size cells in each sub-domain (weak scalability). The sub-domain size is fixed to 20×20 points and the CFL number has the value 0.5. For the SWR-Newton scheme, we choose δt the same time step on each sub-domain and $\Delta T = 5\delta t$ the time window. The Newton stopping tolerance is set to $1e-6$. The Schwarz convergence tolerance is varying as shown on Table 2. This table shows the good weak scalability of all considered methods. Moreover, it proves that a tolerance of $1e-2$ in the Schwarz stopping criteria decreases the number of linear solves without affecting the precision of the non-linear system. Thus, we can conclude that there is no need to achieve convergence in Schwarz. The SWR method is competitive with the Newton-Schwarz, but two times less efficient than the Schwarz-Newton scheme. On Table 2, in order to compute one time window the four processors communicate in average over all time windows 18.6 times (average number of Schwarz iterations per window) when a SWR-Newton scheme is chosen. In order to reach the same time window the Schwarz-Newton scheme communicates in average 35.45 times (Schwarz iterations $\times$ window size) and the Newton-Schwarz scheme communicates in average 250 times (Newton iterations $\times$ Schwarz iterations $\times$ window size). The SWR method is thus ideal for clusters with high latencies. Note: It should be mentioned that higher order coupling conditions like unsteady Robin type conditions can improve the efficiency of the algorithm and should positively influence the number of Schwarz iterates (cf. [7]).

The adaptive time step SWR method converges to the solution in exactly the same way as the fixed time step SWR method. The gain of the SWR method comes from the improved stability of the scheme since the time step is recomputed at each iteration thus less communication between the sub-domains as it appears that when

Table 2 Weak scaling of the schemes

		Schwarz tol. = $1.e-6$			Schwarz tol. = $1.e-3$			Schwarz tol. = $1.e-2$		
Scheme	Iterations\Sub-domains	4	9	16	4	9	16	4	9	16
NS	Newton iterations	8.48	8.11	6.94	8.48	8.11	6.93	8.48	8.11	9.29
NS	Schwarz iterations	5.89	6.28	6.41	5.26	5.70	5.78	4.71	5.00	5.66
NS	Linear solvers	50.00	51.02	44.51	44.64	46.33	40.15	39.89	40.63	52.62
SN	Newton iterations	3.07	2.72	2.28	5.00	4.18	3.28	5.93	4.90	5.31
SN	Schwarz iterations	7.09	7.49	7.15	3.55	4.0	4.0	2.46	3.0	3.0
SN	Linear solvers	21.79	20.40	16.31	17.78	16.75	13.15	14.63	14.72	15.93
SWR	Newton iterations	7.54	6.62	5.41	7.52	6.61	5.37	7.49	6.56	7.13
SWR	Schwarz iterations	18.6	14.28	19.25	7.08	7.98	7.44	4.36	4.77	5.66
SWR	Linear solvers	140.6	95.3	104.28	53.28	52.78	39.99	32.75	31.43	40.40

Computational costs

the coupling conditions are improved, larger times steps are usually needed. This also leads to less CPU memory when fewer coupling conditions need to be stored.

3.3 Sound Generation in a 2D Low-Reynolds Mixing Layer

The second case presented here if the case of a 2D low-Reynolds mixing layer where a high precision scheme is required. It is studied especially focusing on the acoustic waves emitted by the vortex pairings in a perturbed mixing layer. The flow configuration is the same as the one proposed by Colonius et al. [2] consisting in a slightly perturbed hyperbolic tangential shape velocity profile, $u = \bar{u} + 0.125\,tanh(2y)$, with $\bar{u} = (u_\infty + u_{-\infty})/2$ and $u_\infty = 0.5$, $u_{-\infty} = 0.25$, and $\rho_\infty = \rho_{-\infty} = 1$ and $p_\infty = p_{-\infty} = 1/\gamma$, respectively, with $\gamma = 1.4$. We fix the Reynolds number at 250 and add a sponge layer as shown in Fig. 2 to absorb the flow. This is a particularly sensitive case in acoustics and phenomena are quite different within each subdomain. The results presented on Table 3 are for simulations between $t = 200\,\mathrm{s}$ and $t = 250\,\mathrm{s}$, interval inside which all sub-domains are interacting. The initial solution was computed with an explicit second order Runge-Kutta method. We have fixed the stopping criterion in the Newton algorithm to a tolerance of $1.e-4$ and the stopping criterion of the Schwarz decomposition to a tolerance of $1.e-2$ (cf. Sect. 3.2) which gives a good solution. The time window inside the SWR methods equal 5 times the smallest global time step and the global domain was divided in 22 sub-domains : 18 sub-domains of equal size 107×21 cells in the middle region and 4 sponge sub-domains with 107×41 in the sponge area. The number of linear solves is no longer a good measure since sub-domains with different size have been computed and we adapt the time step after each iteration for SWR and for all time steps in SWRA the adaptive SWR. On Table 3 we vary the

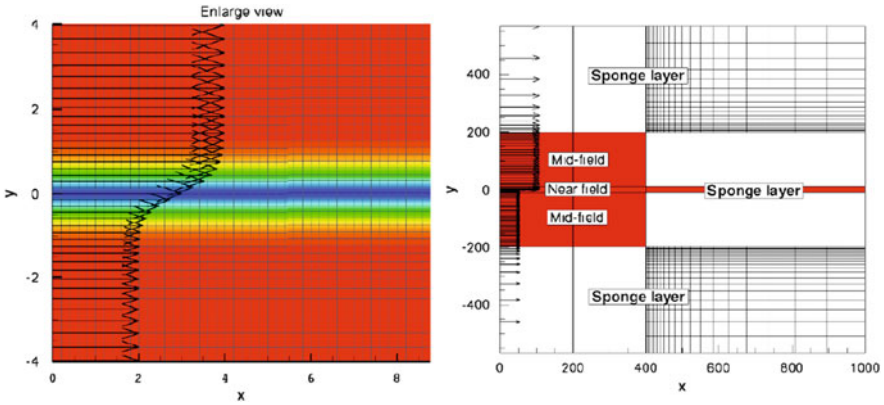

Fig. 2 Mixing layer acoustic pressure field. Initial condition (*left*) and computational domain with sponge layer (*right*)

Table 3 Global computational costs for $\Delta T = 5\delta t$, Schwarz tol. = $1.e-2$ and Newton tol. = $1.e-4$

Scheme\CFL	0.5	1	2	5	10
NS	333.56	167.65	116.45	24.20	18.79
SN	129.97	76.07	89.41	26.76	10.45
SWR	189.13	189.65	121.77	21.90	5.87
SWRA	189.82	191.08	121.54	21.86	5.12

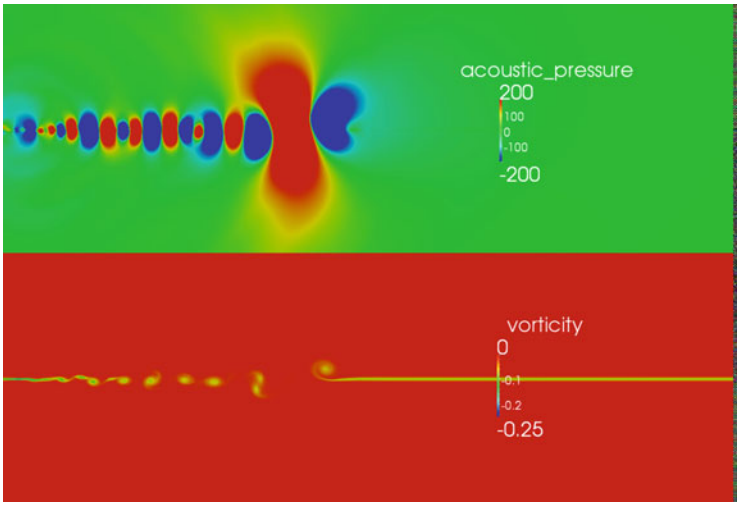

Fig. 3 Mixing layer acoustic pressure field (*top*) and vorticity (*bottom*)

time window length and show only the total computational time cost for all three methods. For low CFL (less than 2) SWR is less efficient than SN. For higher CFL, SWR becomes the most efficient, the SWR with adaptive step becoming the leader in terms of performance.

Results obtained with the time adaptive SWR scheme (see Fig. 3) compare well with those obtained with an explicit third order Runge Kutta Discontinuous Galerkin solver developed by Halpern et al. [8].

4 Conclusion and Remarks

A variation on the non-linear SWR algorithm has been developed using an adaptive time stepping approach to simulate 2D multi-scale Euler and Navier-Stokes problems. The above results show that the method has the ability to treat large data systems without loss of parallel efficiency. This SWR algorithm has similar computational efficiency as the original SWR and adds a new flexibility to the SWR

method. There are at least three ways to improve the SWR technique. One is to optimize the time space interface condition, another is to implement the pipeline SWR iterations as presented by Ong et al. [14] and of course the use of GPUs that can considerably improve the efficiency.

References

1. X.-C. Cai, D.E. Keyes, Nonlinearly preconditioned inexact Newton algorithms. SIAM **24** (1),183–200 (2002)
2. T. Colonius, S.K. Lele, P. Moin, Sound generation in a mixing layer. J. Fluid Mech. **330**,375–409 (1997)
3. CUDA (2015), http://www.nvidia.com/object/cuda_home_new.html
4. M.J. Gander, Overlapping Schwarz waveform relaxation for parabolic problems, in *DD10 Proceedings*, vol. 218 (1998), pp. 425–431
5. M.J. Gander, A.M. Stuart, Space-time continuous analysis of waveform relaxation for the heat equations. SIAM **19**(6), 2014–2031 (1998)
6. F. Haeberlein, Time-space domain decomposition methods for reactive transport. Ph.D. thesis, University Paris 13, 2011
7. F. Haeberlein, L. Halpern, Optimized Schwarz waveform relaxation for nonlinear systems of parabolic type, in *DD21 Proceedings* (2012)
8. L. Halpern, J. Ryan, M. Borrel, Domain decomposition vs. overset Chimera grid approaches for coupling CFD and CAA, in *ICCFD7* (2012)
9. R. Jeltsch, B. Pohl, Waveform relaxation with overlapping splittings. SIAM **16**(1), 40–49 (1995)
10. D.E. Keyes, Domain decomposition in the mainstream of computational science, in *DD14 Proceedings* (2002)
11. D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications. J. Comput. Phys. **193**(2), 357–397 (2004)
12. E. Lelarasmee, A.E. Ruehli, A.L. Sangiovanni-Vincentelli, The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE **1**(3), 131–145 (1982)
13. M.-S. Liou, A sequel to AUSM, part II: AUSM$^+$-up for all speeds. J. Comput. Phys. **214**(1), 137–170 (2006)
14. B. Ong, S. High, F. Kwok, Pipeline Schwarz waveform relaxation, in *22nd DDM Conference* (2013, submitted)
15. H.C. Yee, N.D. Sandham, M.J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters. J. Comput. Phys. **150**(1), 199–238 (1999)