Preface

Parallel CFD 2008, the twentieth in the high-level international series of meetings featuring different aspect of parallel computing in computational fluid dynamics and other modern scientific domains was held May 19 – 22, 2008 in Lyon, France.

The themes of the 2008 meeting included the traditional emphases of this conference, and experiences with contemporary architectures. Around 70 presentations were included into the conference program in the following sessions:

- Parallel Algorithms and solvers
- Parallel performances with contemporary architectures
- Structured and unstructured grid methods, boundary methods
- Software framework and components architecture
- CFD applications (Bio fluid, environmental problem) Lattice Boltzmann method and SPH
- Optimisation in Aerodynamics

This book presents an up-to-date overview of the state of the art in Parallel Computational Fluid Dynamics from Asia, Europe, and North America. This reviewed proceedings included about sixty percent of the oral lectures presented at the conference.

The editors.
Parallel CFD 2008 was organized by the Institut Camille Jordan of the University of Lyon 1 in collaboration with the Center for the Development of the Parallel Scientific Computing.

The Scientific Committee and Local Organizers of Parallel CFD 2008 are delighted to acknowledge the generous sponsorship of the following organizations, through financial or in-kind assistance. Assistance of our sponsors allowed to organize scientific as well as social program of the conference.

Scientific Committee
University Lyon 1

Intel, Germany

SGI, France

Institut Camille Jordan
University Lyon 1

Région Rhône-Alpes

Center for Development of
University Lyon 1

Fluorem

Modelys

Many people worked to organize and execute the conference. We are especially grateful to all members of the international scientific committee. We also want to thank the key members of the local organizing committee David Guibert, Toan Pham Duc, Patrice linel, Simon Pomarede, Thomas Dufaud, Nicolas Kielbasievich, Daniel Fogliani, Fabienne Oudin, Brigitte Hautier, Sandrine Montingy.

We also thank our colleagues Frédéric Desprez from the Laboratoire d’Informatique du Parallélisme (LIP) Ecole Normale Supérieure de Lyon, Michel Lance from the Laboratoire de Mécanique des Fluides et d’Accoustique (LMFA) Ecole Centrale de Lyon, and Patrick Quéré from the Computer Science Laboratory for mechanics and Engineering Sciences (LIMSI) for their help to promote this event.

Damien Tromeur-Dervout
Chairman, Parallel CFD 2008.
Contents

Part I Invited speakers

Large Scale Computations in Nuclear Engineering: CFD for Multiphase Flows and DNS for Turbulent Flows with/without Magnetic Field
Tomoaki Kunugi, Shin-ichi Satake, Yasuo Ose, Hiroyuki Yoshida, Kazuyuki Takase 3

Scalable algebraic multilevel preconditioners with application to CFD
Andrea Aprovitola, Pasqua D’Ambra, Filippo Denaro, Daniela di Serafino, Salvatore Filippone 15

Acceleration of iterative solution of series of systems due to better initial guess
Damien Tromeur-Dervout, Yuri Vassilevski 29

Part II Optimisation in Aerodynamics Design

Aerodynamic Study of Vertical Axis Wind Turbines
Merim Mukanović, Gunther Brenner, Ardavan Rahimi 43

Parallel Shape Optimization of a Missile on a Grid Infrastructure
Erdal Oktay, Osman Merttopcuoglu, Cevat Sener, Ahmet Ketenci, Hasan U. Akay 51

Analysis of Aerodynamic Indices for Racing Sailing Yachts: a Computational Study and Benchmark on up to 128 CPUs.
Ignazio Maria Viola, Raffaele Ponzini, Daniele Rocchi, Fabio Fossati 61

Parallel implementation of fictitious surfaces method for aerodynamic shape optimization
S. Peigin, B. Epstein, S. Gali 71
Path Optimization of Dual Airfoils Flapping in a Biplane Configuration with RSM in a Parallel Computing Environment
Mustafa Kaya, Ismail H. Tuncer ................................. 83

Part III Grid methods

Convergence Improvement Method for Computational Fluid Dynamics Using Building-Cube Method
Takahiro Fukushige Toshihiro Kamatsuchi Toshiyuki Arima and Seiji Fujino . . . 93

Aerodynamic Analysis of Rotor Blades using Overset Grid with Parallel Computation
Dong-Kyun Im, Seong-Yong Wie, Eugene Kim, Jang-Hyuk Kwon, Duck-Joo Lee, and Ki-Hoon Chung, and Seung-Bum Kim .......................... 101

Large scale massively parallel computations with the block-structured elsA CFD software
M. Gazaix, S. Mazet, M. Montagnac ................................. 111

Applications on Hybrid Unstructured Moving Grid Method for Three-Dimensional Compressible Flows
Hiroya Asakawa Masashi Yamakawa and Kenichi Matsuno .................. 119

Shinichi ASAO, Sadanori ISHIHARA, Kenichi MATSUNO, Masashi YAMAKAWA .................................................. 127

Part IV Boundary methods

Flow Computations Using Embedded Boundary Conditions on Block Structured Cartesian Grid
T. Kamatsuch, T. Fukushige, K. Nakahashi ................................. 137

Computation of Two-phase Flow in Flip-chip Packaging Using Level Set Method
Tomohisa Hashimoto, Keiichi Saito, Koji Morinishi,, Nobuyuki Satofuka ...... 145

A Parallel Immersed Boundary Method for Blood-like Suspension Flow Simulations
F. Pacull, M. Garbey ............................................... 153
Part V  High Order methods

3D Spectral Parallel Multi-Domain computing for natural convection flows
S. Xin, J. Chergui, P. Le Quéré ........................................ 163

3D time accurate CFD simulations of a centrifugal compressor
Andreas Lucius, Gunther Brenner ........................................ 173

Part VI  Parallel Algorithms and Solvers

Multicolor SOR Method with Consecutive Memory Access Implementation in a Shared and Distributed Memory Parallel Environment
Kenji Ono, Yasuhiro Kawashima ........................................ 183

Proper Orthogonal Decomposition In Decoupling Large Dynamical Systems
Toan Pham, Damien Tromeur-Dervout ..................................... 193

Performance Analysis of the Parallel Aitken-Additive Schwarz Waveform Relaxation Method on Distributed Environment
Hatem Ltaief, Marc Garbey .............................................. 203

Aitken-Schwarz Acceleration not based on the mesh for CFD
D. Tromeur-Dervout ......................................................... 211

From extruded-2D to fully-3D geometries for DNS: a Multigrid-based extension of the Poisson solver
A. Gorobets, F. X. Trias, M. Soria, C. D. Pérez-Segarra, A. Oliva ............... 219

Parallel direct Poisson solver for DNS of complex turbulent flows using Unstructured Meshes
R.Borrell, O.Lehmkuhl, F.X.Trias, M.Soria, A.Oliva ................................ 227

A numerical scheme for the computation of phase transition in compressible multiphase flows
Vincent Perrier ................................................................. 235

Part VII  Lattice Boltzman and SPH Methods

Lattice Boltzmann Simulations of Slip Flow of Non-Newtonian Fluids in Microchannels
Ramesh K. Agarwal and Lee Chusak ........................................ 247

Multiple Relaxation Time Lattice Boltzmann simulation of binary droplet collisions
Ernesto Monaco, Kai H. Luo, Gunther Brenner .................................. 257
High-Performance Computing and Smoothed Particle Hydrodynamics
C. Moulinec, R. Issa, D. Latino, P. Vezolle, D.R. Emerson and X.J. Gu 265

Part VIII software Framework and Component Architecture

An integrated object-oriented approach for parallel CFD
Dominique Eyheramendy, David Loureiro, Fabienne Oudin-Dardun 275

Fast Multipole Method for particle interactions: an open source parallel library component
F. A. Cruz, M. G. Knepley, L. A. Barba 285

Hybrid MPI-OpenMP performance in massively parallel computational fluid dynamics
G. Houzeaux, M. Vázquez, X. Sáez and J.M. Cela 293

Hierarchical adaptive multi-mesh partitioning algorithm on heterogeneous systems
Y. Mesri, H. Digonnet, T. Coupez 299

Part IX Parallel Performance

Towards Petascale Computing with Parallel CFD codes
A.G. Sunderland, M. Ashworth, C. Moulinec, N. Li, J. Uribe, Y. Fournier 309

Scalability Considerations of a Parallel Flow Solver on Large Computing Systems
Erdal Yılmaz, Resat U. Paylı, Hassan U. Akay, Akin Ecer, and Jingxin Liu 321

Large Scaled Computation of Incompressible Flows on Cartesian Mesh Using a Vector-Parallel Supercomputer
Shun Takahashi, Takashi Ishida, Kazuhiro Nakahashi, Hiroaki Kobayashi, Koki Okabe, Youichi Shimomura, Takashi Soga, Akihiko Musa 331

Dynamic Load Balancing on Networked Multi-core Computers
Stanley Chien, Gun Makinabakan, Akin Ecer, and Hasan Akay 339

On efficiency of supercomputers in CFD simulations
Andrey V. Gorobets, Tatiana K. Kozubskaya, Sergey A. Soukov 347

Part X Environment and biofluids applications

Numerical Study of Pulsatile Flow Through Models of Vascular Stenoses with Physiological Waveform of the Heart
Jason D. Thompson, Christian F. Pinzn and Ramesh K. Agarwal 357
Fluid Flow - Agent Based Hybrid Model for the Simulation of Virtual Prairies
Marc Garbey, Cendrine Mony, Malek Smaoui .......................... 369

HPC for hydraulics and industrial environmental flow simulations
Réza Issa, Fabien Decung, Emile Razafindrakoto, Eun-Sug Lee, Charles Moulinec, David Latino, Damien Violeau, Olivier Boiteau .................. 377

Multi-parametric intensive stochastic simulations for hydrogeology on a computational grid
J. Erhel, J.-R. de Dreuzy, E. Bresciani .................................. 389

Parallel computation of pollutant dispersion in industrial sites
J. Montagnier, M. Buffat, D. Guibert ................................. 399

Part XI General fluid

3D Numerical Simulation Of Gas Flow Around Reentry Vehicles
S.V. Polyakov, T.A. Kudryashova, E.M. Kononov, A.A. Sverdlin ............. 409

Effective Parallel Computation of Incompressible Turbulent Flows on Non-uniform Grid
Hidetoshi Nishida, Nobuyuki Ichikawa ................................. 417

Secondary flow structure of turbulent Couette-Poiseuille and Couette flows inside a square duct
Hsin-Wei Hsu, Jian-Bin Hsu, Wei Lo, Chao-An Lin ...................... 425
Invited speakers
Large Scale Computations in Nuclear Engineering: CFD for Multiphase Flows and DNS for Turbulent Flows with/without Magnetic Field

Tomoaki Kunugi¹, Shin-ichi Satake², Yasuo Ose¹, Hiroyuki Yoshida³, and Kazuyuki Takase³

¹ Kyoto University, Yoshida, Sakyo, Kyoto 6060-8501, Japan
² Tokyo University of Science, 2641 Yamazaki, Noda 278-8510, Japan
³ Japan Atomic Energy Agency, 2-4, Shirakata, Tokai-mura 319-1195, Japan
   kunugi@nucleng.kyoto-u.ac.jp

Abstract. Large scale computations are being carried out in nuclear engineering fields such as light water reactors, fast breeder reactors, high temperature gas-cooled reactors and nuclear fusion reactors. The computational fluid dynamics (CFD) regarding not only the single-phase flows but also the two-phase flow plays an important role for the developments of advanced nuclear reactor systems. In this review paper, some examples of large scale computations in nuclear engineering fields are illustrated by using a parallel visualization.

Key words: Direct numerical simulation; Multiphase flows; Turbulent flows; Parallel visualization; Magnetohydrodynamics; Nuclear reactors; Fusion reactors.

1 Numerical Simulation of Boiling Phenomena

It is important to remove the heat from industrial devices and nuclear reactors with a high heat flux to insure their safety. In order to enhance the heat transfer, phase change phenomena such as evaporation and condensation have to be utilized, so that it is important to understand the mechanism of boiling to design industrial devices. Although many researchers have experimentally studied the boiling phenomena, it has not been clarified yet because it consists of a lot of complicated phenomena. As for pool boiling experiments, Kenning & Yan measured the spatial and temporal variations of wall temperature in nucleate boiling by using a liquid crystal thermometry. They pointed out the importance of non-uniform wall temperature distribution and suggested that the transient heat conduction model proposed by Mikic & Rohsenow [2] was unrealistic because of the assumption of uniform transient heat conduction. It is relatively difficult to perform numerical analysis of boiling phenomenon because it includes the phase change. On the other hands, a critical heat flux (CHF) is also very important for high heat flux removal. However, the empirical correlation of the CHF is used in most designs. In general, the prediction of CHF is very
difficult because of the complexity of relation between the nucleation boiling and the bubble departure due to flow convection. Welch [3] carried out the numerical study on two-dimensional two-phase flow with a phase change model using a finite volume method combined with a moving grid, however it can be applied only to a little deformation of gas-liquid interface. Son & Dhir [4] also carried out the two-dimensional pool boiling simulation using a finite difference method with a moving grid. Recently, Juric & Tryggvason [5] conducted a film boiling with a front tracking method by Unverdi & Tryggvason [6]. They pointed out the importance of the temporal and spatial temperature distribution of the heating plate: heat conduction in the slab. The author developed a new volume tracking method, so-called gMARS: Multi-interface Advecton and Reconstruction Solver[7].

This section describes that the MARS is applied to the force convective flow boiling in the channel with an appropriate phase change model: a model for boiling and condensation phenomena based on a homogeneous nucleation and a well-known enthalpy method. This model is good for metal casting problems because of no superheating of liquid. As for the water, it has to be considered the liquid superheat for nucleate boiling phenomena. The direct numerical simulations with this phase change model for pool nucleate boiling and forced convective flow boiling have been performed. The aims of this study are to develop a direct numerical method to simulate boiling phenomena and to simulate the three-dimensional pool nucleate boiling and forced convective flow boiling by the direct numerical simulation (DNS) based on the MARS combined the enthalpy method considered the liquid superheat as the phase change model.

1.1 Numerical Simulation based on MARS

**Governing equations.**

In this section, the direct numerical multiphase flow solver (MARS) is briefly explained. As for $m$ fluids including the gas and liquid, the spatial distribution of fluids can be defined as

$$\langle F \rangle = \sum F_m = 1.0$$  (1)

The continuity equation of the multiphase flows for $m$ fluids:

$$\frac{\partial F_m}{\partial t} + \nabla \cdot (F_m U) - F_m \nabla U = 0$$  (2)

The momentum equation with the following CSF (Continuum Surface Force) model proposed by Brackbill [9] is expressed as:

$$\frac{\partial U}{\partial t} + \nabla \langle U U \rangle = G - \frac{1}{\langle \rho \rangle} \nabla P - \nabla \cdot \tau + \frac{1}{\langle \rho \rangle} F_V$$  (3)

CSF model:

$$F_V = \sigma \kappa n \langle \rho \rangle / \bar{\rho}$$  (4)
here, the mean density at interface, \( \bar{\rho} = (\rho_g + \rho_l)/2 \), the suffix \( g \) denotes vapor and \( l \) for water. \( \kappa \) is curvature of the surface, \( \sigma \) the surface tension coefficient and \( n \) is the normal vector to the surface.

The momentum equation (3) can be solved by means of the well-known projection method. The Poisson equation for the pressure can be solved by the ILUBCG method. Finally, the new velocity field can be obtained. Once the velocity field can be obtained, it can be transported the volume of fluid by the MARS, i.e., a kind of PLIC (Piecewise Linear Calculation) volume tracking procedure [24] for Eq. (2). The detail description of the solution procedure is described in the reference [7].

The energy equation is expressed as:

\[
\frac{\partial}{\partial t} (\rho C_v) T + \nabla \cdot (\rho C_v U T) = \nabla \cdot (\lambda \nabla T) - P (\nabla \cdot U) + Q, \tag{5}
\]

where \( C_v, T, \lambda, Q \) is specific heat, temperature, heat conductivity, heat generation term, respectively. The second term of the right hand side of the equation (5), the Clausius-Clapeyron relation is considered as the work done by the phase change.

**Phase Change Model.**

As mention in the introduction section, nucleate boiling phenomena need to be modeled. One of ideas for this modeling can be considered:

Nucleate boiling model = Nucleation model + Bubble growth model

**Nucleation model:** This model gives the homogeneous superheat limit of liquid and the size of nucleus of bubble. A superheat limit \( T_s \) is got from the kinetic theory [8] or the usual cavity model. Typically \( T_s \) is around 110°C in water pool boiling at an atmospheric pressure. The equilibrium radius \( r_e \) of nucleus corresponding to \( T_s \) can be calculated by Eq. (6) based on the thermodynamics. The \( F \)-value of nucleus corresponding to the ratio of a cell size to a size of nucleus where the shape of nucleus is assumed to be a sphere is given to a computational cell with greater than \( T_s \). In this study, \( T_s \) is a parameter. Although \( T_s \) varies in spatial on the heated surface from the experiment [1], \( T_s \) is assumed to be uniform on the heated surface in the present study. Therefore, particular nucleation sites are not specified.

\[
r_e = \frac{2\sigma}{P_{sat} (T_l) \exp \{ v_l (P_l - P_{sat} (T_l)) / RT_l \} - P_l} (T_l \geq T_s) \tag{6}
\]

where \( r_e \) is a radius of bubble embryo, \( \sigma \) is an interfacial tension, \( T_l \) is a temperature of liquid, \( P_{sat} \) is a pressure corresponding saturation conditions, \( P_l \) is a pressure of liquid, \( v_l \) is a volume of liquid per unit mass, and \( R \) is the ideal gas constant on a per unit mass basis.

**Bubble growth model:** Increasing the temperature, liquid water becomes vapor partially if the temperature of liquid is greater than the liquid saturation line \( (T_l) \), i.e., liquid-gas mixture (two-phase region) and then eventually becomes the superheated gas phase if the temperature is greater than the gas saturation lime \( (T_g) \). This process can be treated by the enthalpy method.
1.2 Results and Discussions on Forced Convective Flow Boiling

In order to get some insights into the mechanism of three-dimensional forced convective subcooled flow boiling, Figure 1 shows the computational domain that is a three-dimensional vertical channel and the sidewall is heated with constant heat flux because of considering the spatial and temporal variations of temperature in the solid sidewall. The length of flow channel is 60 mm, 6 mm in height and 5 mm in width. The computational domain is a half channel because of the symmetry. The heating conditions are as follows: 0.3 mm in length from inlet is adiabatic and following 50 mm by heating of a constant heat flux, 2.4 MW/m² from the outside of solid sidewall of 0.6 mm in thickness and the remaining 9.7 mm is also adiabatic. The inlet mean velocity is 0.5 m/s with a parabolic profile. The no-slip condition at the sidewall, the slip velocity at the symmetric boundary and constant pressure at the outlet are imposed as the boundary conditions. The periodic velocity and temperature conditions are applied to the spanwise boundaries. The water pressure is atmospheric and the degree of water subcooling is 20 K. The solid wall is assumed to be a stainless steel. The degree of superheat is set to be 50 K. The computational cell is uniform cubic shape and the size of cell is 100 μm and the number of cells is $36(x) \cdot 600(y) \cdot 50(z) = 1,080,000$. Time increment is 5 μsec. The fictitious temperature difference used in the enthalpy method is $ΔT = 0.1$ K.

Resulting from the forced convective flow boiling computation as shown in Fig. 2, the series of bubble growth are depicted as black spots, and the gray contours show the temperature distribution at $x=0.25$ mm every 52.5 msec. From the temperature
Large Scale Computations in Nuclear Engineering 7

evolution, the thermal boundary layer has been developing during computation and does not reach to the equilibrium state at the downstream in this computation, so that it is necessary to carry out much longer computation. The maximum size of bubble is around 1 mm at the present stage. It is interesting that the higher temperature stagnation region is formed in the middle of the channel and three higher temperature streaks are observed in the thermal boundary layer. The bubble generated in the upstream region becomes just like an obstacle and makes a high temperature stagnation or recirculation region behind the nucleated bubbles. Eventually, another bubble will generate in that stagnation or recirculation region because the degree of liquid subcooling could be decreased, i.e., it can be saying a “chain generation of boiling bubbles.”

![Bubble growth and temperature distribution every 52.5 msec](image)

**Fig. 2.** Bubble growth and temperature distribution every 52.5 msec

2 CFD for Two-Phase Flow Behaviors in Nuclear Reactor

Subchannel analysis codes [11]-[13] and system analysis codes [14, 15] are usually used for the thermal-hydraulic analysis of fuel bundles in nuclear reactors. As for the former, however, many constitutive equations and empirical correlations based on experimental results are needed to predict the water-vapor two-phase flow behavior. If there are no experimental data such as an advanced light-water reactor which has been studied at the Japan Atomic Energy Agency (JAEA) in Japan and named as
a reduced moderation water reactor (RMWR), it is very difficult to obtain the precise predictions [16, 17, 18]. The RMWR core has a remarkably narrow gap spacing between fuel rods (i.e., around 1 mm) and a triangular tight lattice fuel rod configuration in order to reduce the moderation of the neutron. In such a tight-lattice core, there is no sufficient information about the effects of the gap spacing and the effect of the spacer configuration on the two-phase flow characteristics. Therefore, in order to analyze the water vapor two-phase flow dynamics in the tight-lattice fuel bundle, a large-scale simulation under the full bundle size condition is necessary. The Earth Simulator [19] enables that lots of computational memories are required to attain the two-phase flow simulation for the RMWR core.

In JAEA, numerical investigation on the physical mechanisms of complicated thermal-hydraulic characteristics and the multiphase flow behavior with phase change in nuclear reactors has been carried out. In this numerical research, some of the authors in JAEA pointed out the improvements of the conventional reactor core thermal design procedures and then proposed a predicting procedure for two-phase flow characteristics inside the reactor core more directly than the conventional procedures for the first time in the world by reducing the usage of constitutive and empirical equations as much as possible [20]. Based on this idea, a new thermal design procedure for advanced nuclear reactors with the large-scale direct simulation method (TPFIT: Two-Phase flow simulation code using advanced Interface Tracking) [21] has been developed at JAEA. Especially, thermal hydraulic analyses of two-phase flow positively for a fuel bundle simulated by the full size using the Earth Simulator are performed [22]. This section describes the preliminary results of the large-scale water-vapor two-phase flow simulation in the tight-lattice fuel bundle of the RMWR core by the TPFIT code.

2.1 Numerical Simulation of Two-Phase Flow Behavior in 37 RMWR Fuel Rods

The TPFIT code is based on the CIP method [23] using the modified interface-tracking method [24]. The surface tension of bubble is calculated using the CSF [9]. Figure 3 shows the computational geometry consisting of 37 RMWR fuel rods. The geometry and dimensions simulate the experimental conditions done by JAEA [25]. Here, the fuel rod outer diameter is 13 mm and the gap spacing between each rod is 1.3 mm. The casing has a hexagonal cross section and a length of one hexagonal side is 51.6 mm. An axial length of the fuel bundle is 1260 mm. The water flows upward from the bottom of the fuel bundle. A flow area is a region in which deducted the cross-sectional area of all fuel rods from the hexagonal flow passage. The spacers are installed into the fuel bundle at the axial positions of 220, 540, 750 and 1030 mm from the bottom. The axial length of each spacer is 20 mm. Inlet conditions of water are as follows: temperature 283°C, pressure 7.2 MPa, and flow rate 400 kg/m²s. Moreover, boundary conditions are as follows: fluid velocities for x-, y- and z-directions are zero on every wall (i.e., an inner surface of the hexagonal flow passage, outer surface of each fuel rod and surface of each spacer); velocity profile at the inlet of the fuel bundle is set to be uniform. The present simulations were carried
out under the non-heated isothermal flow condition in order to remove the effect of heat transfer due to the fuel rods to the fluid. A setup of a mixture condition of water and vapor at the heating was performed by changing the initial void fraction of water at the inlet of the analytical domain.

![Image of three-dimensional analytical geometry of a tight-lattice fuel bundle](image)

**Fig. 3.** Outline of three-dimensional analytical geometry of a tight-lattice fuel bundle

### 2.2 Results and Discussions

Figure 4 shows an example of the predicted vapor structure around the fuel rods. Here, the distribution of void fraction within the region from 0.5 to 1 is shown: 0.5 indicates just an interface between the water and vapor and is shown by green; and 1 indicates the non-liquid vapor and is shown by red. Vapor flows from the upstream to downstream like a streak through the triangular region, and the interaction of the vapor stream to the circumferential direction is not seen. On the other hand, since the vapor is disturbed behind a spacer, the influence of turbulence by existence of the spacer can be predicted.

In order to predict the water-vapor two-phase flow dynamics in the RMWR fuel bundle and to reflect them to the thermal design of the RMWR core, a large-scale simulation was performed under a full bundle size condition using the Earth Simulator. Details of water and vapor distributions around fuel rods and a spacer were
clarified numerically. A series of the present preliminary results were summarized as follows: 1) The fuel rod surface is encircled with thin water film; 2) The bridge formation by water film appears in the region where the gap spacing between adjacent fuel rods is narrow; 3) Vapor flows into the triangular region where the gap spacing between fuel rods is large; 4) A flow configuration of vapor shows a streak structure along the triangular region.

3 DNS for Turbulent Flows with/without Magnetic Field

On the other hand, in the gas-cooled reactor and the fast breeder reactors the coolant is a single-phase flow at mostly turbulent situation. Direct numerical simulations (DNSs) for turbulent flows have been carried out to investigate the turbulent structure in the flow passage and around each fuel rod surface at Reynolds number of 78,000 as shown in Fig. 5. A second order finite difference method is applied to the spatial discretization, the 3rd order Runge-Kutta method and the Crank-Nicolson method are applied to the time discretization, and the time advancing scheme is the fractional time step is used for the coupling scheme, so called, Dukowcz-Dvinsky scheme. The number of used central processing unit is 1,152 and it corresponds to 144 nodes. The total memory is 2 terabytes and the total number of computational grid is 7,200
million points [26, 27]. The flow visualization is very important to grasp the entire picture of the flow behavior, so that a parallel visualization technique is applied in this case. As for the thermofluid behavior in fusion reactors, the magnetohydrodynamic (MHD) effect is very important. DNS for turbulent flow in a parallel channel has been performed as shown in Fig. 7 [28] drawn by the parallel visualization. The turbulent structure is suppressed by the magnetic field, i.e., Lorentz force: Hartmann numbers are 0 (non-MHD) and 65 (MHD).

![Image](image.png)

**Fig. 5.** Second invariant contour surface of velocity tensor of single-phase turbulent flow in a pipe Re=78,000, 7,200 Mega grids, 2 Tera Bytes

### 4 Conclusion

The computational fluid dynamics (CFD) regarding not only the single-phase flows but also the two-phase flow plays an important role for the developments of advanced nuclear reactor systems. To establish the large-scale simulation procedure with higher prediction accuracy is very important for the detailed reactor-core thermal-design in the nuclear engineering. Moreover, the parallel visualization technique is very useful to understand the detailed flow and heat transfer phenomena.
Fig. 6. Magnetic field effect on turbulent flow in a parallel channel: Second invariant contour of velocity tensor


large-scale simulations Proc. 5th International Conference on Multiphase Flow (ICMF04), No.373, Yokohama, Japan, June, 2004.


[28] Satake, S., Kunugi, T., Takase, K., Ose, Y Direct numerical simulation of turbulent channel flow under a uniform magnetic field for large-scale structures at high Reynolds number Physics of Fluid, 18, 125106, 2006.
Scalable algebraic multilevel preconditioners with application to CFD

Andrea Aprovitola$^1$, Pasqua D’Ambra$^2$, Filippo Denaro$^1$, Daniela di Serafino$^3$, and Salvatore Filippone$^4$

$^1$ Department of Aerospace and Mechanical Engineering, Second University of Naples, via Roma 29, I-81031 Aversa, Italy, andrea.aprovitola@tele2.it & denaro@unina.it
$^2$ Institute for High-Performance Computing and Networking (ICAR), CNR, via Pietro Castellino 111, I-80131 Naples, Italy, pasqua.dambra@na.icar.cnr.it
$^3$ Department of Mathematics, Second University of Naples, via Vivaldi 43, I-81100 Caserta, Italy, daniela.diserafino@unina2.it
$^4$ Department of Mechanical Engineering, University of Rome “Tor Vergata”, viale del Politecnico 1, I-00133, Rome, Italy, salvatore.filippone@uniroma2.it

Abstract. The solution of large and sparse linear systems is one of the main computational kernels in CFD applications and is often a very time-consuming task, thus requiring the use of effective algorithms on high-performance computers. Preconditioned Krylov solvers are the methods of choice for these systems, but the availability of “good” preconditioners is crucial to achieve efficiency and robustness. In this paper we discuss some issues concerning the design and the implementation of scalable algebraic multilevel preconditioners, that have shown to be able to enhance the performance of Krylov solvers in parallel settings. In this context, we outline the main objectives and the related design choices of MLD2P4, a package of multilevel preconditioners based on Schwarz methods and on the smoothed aggregation technique, that has been developed to provide scalable and easy-to-use preconditioners in the Parallel Sparse BLAS computing framework. Results concerning the application of various MLD2P4 preconditioners within a large eddy simulation of a turbulent channel flow are discussed.

Key words: Preconditioning technique, Schwarz domain decomposition, Krylov methods.

1 Introduction

The solution of linear systems is ubiquitous in CFD simulations. For example, the integration of time-dependent PDEs modelling CFD problems, by using implicit or semi-implicit methods, leads to linear systems
where $A$ is a real $n \times n$ matrix, usually large and sparse, whose dimension and entries, conditioning, sparsity pattern and coupling among the variables may change during the simulation. Furthermore, because of the high computational requirements of large-scale CFD applications, parallel computers are often used and hence the matrix $A$ is distributed among multiple processors.

Krylov solvers are the methods of choice for such linear systems, but their efficiency and robustness is strongly dependent on the coupling with suitable preconditioners that are able to provide a good approximation of the matrix $A$ at a reasonable computational cost. Unfortunately, among the various available preconditioners, no one can be considered the “absolute winner” and experimentation is generally needed to select the best one for the problem under investigation. Furthermore, developing parallel implementations of preconditioners is not trivial, since the effectiveness and the parallel performance of a preconditioner often do not agree.

Algebraic multilevel preconditioners have received an increasing attention in the last fifteen years, as testified also by the development of software packages based on them [13, 21, 22, 27]. These preconditioners, which approximate the matrix $A$ through a hierarchy of coarse matrices built by using information on $A$, but not on the geometry of the problem originating $A$ (e.g. on the discretization grid of a PDE), are potentially able to automatically adapt to specific requirements of the problem to be solved [31]. Furthermore, they have shown effectiveness in enhancing the convergence and robustness of Krylov solvers in a variety of applications [25, 24].

In this paper we discuss some issues in the design and development of software implementing parallel algebraic multilevel domain decomposition preconditioners based on Schwarz methods. We start from a description of such preconditioners, to identify algorithmic features that are relevant to the development of parallel software (Section 2). Then we present MLD2P4, a package providing parallel algebraic multilevel preconditioners based on Schwarz domain decomposition methods, in the context of the Parallel Sparse BLAS (PSBLAS) computing framework for distributed-memory machines (Section 3). Specifically, we outline the main objectives and the related design choices in the development of this package. Furthermore, we report on the application of different MLD2P4 multilevel preconditioners, coupled with GMRES, to linear systems arising within a Large Eddy Simulation (LES) of incompressible turbulent channel flows, and discuss the results obtained in terms of numerical effectiveness and parallel performance (Sections 4 and 5). We give a few concluding remarks at the end of the paper (Section 6).

2 Algebraic Multilevel Schwarz Preconditioners

Domain decomposition preconditioners are based on the divide and conquer technique; from an algebraic point of view, the matrix to be preconditioned is divided into submatrices, a “local” linear system involving each submatrix is (approximately) solved, and the local solutions are used to build a preconditioner for the whole original matrix. This process often corresponds to dividing a physical domain associated
to the original matrix into subdomains (e.g. in a PDE discretization), to (approximately) solving the subproblems corresponding to the subdomains and to building an approximate solution of the original problem from the local solutions. On parallel computers the number of submatrices usually matches the number of available processors.

Additive Schwarz (AS) preconditioners are domain decomposition preconditioners using overlapping submatrices, i.e. with some common rows, to couple the local information related to the submatrices (see, e.g., [29]). We assume that the matrix $A$ in (1) has a symmetric nonzero pattern, which is not too restrictive if the matrix arises from some PDE discretization. By using the adjacency graph of $A$, we can define the so-called $\delta$-overlap partitions of the set of vertices (i.e. row indices) $W = \{1, 2, \ldots, n\}$ [8]. Each set $W_i^\delta$ of a $\delta$-overlap partition of $W$ identifies a submatrix $A_i^\delta$, corresponding to the rows and columns of $A$ with indices in $W_i^\delta$. Let $R_i^\delta$ be the (restriction) matrix which maps a vector $v$ of length $n$ onto the vector $v_i^\delta$ containing the components of $v$ corresponding to the indices in $W_i^\delta$. The matrix $A_i^\delta$ can be expressed as $A_i^\delta = R_i^\delta A (R_i^\delta)^T$ and the classical AS preconditioner is defined by

$$M_{AS}^{-1} = \sum_{i=1}^{m} (R_i^\delta)^T (A_i^\delta)^{-1} R_i^\delta,$$

where $m$ is the number of sets of the $\delta$-overlap partition and $A_i^\delta$ is assumed to be nonsingular. Its application to a vector $v$ within a Krylov solver requires the following basic operations: restriction of $v$ to the subspaces identified by the $W_i^\delta$’s, i.e. $v_i = R_i^\delta v$; solution of the linear systems $A_i^\delta w_i = v_i$; prolongation and sum of the $w_i$’s, i.e. $w = \sum_{i=1}^{m} (R_i^\delta)^T w_i$. The linear systems at the second step are usually solved approximately, e.g. using incomplete LU (ILU) factorizations. Variants of the classical AS preconditioners exists; the most commonly used one is the Restricted AS (RAS) preconditioner, since it is generally more effective in terms of convergence rate and of parallel performance [9].

From the previous description we see that the AS preconditioners exhibit an intrinsic parallelism, which makes them suitable for a scalable implementation, i.e. such that the time per iteration of the preconditioned solver is kept constant as the problem size and the number of processors are proportionally scaled. On the other hand, the convergence rate of iterative solvers coupled with AS preconditioners deteriorates as the number of sets $W_i^\delta$, and hence of processors, increases [29]. Therefore such preconditioners do not show algorithmic scalability, i.e. the capability of keeping constant the number of iterations to get a specified accuracy, as the number of processors grows.

Optimal Schwarz preconditioners, i.e. such that the number of iterations is bounded independently of the number of the submatrices (and of the size of the grid, when the matrix comes from a PDE discretization) can be obtained by introducing a global coupling among the overlapping partitions, through a coarse-space approximation $A_C$ of the matrix $A$. The two-level Schwarz preconditioners are obtained by combining a basic Schwarz preconditioner with a coarse-level correction based on $A_C$. In this context, the basic preconditioner is called smoother.
In a pure algebraic setting, $A_C$ is usually built with a Galerkin approach. Given a set $W_C$ of coarse vertices, with size $n_C$, and a suitable $n_C \times n$ restriction matrix $R_C$, $A_C$ is defined as $A_C = R_C A R_C^T$ and the coarse-level correction operator to be combined with a generic AS preconditioner $M_{1L}$ is obtained as

$$M_{C}^{-1} = R_C A_C^{-1} R_C,$$

where $A_C$ is assumed to be nonsingular. The application of $M_C^{-1}$ to a vector $v$ corresponds to the restriction $w = R_C v$, to the solution of the linear system $A_C y = w$ and to the prolongation $z = R_C^T y$.

The operators $M_C$ and $M_{1L}$ may be combined in either an additive or a multiplicative framework. In the former case, at each iteration of a Krylov solver, $M_C^{-1}$ and $M_{1L}^{-1}$ are independently applied to the relevant vector $v$ and the results are added. This corresponds to the two-level additive Schwarz preconditioner

$$M_{2L}^{-1} = M_C^{-1} + M_{1L}^{-1}.$$

In the multiplicative case, a possible combination consists in applying first $M_C^{-1}$ and then $M_{1L}^{-1}$, as follows: coarse-level correction of $v$, i.e. $w = M_C^{-1} v$; computation of the residual $y = v - A w$; smoothing of $y$ and update of $w$, i.e. $z = w + M_{1L}^{-1} y$. These steps correspond to the following Schwarz preconditioner, that we refer to as two-level hybrid post-smoothed preconditioner:

$$M_{2LH-post}^{-1} = M_{1L}^{-1} + (I - M_{1L}^{-1} A) M_C^{-1}.$$

Similarly, the smoother may be applied before the coarse-level correction operator (two-level hybrid pre-smoothed preconditioner), or both before and after the correction (two-level hybrid symmetrized preconditioner).

An algebraic approach to the construction of the set of coarse vertices is provided by the smoothed aggregation [4]. The basic idea is to build $W_C$ by suitably grouping the vertices of $W$ into disjoint subsets (aggregates), and to define the coarse-to-fine space transfer operator $R_C^T$ by applying a suitable smoother to a simple piecewise constant prolongation operator. The aggregation algorithms are typically sequential and different parallel versions of them have been developed with the goal of achieving a tradeoff between scalability and effectiveness [32]. The simplest parallel aggregation strategy is the decoupled one, in which every processor independently applies the sequential algorithm to the subset of $W$ assigned to it in the initial data distribution. This version is embarrassingly parallel, but may produce non-uniform aggregates near boundary vertices, i.e. near vertices adjacent to vertices in other processors, and is strongly dependent on the number of processors and on the initial partitioning of the matrix $A$. Nevertheless, the decoupled aggregation has been shown to produce good results in practice [32].

Preconditioners that are optimal in the sense defined above do not necessarily correspond to minimum execution times. For example, when the size of the system to be preconditioned is very large, the use of many processors, i.e. of many small
submatrices, may lead to large coarse-level systems, whose exact solution is generally computationally expensive and deteriorates the implementation scalability of the basic Schwarz preconditioner. A possible remedy is to solve the coarse level-system approximately; this is generally less time expensive, but the correction, and hence the preconditioner, may lose effectiveness. Therefore, it seems natural to use a recursive approach, in which the coarse-level correction is re-applied starting from the current coarse-level system. The corresponding preconditioners, called multilevel preconditioners, can significantly reduce the computational cost of preconditioning with respect to the two-level case. Additive and hybrid multilevel preconditioners are obtained as direct extensions of the two-level counterparts; a detailed description may be found in [29, Chapter 3]. In practice, finding a good combination of the number of levels and of the coarse-level solver is a key point in achieving the effectiveness of a multilevel preconditioner in a parallel computing setting; the choice of these two features is generally dependent on the characteristics of the linear system to be solved and on the characteristics of the parallel computer.

3 The MLD2P4 software package

The MultiLevel Domain Decomposition Parallel Preconditioners Package based on PSBLAS (MLD2P4) [13] implements multilevel Schwarz preconditioners, that can be used with Krylov solvers available in the PSBLAS framework [20] for the solution of system (1). Both additive and hybrid multilevel variants are available; the basic AS preconditioners are obtained by considering just one level. An algebraic approach, based on the decoupled smoothed aggregation technique, is used to generate a sequence of coarse-level corrections to any basic AS preconditioner, as explained in Section 2. Since the choice of the coarse-level solver is important to achieve a trade-off between optimality and efficiency, different coarse-level solvers are provided, i.e. sparse distributed and sequential LU solvers, as well as distributed block-Jacobi ones, with ILU or LU factorizations of the blocks. More details on the various preconditioners implemented in the package can be found in [13].

The package has been written in Fortran 95, to enable immediate interfacing with Fortran application codes, while following a modern object-based approach through the exploitation of features such as abstract data type creation, functional overloading and dynamic memory management. Single and double precision implementations of MLD2P4 have been developed for both real and complex matrices, all usable through a single generic interface.

The main “object” in MLD2P4 is the preconditioner data structure, containing the matrix operators and the parameters defining a multilevel Schwarz preconditioner. According to the object-oriented paradigm, the user does not access this structure directly, but builds, modifies, applies and destroys it through a set of MLD2P4 routines. The preconditioner data structure has been implemented as a Fortran 95 derived data type; it basically consists of an array of base preconditioners, where a base preconditioner is again a derived data type, storing the part of the preconditioner associated to a certain level and the mapping from it to the next coarser level. This
choice enables to reuse, at each level, the same routines for building and applying the preconditioner, and to combine them in various ways, to obtain different preconditioners. Furthermore, starting from a description of the preconditioners in terms of basic sparse linear algebra operators, as outlined in Section 2, the previous routines have been implemented as combinations of building blocks performing basic sparse matrix computations (for more details see [6, 12]). The PSBLAS library, which includes parallel versions of most of the Sparse BLAS computational kernels proposed in [17] and sparse matrix management functionalities, has been used as software layer providing the building blocks. The vast majority of data communication operations required by MLD2P4 have been encapsulated into PSBLAS routines; only very few direct MPI calls occur in the package. The choice of a modular approach, based on the PSBLAS library, has been driven by objectives such as extensibility, portability, sequential and parallel performance.

The modular design has naturally led to a layered software architecture, where three main layers can be identified. The lower layer consists of the PSBLAS kernels. The middle one implements the construction of the preconditioners and their application within a Krylov solver. It includes the functionalities for building and applying various types of basic Schwarz preconditioners, for generating coarse matrices from fine ones and for solving coarse-level linear systems; furthermore, it provides the routines combining these functionalities into multilevel preconditioners. The middle layer includes also interfaces to the third-party software packages UMFPACK [14], SuperLU [15] and SuperLU_DIST [16], performing sequential or distributed sparse LU factorizations and related triangular system solutions, that can be exploited at different levels of the multilevel preconditioners. The upper layer provides a uniform and easy-to-use interface to all the preconditioners implemented in MLD2P4. It consists of few black-box routines suitable for users with different levels of expertise; non-expert users can easily select the default basic and multilevel preconditioners, while expert ones can choose among various preconditioners, by a proper setting of different parameters.

A more detailed description of MLD2P4 can be found in [7, 13]; a deep analysis of the effectiveness and parallel performance of MLD2P4 preconditioners, as well as a comparison with state-of-the-art multilevel preconditioning software, can be found in [7, 12].

4 Using MLD2P4 in the LES of turbulent channel flows

MLD2P4 has been used within a Fortran 90 code performing a LES of turbulent incompressible channel flows, in order to precondition linear systems which are a main computational kernel in an Approximate Projection Method (APM). We briefly describe the numerical procedure implemented in the code, to show how MLD2P4 has been exploited; for more details the user is referred to [1].

Incompressible and homothermal turbulent flows can be modelled as initial boundary-value problems for the Navier-Stokes (N-S) equations. We consider a non-dimensional weak conservation form of these equations, involving the volume