Comput Sci Res Dev (2011) 26: 197–203
DOI 10.1007/s00450-011-0167-z

SPECIAL ISSUE PAPER

Wavelet-based adaptive multi-resolution solver on heterogeneous parallel architecture for computational fluid dynamics

L.H. Han · T. Indinger · X.Y. Hu · N.A. Adams

Published online: 20 April 2011

Abstract For the efficient simulation of fluid flows governed by a wide range of scales a wavelet-based adaptive multi-resolution solver on heterogeneous parallel architectures is proposed for computational fluid dynamics. Both data- and task-based parallelisms are used for multi-core and multi-GPU architectures to optimize the efficiency of a high-order wavelet-based multi-resolution adaptative scheme with a 6th-order adaptive central-upwind weighted essentially non-oscillatory scheme for discretization of the governing equations. A modified grid-block data structure and a new boundary reconstruction method are introduced. A new approach for detecting small scales without using buffer levels is introduced to obtain additional speed-up by minimizing the number of required blocks. Validation simulations are performed for a double-Mach reflection with different refinement criteria. The simulations demonstrate accuracy and computational performance of the solver.

Keywords Adaptive multi-resolution · Multi-core · Multi-GPU · Data-based parallelisms · Task-based parallelisms · Error analysis

L.H. Han · T. Indinger · X.Y. Hu (✉) · N.A. Adams
Institute of Aerodynamics and Fluid Mechanics,
Technische Universität München, 85748 Garching, Germany
e-mail: genre.hu@googlemail.com

L.H. Han
e-mail: luhui.han@aer.mw.tum.de

N.A. Adams
e-mail: nikolaus.adams@aer.mw.tum.de

T. Indinger
FluiDyna GmbH, 85748 Garching, Germany
e-mail: thomas.indinger@aer.mw.tum.de

1 Introduction

The demand for computational fluid dynamics (CFD) simulations of increasingly complex technical problems requires the development of increasingly complex numerical solutions methods with inherent scale adaptivity that perform highly efficient on heterogeneous commodity architectures. One of the difficulties arises from the presence of dynamic small length scales which are essential for the overall flow evolution.

For uniform meshes the smallest flow scale would determine the overall grid size, leading to a waste of computational resources whenever these scales occur only in limited flow regions. As these regions are not known a priori, adaptive simulations, such as adaptive mesh refinement [1] and multi-resolution methods [2, 11], have been introduced for adjusting grid resolution locally and dynamically according to error estimates from different resolution levels. The original multi-resolution schemes do not result in highly parallel algorithms due to their inherently sequential, hierarchical, nested data structure. Recently, Hejazialhosseini et al. [7] presented a new method based on wavelet-adaptive blocks with local time-stepping on multi-core architectures. Instead of using a single grid cell in the dynamic graded tree data structure, they introduced the concept of grid blocks which have a predefined number of computational cells. This change decreases the granularity at the expense of efficiency, but significantly decreases the amount of sequential operations and improves parallelism.

Data- and task-based parallelisms are the two most common classifications. Graphics processing units (GPU) have an inherent advantage of dealing with data parallelism, they execute many concurrent threads rather than sequences of single threads. It has been demonstrated that GPU offer one to two orders of magnitude speedup over the respec-

tive CPU implementations. However, multi-resolution methods do not only contain the solution of the discrete equations for the flow evolution with inherent data parallelism but also compute-intensive logical events such as updating dynamic trees. Intel Threading Building Blocks (TBB) allows for task-based multi-core parallelism which is suitable for such task-based parallelism.

In this work, we present an adaptive multi-resolution solver designed for solving fluid dynamics problems on heterogeneous (multi-core and multi-GPU) parallel architectures. We propose task-based parallelism for handling multi-resolution adaptivity with high-order wavelets, and data-based parallelism for solving the flow transport equations with a 6th-order adaptive central-upwind weighted essentially non-oscillatory (WENO-CU6) scheme [8]. In order to obtain additional speedup, we use overlapping blocks to decrease the complexity of ghost reconstruction. Furthermore, through detecting the propagation of small length scales, we minimize the number of blocks. We also implement a local time stepping technique to achieve extra speedup.

2 Governing equations

The fluid dynamics considered in this work are described by the two-dimensional compressible Euler equations

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}_y = 0, \tag{1}$$

which describe the conservation laws expressed in conservative variables: mass density ρ, momentum density $\rho \mathbf{v} \equiv (\rho u, \rho v)$ and total energy density

$$E = \rho e + \frac{1}{2}\rho u^2, \tag{2}$$

where e is the internal energy per unit mass. To close this set of equations, the ideal-gas equation of state

$$p = (\gamma - 1)\rho e, \tag{3}$$

where $\gamma = 1.4$ is the ratio of specific heat, is used.

3 Numerical method on uniform grid

For simplicity we explain the numerical discretization of (1) by considering a one-dimensional generic conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0, \tag{4}$$

by the local linearized characteristic decomposition technique. With the method of lines, (4) is approximated as

$$\frac{du_i}{dt} \approx -\frac{1}{\Delta x_i}(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}), \tag{5}$$

where Δx_i is the grid size or size of the computational cell i, $\hat{f}_{i\pm1/2}$ are numerical fluxes at cell faces. According to the Lax-Friedrichs flux splitting technique it is

$$\hat{f}_{i+1/2} = \frac{1}{2}\left(\hat{f}^+_{i+1/2} + \hat{f}^-_{i+1/2}\right), \tag{6}$$

where the positive and negative fluxes $\hat{f}^+_{i+1/2}$ and $\hat{f}^-_{i+1/2}$ are obtained by WENO-CU6 reconstruction. The WENO-CU6 reconstruction is applied to the fluxes projected onto characteristic directions by the Roe matrix [9]. After reconstruction the numerical fluxes are projected back. In this work, a 2nd-order TVD Runge-Kutta scheme is used for time integration [12].

3.1 WENO-CU6 scheme

The positive flux $\hat{f}^+_{i+1/2}$ is obtained by

$$\hat{f}^+_{i+1/2} = \sum_{r=0}^{3} \omega^+_r \hat{f}^{k,r}_{i+1/2}. \tag{7}$$

$\hat{f}^{k,r}_{i+1/2}$ is the cell-face value of a local polynomial $\hat{f}^{k,r}_i(x)$ obtained from a k-th order reconstruction with neighboring cell averages $\overline{f}_{i-k+r+1}, \ldots, \overline{f}_i, \ldots, \overline{f}_{i+r}$, where

$$\overline{f} = f(\overline{u}) + a_{max}\overline{u}, \tag{8}$$

and a_{max} is the maximum sound speed. The WENO weights ω^+_r are given by

$$\omega^+_r = \frac{\alpha_r}{\sum_{r=0}^{3}\alpha_r}, \qquad \alpha_r = d_r\left(C + \frac{\tau^6}{\beta_r^3 + \epsilon}\right), \tag{9}$$

where d_r are the linear weights which combining $\hat{f}^{3,r}_i(x)$, $r = 0, 1, 2, 3$ to achieve the numerical flux of $\hat{f}^{6,3}_i(x)$, which is a 6th order, non-dissipative, symmetric reconstruction. The smoothness indicators β^k_r is given by

$$\beta^k_r = \sum_{n=1}^{k-1} \Delta x^{2n-1} \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} \left(\frac{d^n}{dx}\hat{f}^{k,r}_i(x)\right)^2 dx \tag{10}$$

$C \gg 1$ is a positive parameter, ϵ is a small positive number, $\beta^3_3 = \beta^6_3$ and τ^6 is a linear combination of β^6_3 and β^3_r, $r = 0, 1, 2$. The negative flux $\hat{f}^-_{i+1/2}$ is calculated in the same fashion except for flipping the index of the cell averages to $\overline{f}_{i+r}, \ldots, \overline{f}_{i+1}, \ldots, \overline{f}_{i-k+r+1}$ and using

$$\overline{f} = f(\overline{u}) - a_{max}\overline{u}. \tag{11}$$

The WENO-CU6 has formally 6th-order accuracy in smooth regions of the solution, and achieves very small numerical dissipation while preserving the good shock-capturing properties of the classical WENO scheme.

4 Adaptive multi-resolution solver

The parallel algorithm is designed for heterogeneous parallel computing architectures (including multi-core and multi-GPU). The algorithm is divided into two separate parts, (i) adaptive multi-resolution, (ii) discrete flow transport equations. Part (i) exhibits task parallelism and is mainly handled by a multi-core architecture (MC). Part (ii) exhibits data parallelism and is mainly handled by multi-GPU architecture (GPU). Necessary communication between MC and GPU involves only the exchange of conservative flow variables. All the intermediate data which are needed for (ii) are created calculated directly in GPUs.

4.1 Multi-resolution adaptivity

The data on different resolution levels are structured by a quad-tree, Fig. 1, whose nodes are grid blocks with cell-averaged values of the conservative variables. By using bi-orthogonal wavelets [3], a multi-resolution analysis of the flow field can be performed. It includes the projection operator and its inverse, the prediction operator, corresponding to the analysis function and the synthesis function, respectively, in the fast wavelet transform. With the projection operator the cell-averages at a resolution level can be estimated from its child-level exactly and uniquely. However, the prediction operator can only give an approximation of cell-averages at the child level by interpolation because the detail coefficients of the child level are not involved. Therefore, the multi-resolution adaptivity is based on a compressed representation of the original field. When the exact values on the fine grid have negligible difference from their approximation interpolated from the corresponding coarser grid, the flow field can be represented by the values on the coarser grid.

4.2 Grid blocks

A grid block consists of a predefined number of cells in each dimension. All blocks contain the same number of cells. In physical space, the blocks have variable sizes and therefore different resolutions. Since both, WENO-CU6 stencil and the prediction stencil of the adaptive multi-resolution, use ghost cells, as shown in Fig. 1, the blocks mentioned above are padded to contain their full (inner and physical) boundaries with a predefined width. Note that this type of blocks is similar to that in the adaptive mesh refinement method [10], the difference is that the present blocks have uniform predefined number of cells.

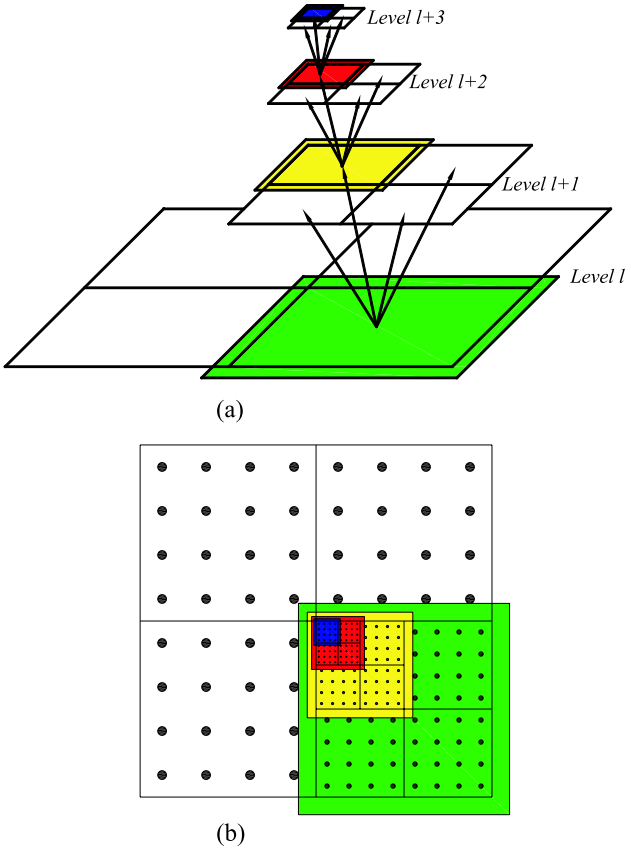

Fig. 1 Hierarchy of blocks with different level of resolutions: (**a**) *prospective view*, (**b**) *top view*. The overlap regions are the boundaries

4.3 Block-boundary reconstruction

For parallel programming the most complicated work is the communication between tasks or data on different threads. Since every block has its own boundaries each of them can be viewed as a subfield with uniform grid. The only difference between these subfields and the original flow field is that the boundary conditions of subfields not only come from the physical boundaries of the full domain but also from other neighboring blocks (inner boundaries). The inner boundary condition of each block encounters no more than two different situations. As shown in Fig. 2, the block a at level $l+1$ has three brothers b, c and d but no cousins in the lower-left direction. In the other directions, blocks at the same resolution level do not exist.

A direct way to reconstruct a boundary from brothers is to look up all neighbors including the diagonal directions. This is very complex especially for high spatial dimensions. As less complex alternative we introduce a two-step exchange method. At step 1 every block reads data from its neighbors, see e.g. Fig. 3 where the data in the green regions belonging to block b and c are copied to block a. At the same time the values in blue regions of block d are moved into block

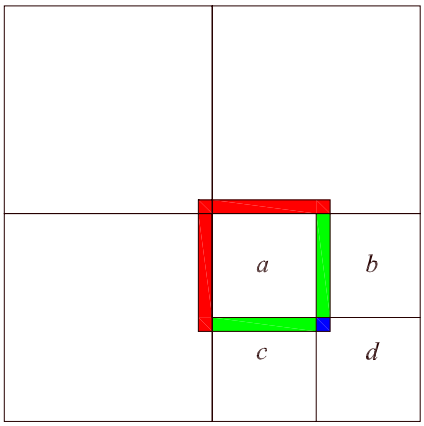

Fig. 2 A grid block a at level $l+1$ has the three uncles, two adjacent brothers b and c, and a diagonal brother d

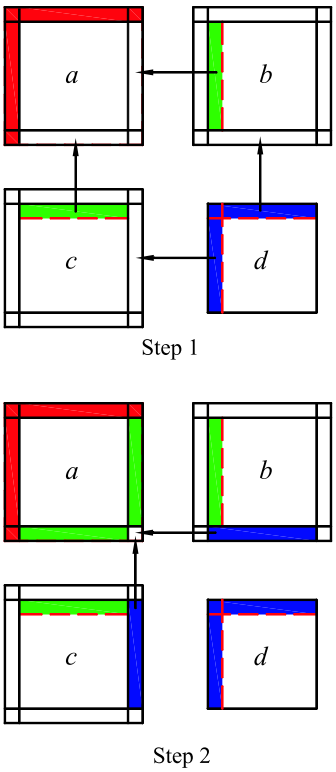

Fig. 3 The two-step boundary reconstruction from brothers

b and c separately. At step 2, each block can find its diagonal boundaries from its neighbors. Note that this method extends in a straightforward way to three dimensions.

The boundary reconstruction from parent and uncles is more complex. If the width of the inner boundaries for every block is set to w cells, as shown in Fig. 4, the accuracy order of the wavelet-based multi-resolution method is $w+1$, see [2]. With this constraint the red regions, as shown in Fig. 2, can be reconstructed easily by interpolating from its parent and uncles with prediction operations. In this work, it is set $w=4$ to obtain 5th-order averaging/interpolating

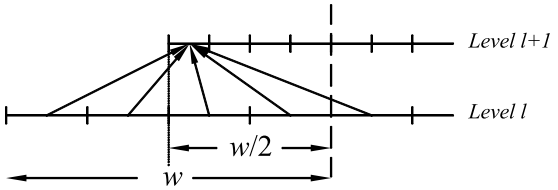

Fig. 4 Reconstruction of a farthest boundary cells of a level $l+1$ grid block

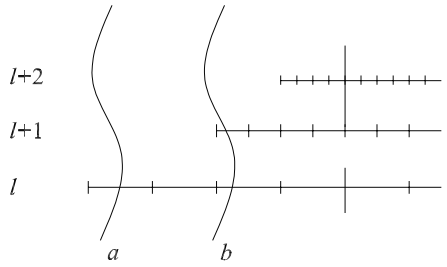

Fig. 5 A discontinuity moves to grid blocks and leads to successive refinements

wavelets. This width is also sufficient for the WENO-CU6 scheme as it needs a three cell boundary only.

4.4 Tree updating

A fully adaptive scheme contains three steps: refinement, computation, and thresholding. The dynamic property of the adaptive tree structure is mainly present during the refinement and thresholding procedures. When the details are smaller than a prescribed relative error tolerance or thresholding coefficient ϵ, the thresholding operator removes the child level. In Hejazialhosseini et al. [7] the refinement operation adds one more level as buffer zone after thresholding to avoid small scales passing through grid blocks with a jump in resolution within a single time step. However, in most situations small scales move only a small distance during a time step. The above refinement operation, which adds an extra buffer level, obviously is not very efficient and may not be necessary. Since every grid block has its own boundary, naturally this boundary can be used as buffer region too. As shown in Fig. 5, suppose a small-scale structure that needs the finest level $l+2$ for being resolved is moving to the right. When it reaches position a that is within the padded boundary of a grid block at level l, it will be detected by the block which consequentially is refined to level $l+1$. When the discontinuity moves further to position b which is within the padded boundary of a grid block at level $l+1$, this block will be refined to level $l+2$. At this time, since the distance between the discontinuity and the interior cells of the finest block is about eight fine-resolution padded boundary cells, there is a sufficient buffer region. This algorithm avoids unnecessary buffer blocks and decreases the number of grid blocks to a minimum, resulting in an improved efficiency.

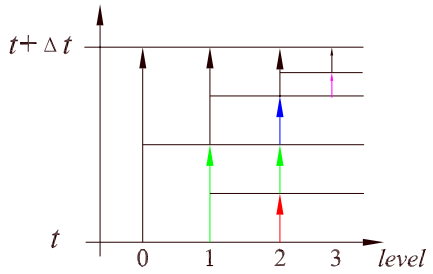

Fig. 6 (Color online) A typical cycle of time integration. The sequences of advance in time are *red* (level 2), *green* (levels 1 and 2), *blue* (level 2), *magenta* (level 3) and *black* (levels 0 to 3)

5 Local time stepping

Since small spatial scales are often associated with small time scales a local time-stepping scheme is implemented [5]. The main principle of local time-stepping is that the finer blocks and their boundaries are updated more frequently than the coarser blocks. Assume that δt is the time step for cells at the finest level. Within one cycle time is advanced as $\Delta t = 2^{L_{max}}\delta t$, where L_{max} is index of the finest resolution level. As shown in Fig. 6, assume that the finest block has resolution level $l = 3$, and that at the begin of the cycle there are only blocks at resolution levels $l = 0, 1, 2$. At the time $t + 6\delta t$, the refinement operator creates the finest block of resolution level $l = 3$. Accordingly, the full cycle is complete after five steps, as represented by the right-most arrows for each resolution level in Fig. 6. Note that blocks at different resolution levels can evolve concurrently. When two successive levels reach the same discrete time, the finer-level values are projected onto the coarser level because the solution on the finer level is more accurate. Since the projection leads can violate discrete conservation on the coarser level a flux correction step is applied to the numerical fluxes at the coarse-fine resolution-level boundaries so that discrete conservation across the grid-block hierarchy is ensured. Note that discrete conservation is essential for the physically consistent transport of shocks and interfaces.

6 Results

For demonstration of our method we consider an important two-dimensional flow configuration proposed by Colella and Woodward [4], i.e. the double Mach reflection of a strong shock. A Mach 10 shock in air is reflected from a wall with incidence-angle of 60^0. The initial condition is

$$(\rho, u, v, p) = \begin{cases} (1.4, 0, 0, 1), & \text{after shock,} \\ (8, 7.145, -4.125, 116.8333), & \text{else,} \end{cases}$$

and the final time is $t = 0.2$. The computational domain for this problem is $[0, 0] \times [4, 1]$. Initially, the shock extends from the point $x = 0.1667$ at the bottom to the top of the computational domain along the line $y < 1.732(x - 0.1667)$. Along the bottom boundary at $y = 0$ the region from $x = 0$ to $x = 0.1667$ is always assigned with the post-shock conditions, whereas a reflecting wall condition is set from $x = 0.1667$ to $x = 4$. Inflow and outflow boundary conditions are applied at the left and right ends of the domain, respectively. The values at the top boundary are set to describe the exact motion of a Mach 10 shock. In the simulations the number of grid points in each block is set to 40 per direction and the maximum resolution level is set to 5. This setup can be compared to an effective grid resolution of 2048×512 points. Time step is chosen as $CFL = 0.6$. The influence of the threshold coefficient ϵ is studied by choosing four different values $\epsilon = 0.02$, 0.05, 0.1 and 0.3 The simulations are carried on a workstation with 4 quad-core Intel® Xeon® Processor E5620 processors (12M Cache, 2.40 GHz) with 24 GB of RAM, and 2 NVIDIA® Tesla™ C2050 Computing Processors (with 3 GB GDDR5 Memory).

Figure 7 shows the density contours and grid-block distribution at $t = 0$ and 0.2 when the used threshold coefficient is 0.05. The results are in good agreement with those on a uniform grid of 1921×481 of Gerolymos et al. [6] computed with a 17th-order WENO scheme. As shown in a close-up view of the "blow-up" region (Fig. 8), the present simulation resolves much better the wave structures near the second triple point and predicts a stronger jet near the wall, and also resolves considerably finer vortical structures and small scales.

The total number of blocks for different threshold coefficients are is in Table 1. Note that increasing small threshold coefficients can decrease the number of blocks considerably. However, the number of blocks is not sensitive to varying larger threshold coefficients. The block distributions for threshold coefficients from 0.05 to 0.3 have very little differences. So, for this case an optimum coefficient 0.05 can be obtained. However, whether this value generally holds for other cases requires further investigation.

7 Concluding remarks and outlooks

We have presented a wavelet-based multi-resolution solver designed for solving complex fluid dynamics problems on heterogeneous (multi-core and multi-GPU) parallel architecture. We propose both task-based and data-based parallelisms for handling adaptive multi-resolution and solution of the discrete flow transport equations. As an improvement to previous multi-resolution methods we use overlapping blocks to decrease the complexity of ghost reconstruction, and are able to decrease the number of blocks to a minimum through detecting the propagation of small length scales. We

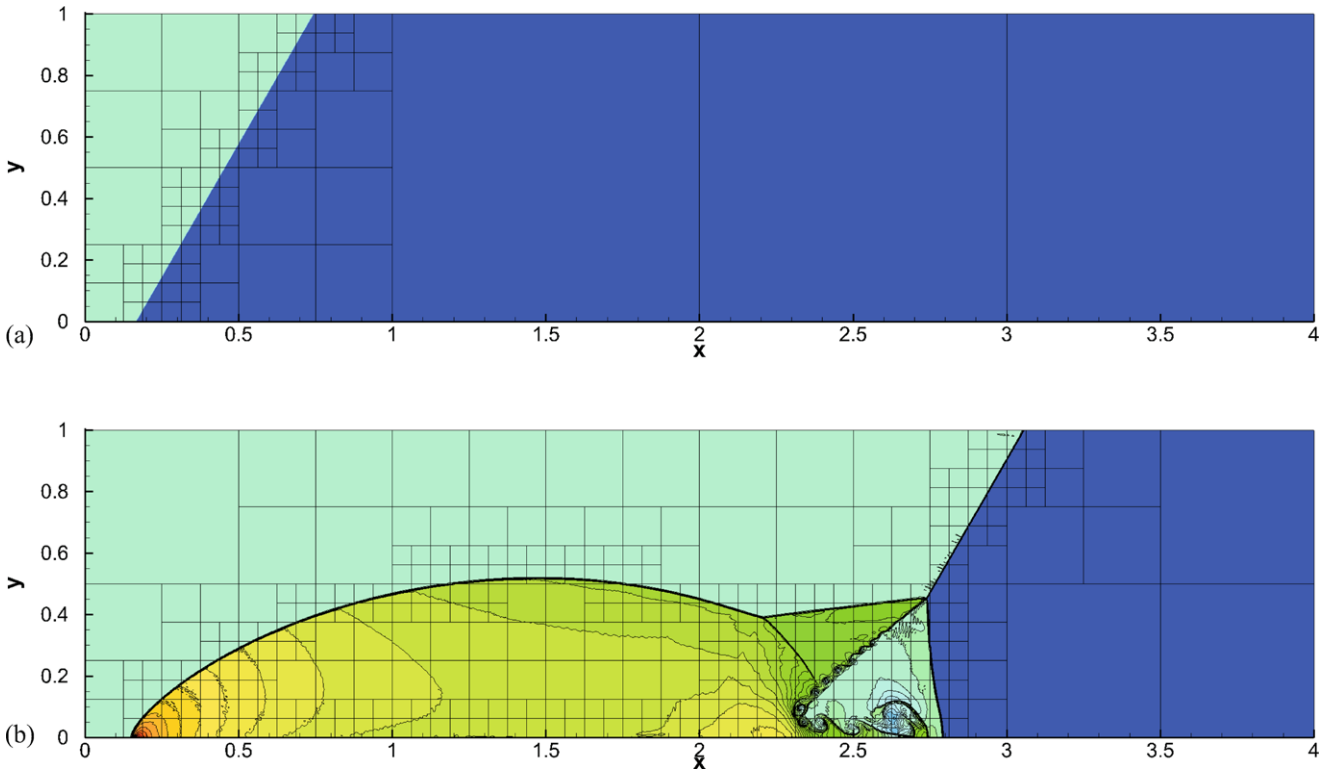

Fig. 7 Double-Mach reflection of a Mach 10 shock wave: block distribution and density profile at $t = 1$ and 0.2 on an effective grid of 2048×512 points

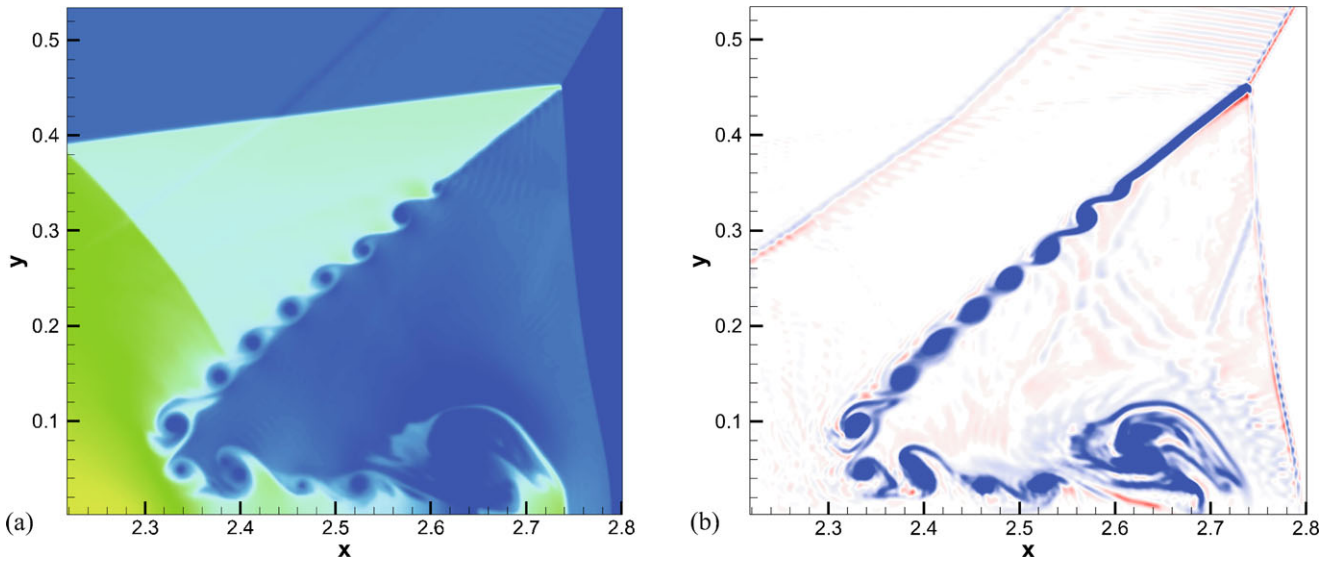

Fig. 8 Close-up view of the "blow-up" region: density (*right*, *blue*/*red*: positive/negative vorticity) and vorticity (*left*, *blue*/*green*: low/high density) profile

also implement a local-time-stepping technique to achieve extra speedup. Although this technique needs frequent data transferring between host and GPU devices, the majority of the communication time is hidden due to its overlap with computation.

According to Table 1 the simulation using two GPU obtains an about 32 times speedup. Our previous simulations based on one CPU without parallelism showed that the adaptive solver with local time-stepping could achieve about 25 times speedup. Totally we could arrive at a speed-up of 800

Table 1 Threshold coefficient ϵ, computational time and number of blocks at the last time step (N)

ϵ	Single CPU (min)	Two GPUs (min)	N
0.02	–	4.490	415
0.05	118.533	3.663	367
0.1	–	3.573	349
0.3	–	3.423	334

times. Also note that the compression rate in our simulations is about 65.8%. For other cases, especially for 3D problems, such as shock-bubble interaction [7], the compression rate can be as high as 85%. Therefore, the overall speed-up can reach up to 1000 times.

References

1. Berger MJ, Oliger J (1984) Adaptive mesh refinement for hyperbolic partial differential equations. J Comput Phys 53(3):484–512
2. Bihari BL, Harten A (1995) Application of generalized wavelets: an adaptive multiresolution scheme. J Comput Appl Math 61(3):275–321
3. Cohen A, Daubechies I, Feauveau JC (1992) Biorthogonal bases of compactly supported wavelets. Commun Pure Appl Math 45(5):485–560
4. Colella P, Woodward PR (1984) The piecewise parabolic method (PPM) for gas-dynamical simulations. J Comput Phys 54(1):174–201
5. Domingues MO, Gomes SM, Roussel O, Schneider K (2008) An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. J Comput Phys 227(8):3758–3780
6. Gerolymos GA, Sénéchal D, Vallet I (2009) Very-high-order WENO schemes. J Comput Phys 228(23):8481–8524
7. Hejazialhosseini B, Rossinelli D, Bergdorf M, Koumoutsakos P (2010) High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions. J Comput Phys 229(22):8364–8383
8. Hu XY, Wang Q, Adams NA (2010) An adaptive central-upwind weighted essentially non-oscillatory scheme. J Comput Phys 229(23):8952–8965
9. Jiang GS, Shu CW (1996) Efficient implementation of weighted ENO schemes. J Comput Phys 126:202–228
10. Pantano C, Deiterding R, Hill DJ, Pullin DI (2007) A low numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows. J Comput Phys 221(1):63–87
11. Schneider K, Vasilyev OV (2010) Wavelet methods in computational fluid dynamics. Annu Rev Fluid Mech 42:473–503
12. Shu CW, Osher S (1989) Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. J Comput Phys 83(1):32–78

L.H. Han raised in China, is a Ph.D. student in the Institute of Aerodynamics and Fluid Dynamics, Technische Universität München. He graduated in 2009 from Harbin Institute of Technology, earning a Master degree of Engineering in the field of Mechatronics. During these two years' study, his main research work is concentrated on the numerical simulation of Robot fish, which greatly enhanced his interest in numerical simulation, especially in CFD. After that, he went to Germany and has been a research assistant in the institute of aerodynamics. Now he specializes primarily in parallel algorithm and multi-resolution method for Multi-Phase Micro-Fluidic Systems.

T. Indinger is Head of Automotive Aerodynamics at the Institute of Aerodynamics and Fluid Dynamics, Technische Universität München. Additionally he is Founder and General Manager of FluiDyna GmbH which is a leading company in the development of GPGPU accelerated CFD software.

X.Y. Hu received his Ph.D. from Beijing Institute of Technology on Explosion Dynamics. From 2005, he is the Head of Complex Fluids Group at the Institute of Aerodynamics and Fluid Dynamics, Technische Universität München. His research interests are multi-scale, multi-physics modeling of fluid mechanics with finite volume and particle methods.

N.A. Adams received his Ph.D. from Technische Universität München. From April 2002, he was professor in Institute of Fluid Mechanics, Technische Universität Dresden. In 2004, he was appointed professor and director of the Institute of Aerodynamics and Fluid Dynamics, Technische Universität München. In September 2000, his was rewarded Zienkiewicz Award to Young Scientists in Computational Engineering Sciences. His current research interests are: modelling of transitional and turbulent flows, unsteady aerodynamics und flow-structure interaction, micro-fluidics and multiphase flows, numerical methods.