Simulations of Three-Dimensional Detonations by the CE/SE Method Using a Very Low-Cost Beowulf Cluster

Hao He[1], Zeng-Chan Zhang[2], S.T. John Yu[3]

Wayne State University
Detroit, MI 48202

Sin-Chung Chang[4]

NASA Glenn Research Center
Cleveland, OH 44135

ABSTRACT

In this paper, we report the experience of calculating three-dimensional detonations by the Space-Time Conservation Element and Solution Element (CE/SE) Method using a cost-effective Beowulf cluster. The Time-Surfer, a 16-processor PC cluster, was built and tested as part of this effort. The total cost was about twenty thousand 1999-dollars. The efficiency and accuracy of the CE/SE method on the Beowulf cluster are assessed to determine feasibility of applying the CE/SE method to CFD calculations of using five million mesh nodes and above on this low-cost platform. The numerical solutions of three-dimensional detonations propagating in a square tube are calculated. The diagonal instability pattern of three-dimensional detonations is reported.

1. INTRODUCTION

This paper reports our experience in using a high performance but affordable parallel computer system and parallel programming to calculate three-dimensional traveling detonation waves. The numerical method employed is the CE/SE method [4], a novel approach based on treating space and time as one entity in enforcing flux conservation. For three-dimensional detonations, the demand for the computer resource for large memory and reasonable turn-around time is more than a regular sequential machine can provide. To this end, one can employ powerful super computers, which of course are very expensive for the small research groups and consulting corporations. Fortunately, due to the support of high-speed networking in the Linux operating system, we can construct high-performance system using off-the-shelf personal computer (PC) components. Beowulf, which becomes increasingly popular, is an implementation of such computer systems.

There isn't a specific computer or a software package called “Beowulf.” A Beowulf system is a cluster of PCs interconnected by LAN technology. These PCs are running an open source UNIX-like operating system, such as Linux or free BSD, and executing parallel applications, programmed with an industry standard message passing model and library, such as MPI.

1.1 Hardware

A regular PC costs about two thousand dollars or less. Compared with the commercially available super computers, this is about one percent or less. With continuously improvement and commodity-like components, a typical PC nowadays has very respectable computational power. For example, more then 1G Hz processors made by Intel and AMD are quite affordable and have become mainstream products in the market. These CPUs have many new features, including multiple branch prediction, data flow analysis, speculative execution, and dual independent bus. With the help of fast system bus (up to 150M Hz now) and new Rambus and DDR-SDRAM memory, PCs using these CPUs could easily outperform more expensive UNIX workstations in near future. Riding on the fast development of the commodity computer components, Beowulf cluster

[1] Ph.D. Student, AIAA Student Member, Email: haohe@me1.eng.wayne.edu
[2] Senior Research Associate, AIAA Member, zczhang@me1.eng.wayne.edu
[3] Associate Professor, AIAA Member, Email: styu@eng.wayne.edu ; http://141.217.13.61/
[4] Senior Scientist, AIAA Member, Email: Sin-Chung.Chang@grc.nasa.gov

using such PCs could become the mainstream platforms for CFD applications.

Memory is very cost effective nowadays, and we have installed 512K SDRAM in each PC in our cluster and the total memory is four giga bytes, which is large enough to simulate big CFD problems with millions of mesh nodes. In particular, we are very interested in routine computations using more than five millions mesh nodes.

1.2 Software

In parallel computation, the workload is partitioned into smaller pieces, which are taken care by a group of processors. To solve CFD problems by parallel computing, a proper distribution of the tasks is required. One should partition the work evenly to the available computational elements (nodes). It should be done such that no node is overloaded while other node is starving for work. The communication between nodes, which usually is the source of computational overhead, must be minimized. Minimizing communication between nodes is the key for linear speedup. For CFD, evenly balanced workload and minimization of communication are dictated by the procedure of mesh partition. We found that the use of METIS to partition a unstructured mesh works well for us.

To realize parallel programming, a parallel library such as MPI is needed to provide the communication among the computer nodes in the network. Effective programs also need suitable CFD algorithm for parallel computation. The adopted CFD scheme is the Space-Time CE/SE method, which was originally developed by Chang and coworkers at NASA Glenn. In this paper, an extended version of the CE/SE method developed by Zhang et al. is adopted. The detail of the CE/SE method is well documented in the cited references and for the sake of conciseness, no illustration will be provided here.

Linux is popular due to its stability and powerful utility. Moreover, it is designed to get best performance in running program, which is what we need for number crunching. Essentially Linux is a UNIX system, and all operations in UNIX can be implemented by using Linux, including multi-thread, semaphores, shared memory, and RPC for advanced performance and utilities. In numerical simulations, our main concern is the performance of high-level languages such as C, C++, and Fortran. There are many software can be used in Linux environment including free Fortran and C compilers and Massage-Passing library such as MPI. High performance commercial software for the same purpose is also available.

2. SYSTEM CONSTRUCTION

2.1 Cluster Configuration

Our Beowulf cluster, named ***Time-Surfer*** is illustrated in the following diagram:

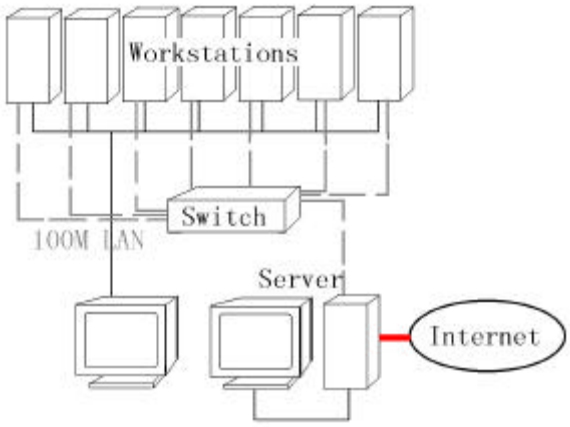

Fig. 1 A schematic of ***Time Surfer*** Beowulf cluster at Wayne State.

Fig. 2 Wayne State ***Time-Surfer***

As shown in Figs. 1 and 2, our Beowulf system consists of eight computers, each of them has dual Intel Pentium III 450 CPU, 512M memory and 3Com 3C905B fast Ethernet card. All the nodes are connected to an Intel Express 510T switch. In building our Beowulf cluster, we use multi-processor PC's

instead of traditional single processor PC. This arrangement, supported by UNIX, Linux and Windows NT, supplies more power in computation. For cost-effectiveness, we have adopted the dual processor PCs in our cluster.

Software environment includes SuSE 6.1 Linux operating system and MPICH 1.2 parallel library. The total cost of our system in 1999 was about 20,000 dollars.

Table 1 Approximate Cost of a 16-node PC Cluster.

Type	**Model**	**Price**	**Qtt.**	**Sub Total**
CPU	Intel Pentium III 800EB/ AMD Thunder Bird 900MHz	200/150	1*16	3200 / 2400
Memory	384M / 768M PC133 SDRAM	200/450	1*16	3200/ 7200
NIC*	3Com 905B-TX-NM	70	(1-2) *16	1120/ 2240
Mother-board	Asus CUSL2 / Asus A7v	140	1*16	2240
Hard Drive	IBM DeskStar 30G 7200RPM	150	1*16	2400
Other Parts		<200	1*16	3200
Monitor	17” CRT	200	1-2	200/ 400
Switch	100BTX 24 –Port	1700	1-2	1700/ 3400
Accesso-ries	Cable, Data Switch Box, etc.	<500	1	500
Software	Free			
Total	16960~24780			

* Network Interface Card

Table 1 provides approximate costs of a similar 16-single-node PC cluster with updated components at the end of 2000. Based on about the same price, one can construct a much faster cluster now. Note that the price list in Table 1 is based on 16 single-CPU boxes instead of 8 dual-CPU boxes as shown in Figs. 1 and 2.

In general, constructing a cluster based on dual-processor PC brings saving and some performance benefit. However, due to continuous upgrading of bus frequency, most old chipsets for dual-processor-motherboard are discontinued. Currently, only a few manufacturers provide new-generation products, e.g., CUR-DLS from ASUS Inc. Due to less competition, prices of dual-processor motherboard is more expensive than two single-processor ones. As always, more products will be available and the price will drop down. Websites such as http://www.pricewatch.com is helpful to assess the cost of building a cluster.

In order to coordinate the computational tasks among all computer nodes, it is necessary to exchange a large amount of information during the computing process. High-speed communication among them is imperative. Currently we are using 100M LAN Fast Ethernet. Giga-bit switch and the associated Ethernet product are also available. However, the cost is considerably higher. With the help of the channel-bonding patch to the Linux kernel, we plan to try 'bond' multiple Ethernet interfaces into a faster 'virtual' Ethernet interface to get higher data transmission speed.

2.2 Parallel Programming

To realize parallel computing, we must split computational tasks into parallel tasks by using message passing, network sockets, and/or Inter-Procedure Call (IPC). Usually, message passing is adopted in Beowulf systems. Software systems such as MPI and PVM allow us to write message-passing parallel programs that run on a cluster. The numerical crunching codes are usually written in Fortran, C and C++.

Parallel Virtual Machine (PVM) used to be message-passing standard until Message Passing Interface (MPI) appeared. PVM is still being widely used. MPI is a library of functions and macros that can be used in the CFD programs for communication. MPI is the de facto standard for portable parallel programs. The MPI is standardized by the MPI Forum and is available on all massively parallel supercomputers.

Two MPI versions are available on the web for free download: (1) MPICH by the Argonne National Laboratory http://www-unix.mcs.anl.gov/mpi/mpich/index.html and (2) Local Area Multi-computer (LAM) MPI developed at the Ohio Supercomputer Center and University of Notre Dame http://www.mpi.nd.edu/lam/. A version of PVM can also be obtained from http://www.epm.ornl.gov/pvm.

In contrast to sequential program, a parallel program needs to be well organized in task load and communication to maximize the performance. Message passing methods provide in MPICH provides us the organized structure for parallel computing. If not done carefully, the executing processors may spend much time waiting for the information communication than actual computing, and thus the overall performance of the cluster will be unacceptable. A basic solution is to exploit the so-called "non-blocking communication," which allows massage passing simultaneously with computing.

In general, the bottleneck of parallel computing is the information communication. A well-designed network with complicated interconnection topologies compared with simple Ethernet may be helpful in efficient communication. Moreover, virtual topology technology, a mechanism for naming the processes that fits the communication pattern, may be helpful to specific problem. In the final paper, we shall report our experience in these futuristic developments.

2.3 Domain Decomposition

Effective use of the Beowulf system requires a proper distribution of simulation tasks among the available processing nodes. A common approach in CFD is to decompose the computational domain into a number of partitions and assign the partitions to different nodes. The processing nodes execute the same CFD solver but on different sub-domains. At the end of each numerical iteration, CPU processors exchange intermediate results at sub-domain boundaries. The objective of domain decomposition is to balance the computational workload and memory occupancy of processing nodes while keeping the inter-node communication as less as possible.

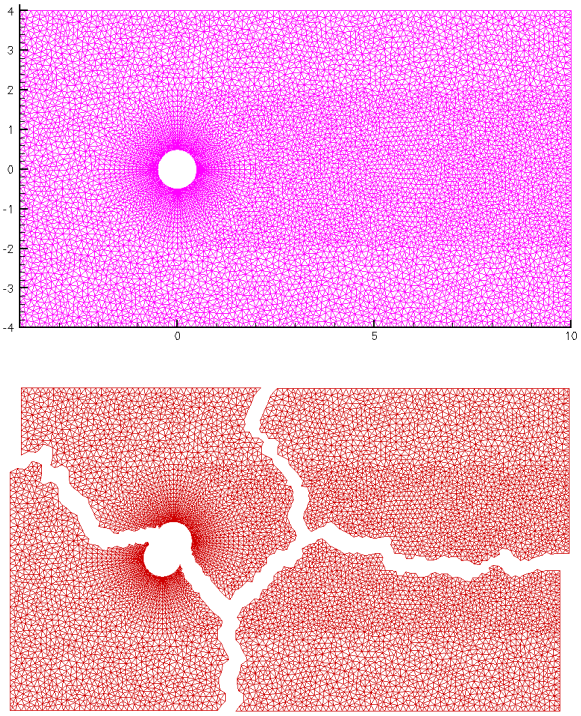

Fig. 3 Domain decomposition by using METIS.

For regular structured mesh, the domain decomposition is straightforward. For unstructured mesh, a robust and effective algorithm to automatically divide the domain is a necessity. In the present work, we use the software package, METIS, to handle the task. METIS provides a fast and high-quality multilevel scheme for partitioning irregular domain. Figure 3 shows the use of METIS to

decompose a two-dimensional unstructured mesh for calculating flows over a circular cylinder. There are total 8454 nodes and 16458 triangle cells. The domain is divided into four parts with balance factor of 1.02. METIS is available for free download on the web page at http://www.cs.umn.edu/~metis.

3. RESULTS AND DISCUSSIONS

We apply our Beowulf system to simulate three-dimensional detonations, in which a steadily propagating detonation wave with complex traversing waves and cellular structure is simulated. Figure 4 shows the computational domain and a simple partition of the domain for parallel computations, in which the domain is divided along the *z* direction.

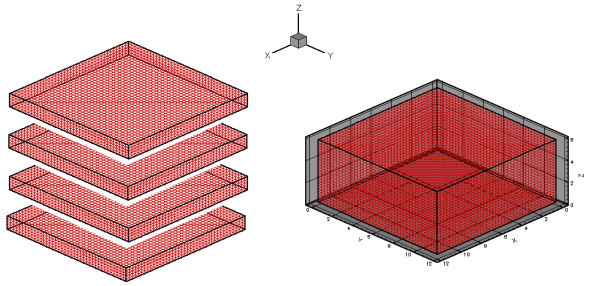

Fig. 4: The computational domain for the three-dimensional detonations.

To model detonations, we solve three-dimensional Euler equations coupled with a species equation. Detailed of the model equations can be found in our previous work [6] or another our paper in the same conference: AIAA 2001-0476. The flow parameters in the present calculations are the heat release q_0 = 50, the activation energy E^+ = 50, the specific heat ratio g = 1.2, and the overdriven factor f=1.6. According to the classical theories, these parameters would trigger flow instability.

Figure 5 shows pressure contours of a three dimensional detonation in a square tube at different times. The unburned reactant flows downward to be consumed by the flame front. We took notice of the peak pressure in three-dimensional detonations is much higher than that in two-dimensional detonations under similar flow conditions. According to our result, the difference is about 380 versus 220 in the dimensionless unit. Figure 6 shows a snapshot of temperature contours of the same flow. The computation is initiated by imposing a ZND classical solution in the *z* direction.

Depending on the imposed perturbations to the ZND solution at the initial stage of the computation, three-dimensional structure of unstable detonations can be classified into two fundamental types: a rectangular structure and a diagonal structure. The rectangular structure consists of two two-dimensional waves. The two-dimensional waves are orthogonal to each other and they travel independently. Therefore, the rectangular structures are not fundamentally three-dimensional but may be described as two superimposed two-dimensional detonations. Figures 5 and 6 are results of the rectangular instability structure. The structure can be clearly seen in Fig. 7, in which snapshots of two-dimensional pressure contours near the flame front are plotted.

To the best of our knowledge, simulated diagonal structure in three-dimensional detonations has not yet been reported. However, such structure is the essential mechanism in actual three-dimensional detonations with triple points moving along the diagonal lines of the cross section of the square tube. The axes of the transverse waves are canted at 45° to the wall. Figure 8 shows snapshots of two-dimensional pressure contours near the flame front. Alternating diagonal waves can be clearly seen.

Table 2 shows the CPU time used for 20 iterations by the CE/SE code for calculating the detonations when using 1, 2, 4, 8, and 16 processor nodes. A corresponding speedup is shown in Fig. 9.

Table 2 Parallel Computation Performance

Processes	Time for 20 iterations (s)	Speedup
1	143	-
2	86	1.66
4	50	2.86
8	28	5.10
16	24	5.95

To improve the speedup, we could change the configuration of the domain decomposition to lighten the communication, or to increase network bandwidth by adding multiple network interfaces to every node. We also did a test about communication mode. Table 3 shows that when using more nodes, blocking

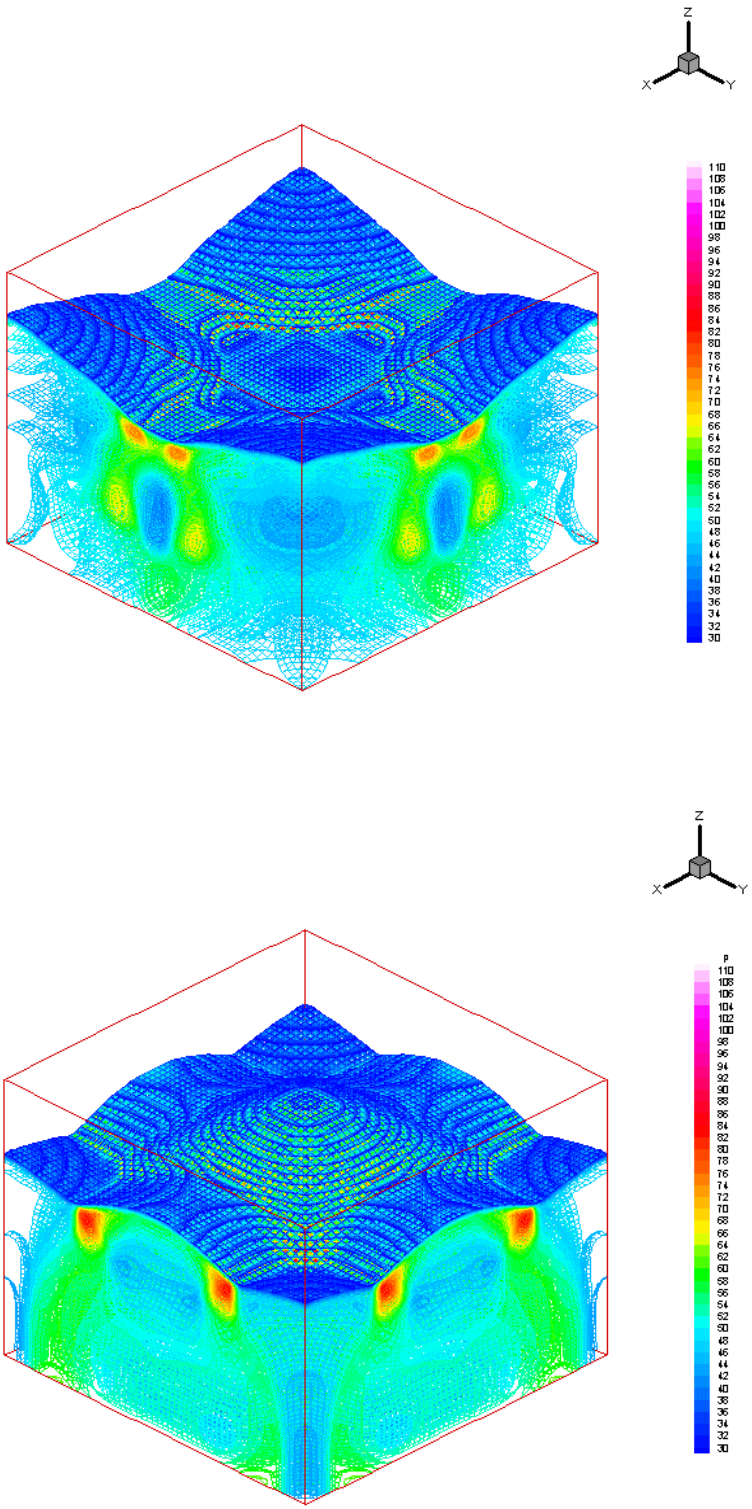

Fig. 5 The pressure contours of a three-dimensional detonation at different times.

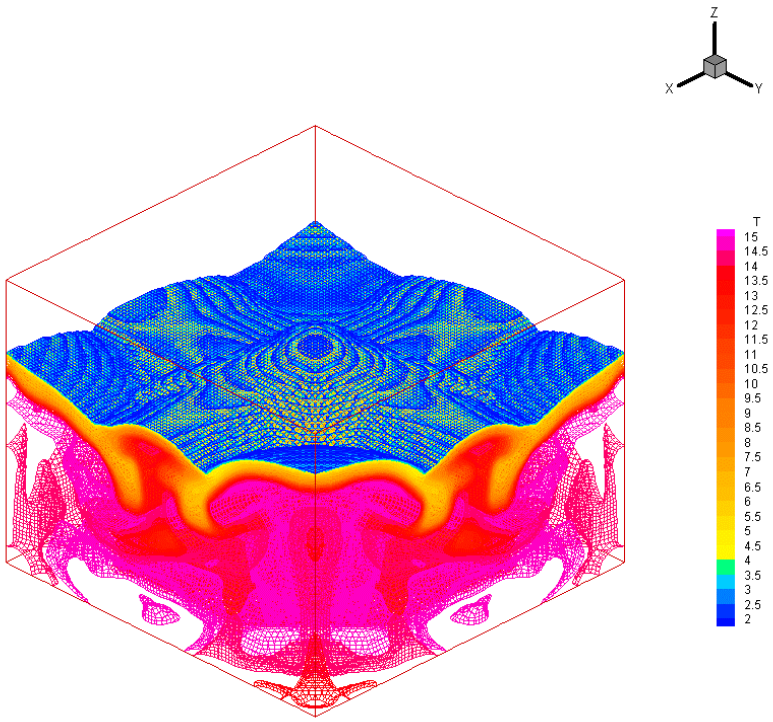

Fig. 6 Temperature contours of a three-dimensional detonation with solid boundary.

communication will severely depress the efficiency of parallel computing.

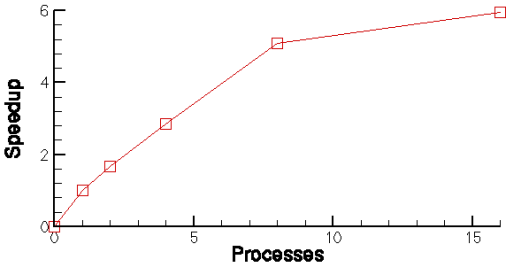

Fig. 9 Computational speed-up of the cluster.

Table 3 Blocking and non-blocking modes

Processes	Blocking Communication	Non- Blocking Communication
4	56	50
8	55	28
16	90	24

4. CONCLUSION

In this paper, we report our experience of developing and using a Beowulf cluster computer to perform large-scale CFD calculations. We also report the numerical results of three-dimensional detonations. The numerical method adopted, the CE/SE method is a highly accurate and efficient method. The synergy of this novel numerical method and the cost-effective parallel computation could point to a new direction for high performance computing for three-dimensional unsteady turbulence flows. According to our experience, data transfer rate is the bottleneck of large CFD simulation. To improve network performance, channel-bonding technology and well-organized network topology are necessary.

Acknowledgement

This work was performed under the support of NASA Glenn Research Center NCC3-580, and is part of an ongoing program at Wayne State University in applying the CE/SE method to practical engineering problems. The third author of the present paper is indebted to Professor L. Bauwens of University Calgary, Canada, for very helpful discussions.

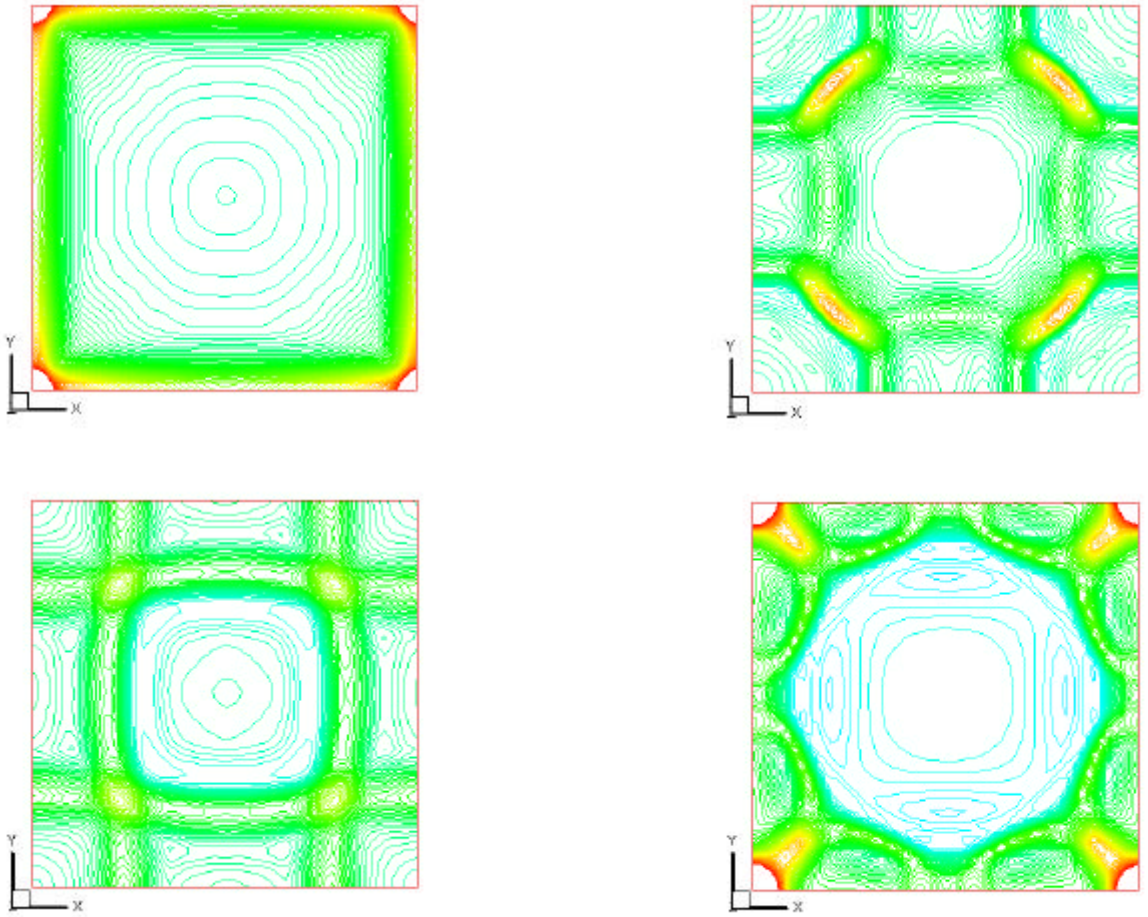

Fig. 7: Snapshots of pressure contours for the rectangular instability waves.

Fig. 8: Snapshots of pressure contours of diagonal instability waves at t = 12.2, 13.8, 14.6, 15.2, and 15.9.

REFERENCES

1. W. Fickett and W. C. Davis, "Detonation," University of California Press, Berkeley California, (1979).
2. Bourlioux and A. J. Majda, "Theoretical and Numerical Structure of Unstable Detonations," *Phil. Trans. R. Soc. Lond. A*, **350** (1995), pp.29-68.
3. D.N. Williams, L. Bauwens and E.S. Oran, "Detailed Structure and Propagation of Three-Dimensional Detonations", 26th Symposium (International) on Combustion, pp. 2991-2998, 1996.
4. S.C. Chang, "The Method of Space-Time Conservation Element and Solution Element – A New Approach for Solving the Navier Stokes and Euler Equations," *J. Comput. Phys*., **119**, pp.295-324, 1995.
5. S.T. Yu and S.C. Chang, "Treatments of Stiff Source Terms in Conservation Laws by the Method of Space-Time Conservation Element and Solution Element," *AIAA Paper 97-0435*, 35th AIAA Aerospace Sciences Meeting, January 1997, Reno, NV.
6. S.J. Park, S.T. Yu, M.C. Lai, S.C. Chang, and P.C.E. Jorgenson, "Direct Calculations of Stable and Unstable ZND Detonations by the Space-Time CE/SE method," *AIAA Paper 98-3212*, 34th AIAA Joint Propulsion Conference, July 1998, Cleveland, Ohio.