Application of Computational Steering Technique in Numerical Simulation of Explosion Problems

Jianguo Ning†, Cheng Wang†, Tianbao Ma†, and Wei Li††

†State Key Laboratory of Explosion Science and Technique, Beijing Institute of Technology, Beijing, 100081,China
††Beijing Municipal Institute of Labor Protection, Beijing, China

Summary
Steering computation is a fresh-new operational mode for constituting computational program. With the steering mechanism constituted by component communication mode, this paper designs CBCSS (COM Based Computational Steering System) to realize steering computation for numerical simulation program MMIC-2D of the two-dimensional explosion and impact problem developed by ourselves. Such constituted computational steering system well integrate the numerical simulation, visualized processing, user interface, and tracking and steering during operation of computational program. This steering mechanism only modifies the original computational program a little, so it facilitates its broad application.

Key words:
Numerical simulation, steering computation, scientific computation, visualization, components

1. Introduction

Ever since the 1980s, visual technology is widely applied in almost all fields that involve large-scale numerical simulation. According to the intimate extent of combination between visual processing course and numerical computation, we can divide visual technology into three forms: post-processing, tracking and computational steering.

Computational steering technology is also called interactive visual computing, which came into existence during the 1990s. Through computer software technology, it closely integrates numerical computation course, visual processing, and intervening method on computational course, ultimately constituting a computational steering system to offer interactive control capability to computational course. Namely, in parallel with numerical computation, it can choose and gather current computational data out of observational and research needs, real time generate images, and understand current computational status at any moment. It enables to feed back someone's judgment and decision to computational course, change parameter setting or computational method, and manipulate computational course for research and finding. This is of great significance to numerical simulation in need of complicated and time-consuming operation.
In other countries, computational steering technology has been placed due emphasis since its very inception. More than one decade's painstaking efforts contribute to significant development of computational steering technology in other countries. Furthermore, many a computational steering system that are put into practice have emerged, for instance, VASE [1] (Visualization and Application Steering System) by Illinois, SCIRun [2] (Scientific Computing &Image Research Group) by Utah State University’s Scientific Computation and Image Research Team, Progress (Progress stands for Program and Resource Steering System) by Georgia Institute of Technology, and MOSS (Mirror Object Steering System) by Magellan, Georgia Institute of Technology. They have been applied in a lot of fields like computational fluid mechanics and medical image processing.

There are a couple of ways to constitute computational steering system, but its basic structure is shown in Fig.1.

In this paper, we pinpoint MMIC-2D [3, 4, 5], a numerical simulation program for two-dimensional explosion and impact problem developed by ourselves, to realize steering computation. The computational steering system is constituted by components, and component communication mechanism is used in the design of steering mechanism, so it is named CBCSS (COM Based Computational Steering System) [6].

Manuscript received October 5, 2006.
Manuscript revised October 25, 2006.

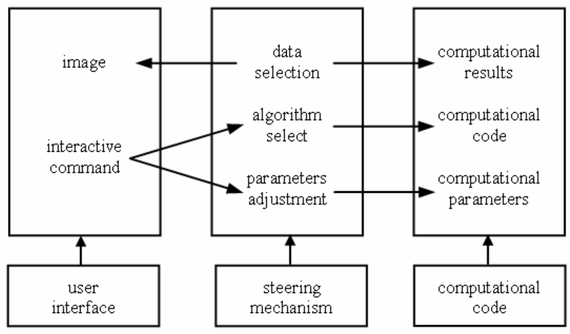

Fig. 1 Sketch of computational steering system

2. Numerical Algorithm of MMIC-2D

The MMIC-2D code is an Eulerian code for explosion and impact problems, which can simulate explosion in air, water, concrete, shaped charge jet, penetration problems and so on.

Governing equations of hydrodynamic model consist of mass conservation, momentum conservation, energy conservation, and equation of state. These partial differential equations form a two-dimensional unsteady problem of isothermal inviscid compressible fluid without body force :

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial z}+\frac{\partial \mathbf{G}}{r\partial r}=\mathbf{H}$$

$$\mathbf{U}=\begin{bmatrix}\rho \\ \rho u_z \\ \rho u_r \\ \rho E\end{bmatrix},\quad \mathbf{F}=\begin{bmatrix}\rho u_z \\ \rho u_z^2+p-S_{zz} \\ \rho u_r u_z - S_{rz} \\ \rho E u_z - p u_z - S_{zz}u_z - S_{rz}u_r\end{bmatrix}$$

$$\mathbf{G}=\begin{bmatrix} r\rho u_r \\ r(\rho u_z u_r - S_{rz}) \\ r\rho u_r^2 + p - rS_{rr} \\ r(\rho E u_r + p u_r - S_{rr}u_r - S_{rz}u_z)\end{bmatrix}$$

$$\mathbf{H}=\begin{bmatrix}0 \\ 0 \\ \dfrac{S_{zz}+S_{rr}}{r} \\ 0\end{bmatrix}$$

where t is time; r ,z are Euler coordinate system, r is radical direction, z is axial direction; u_r , u_z are the velocity components in r direction and z direction; ρ , p ,e are density, pressure and specific internal energy respectively; S_{rr}, S_{zz}, S_{rz} are deviatoric stresses.

In the MMIC-2D code, the operator split method is adopted, that is to say, the calculation for each time step is divided into two phases. The first phase is a Lagrangian phase in which the mesh is allowed to distort with the fluid. In this phase the changes in velocity and internal energy due to the pressure and deviatoric stress terms are calculated. In the Lagrangian phase both spatial directions z and r are calculated simultaneously. In the second advection phase, transport of mass, internal energy and momentum across cell boundaries are computed. This may be thought of as remapping the displaced mesh at the end of phase 1 back to the fixed Eulerian frame.

For the Eulerian method, the cell is fixed and materials flow in or out of the cell, so the mixed cell will exist. Here the Young's interface algorithm is adopted: using a line to stand for the interface, the slope of the line is determined by the material distribution in the surrounding eight cells, and the position of the line is determined by the portion of volume in the cell.

The MMIC-2D code uses the structural design method and is compiled by Fortran. As a batch program, we have no means to control it in the computational progress, so it is necessary to design a computational steering system for this code.

3.CBCSS Computational Steering System

CBCSS (COM Based Computational Steering System) mentioned in this paper is a component based computational steering system. Component is a method of constituting software system broadly used nowadays. The approach is put forward in response to increasingly complicated software system. In terms of ideology of component design, complicated software system is separated into small and functionally simple component modules. These modules can be compiled by different languages, be operating in different computers, and even be running in different operating systems. In order to allow such component modules to combine into a complete system, components have to follow detailed design criteria. CBCSS Computational Steering System in this paper uses the COM (Common Object Model) design criteria recommended by Microsoft.

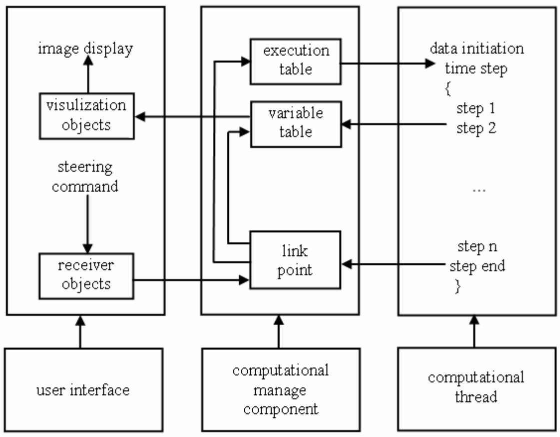

Fig.2 CBCSS system's structure

CBCSS Computational Steering System is composed of computational management component, computational thread and user interface, as shown in Fig.2. The core of the entire system is one component — computational management component; computational thread is a thread of special structure established by the computational management component, serving to execute computational task; user interface takes charge of visual processing, data registration, and send interactive commands to trace and steer computational thread.

3.1 Dynamic Link Library

Dynamic link library is formed during compilation of MMIC-2D computational program of Fortran version, which is called computational function library in CBCSS. The computational thread of CBCSS is designed by C++ language, so the computational modules (i.e., all sub-programs in MMIC-2D) in computational function library are exported in the form of C function, which are used in CBCSS system after the formation of dynamic link library. The computational thread can use the sub-programs as standard C function.

3.2 Computational Management Component

Computational management component is the core of the entire system. On one hand, it manages computational thread, and on the other hand, it communicates with user interface. There is a two-way communication between computational management component and user interface: real time data obtained is transmitted to user interface, whilst the interactive commands from user interface are sent to computational management component. Two-way communication commitment between application components serves to complete communication between computational management component and user interface and synchronous constraint with computational thread. Generally, components provide services through interface. Components achieve their functions on their own. Clients use the functions of the components via interface. Such interface is called incoming interface. In this occasion, clients are in active status, so it can be treated as communication from clients to components. Another kind of interface of components is outgoing interface. At this time, clients achieve the functions of interface. The object among clients that achieves outgoing interface of components is called receiver. Outgoing interface allow components to send event, notice or request to clients, thus achieving the communication from components to clients.
Computational management component manages computational thread by way of:

(1) Setting up execution list to define concrete execution procedures for computational thread.

(2) Taking charge of creating and starting computational thread.

(3) Outgoing interface receives the steering commands from user interface, which is manifested as a connecting point in computational management component. Computational thread tests the information of the connecting point obtaining steering commands. The constitution of the outgoing interface of computational management component ensures the execution of steering commands to occur at the interval between the computation completion of one time step of computational thread and yet-starting of next time step, namely, achieving synchronous constraint.

Computational management component provides the basic functions for achieving steering computation. Execution list is defined in computational management component to prepare information necessary to execute and change computation steps. And the execution list specifies the execution contents of computational thread. Before creating computational thread, computational management component firstly constitutes an execution list for it, and arranges in order the pointer of each computational module in the execution list according to the execution procedures of computational program. After being created, computational thread read function pointer from the execution list, and call upon function in turn.

3.3 Computational Thread

Computational thread adopts the form of thread in Windows system, which is an independently run unit, and can work in parallel with other parts of the system. It mainly accomplishes the computational task specified by computational management component, and responds to control command.

The execution contents of computational thread are identified by the execution list defined by computational management component. The execution list gives the pointer of computational module in computational function library. Computational thread reads the pointers of computational module in turn according to the execution list and executes corresponding computation function.

3.4 User Interface

User interface displays the graphics and images generated by visualization processing, and forms interactive command to manage computation course, visualization processing, data and image.

4. Function of CBCSS in Application

4.1 Numerical Simulation of Explosion Field with Protective Wall

Protective wall has protective functions as to explosion wave. If the function of earth shock wave is ignored, the ground and protective wall can be considered to be rigid. That is to say, we suppose the ground and protective wall can not absorb explosion energy by elastoplasticity deformation. Thus there are only air and explosive products in computation domain [7]. The initial field distribution is shown in Fig.3.

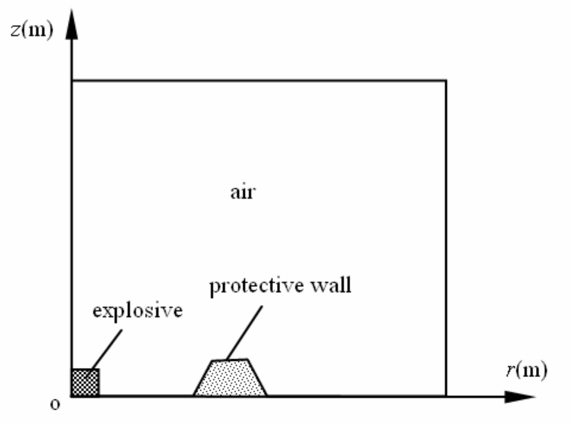

Fig.3 Sketch of explosion field with protective wall

In the simulation of explosion field with protective wall, some functions of CBCSS are as follows:

1) Markers

Some markers can be inserted at any time during computation course. The markers, which are set in user interface, will not attend the actual computation process. A group of marks are set in a given position in computation domain and the corresponding receivers are set to get the velocity data for each time step. According to the velocity and time step, we can get the displacement of each marker for each time step, and further the movement track of each marker can be obtained, reflecting the local variation situation of fluid field. Fig. 4 indicates the movement tracks of markers in computation process.

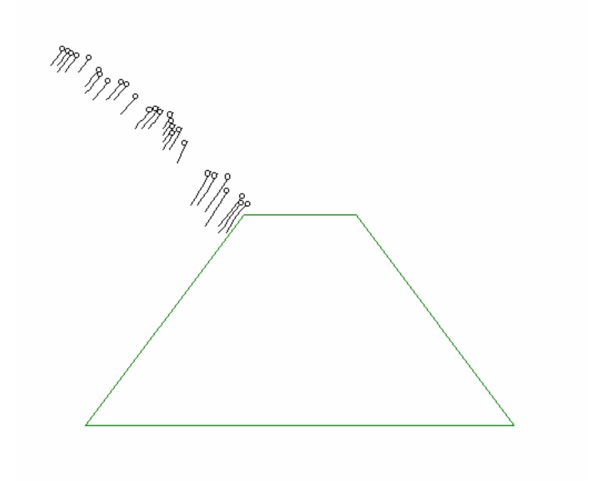

Fig.4 Sketch of markers tracks

2) Tracking

CBCSS can obtain computational data at any time during computation process, and then we get real-time image by visualization to track the physical process by means of pressure data analysis. Fig. 5 shows the image that reflects the propagation process of shock wave. The front of shock wave and Mach reflection can be identified in Fig. 5.

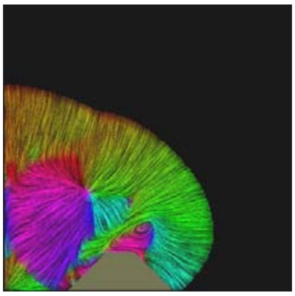

Fig.5 Pressure data image

3) The visualization treatment of vector data

The CBCSS improves the traditional LIC method and identifies the direction of velocity by different colors. The texture produced according to velocity data is shown in Fig. 6.

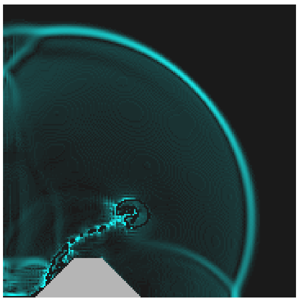

Fig.6 Velocity data image

4) The movement of computation domain
The size of the initial computation domain should not be too large in order to save memory space and computation time. However, the main physical process possibly goes beyond the computation domain as the computation time advances. Thus the computation domain needs to be moved for tracking the physical process. The computation domain movement is steered by CBCSS.

4.2 Numerical Simulation of Shaped Charge Jet

A cylinder of explosive with a hollow cavity in one end and a detonator at the opposite end is known as a shaped charge. If the hollow cavity is lined with a thin layer of metal, the liner forms a jet when the explosive charge is detonated. During the process of jet formation, depicted in Fig. 7 for a typical conical shaped charge, the liner material region is driven to very violent distortions over very short time intervals. In numerical simulation, these regions need finer mesh spacing than other regions. In view of this, local mesh refinement algorithm is proposed to achieve high interface tracking resolution and make it accord with actual physical process [8, 9].

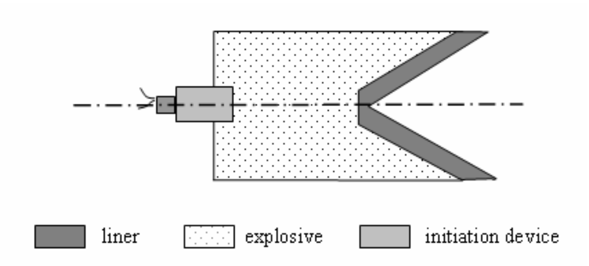

Fig.7 Sketch of conical shaped charge

The basic idea of mesh refinement is that when there exists an interface in a grid, the grid needs to be refined, namely, the grid is divided into four subgrids. When there is no material interface in the grid after a period of computation time, the grid is incorporated. The quadtree date structure is adopted for in mesh refinement. When one basic grid (called “parents”) is refined into next level grids called “children”, as shown in Fig. 8 and Fig. 9. According to the tree structure, some necessary related information in computation can be easily found. Fig. 10 illustrates the material interface after mesh refinement

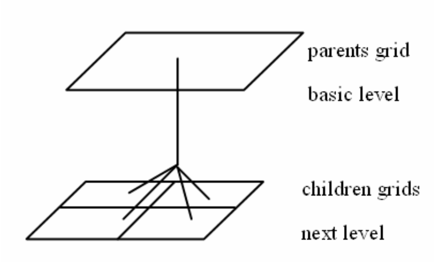

Fig.8 Sketch of quadtree mesh

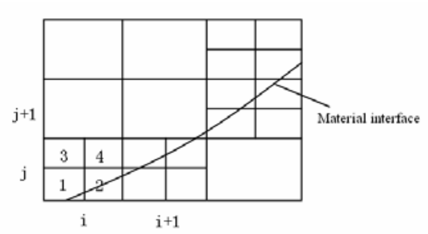

Fig.9 Sketch of mesh distribution

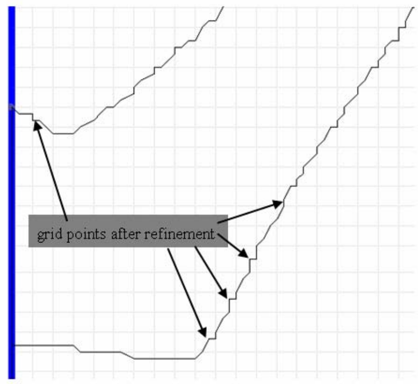

Fig.10 Material interface after refinement

5. Conclusion

CBCSS Computational Steering System proposed in this paper can achieve steering computation for the existing

FORTRAN version numerical simulation and computation program MMIC-2D for two-dimensional explosion and impact problem. It can obtain flexible and diversified tracking to the execution process of the program, whilst it only slightly changes the original computational program. Moreover, CBCSS is constituted by applying components, which is easy to be transplanted into distribution and network environment.

Acknowledgments

This work is supported by National Nature Science Foundation of China under grant number 10572026.

References

[1] D.J. Jablonowski, J.D. Bruner, B. Bliss and R.B. Haber. VASE: The Visualization and Application Steering Environment. In Proceedings of Supercomputing '93, (1993) 560-569
[2] C.Upson, et al.: The Application Visualization System: A Computational Environment for Scientific visualization. IEEE CG&A, Vol.9, No. 4. (1989) 30-42
[3] Tianbao Ma and Jianguo Ning: Numerical Simulation of Sphere Charge Exploding in Soil Medium. Theory and Practice of Energetic Materials, Vol. VI. (2005) 965-969
[4] Wenzhao Zhang, Jianguo Ning and Zuguo Zheng: Visualization of Explosive Field. Journal of Beijing Institute of Technology, Vol.8, No.4. (1999) 424-429
[5] Jianguo Ning and Kaiteng Wu: A Novel Space-Time MmB Difference Scheme with Second-order Accuracy for 3 Dimensional Flows. International Journal of Nonlinear science and Numerical Simulation, Vol.6, No.4. (2005) 417-420
[6] Wei Li: Study on Computational Steering Technique in Numerical Simulation of Explosion and Shock Problems. PhD Thesis of Beijing Institute of Technology. (2002)
[7] Yulin Shen and Jianguo Ning: Numerical Simulation of the Shape of the Protective Wall's Effect on the Shock Wave on 2-D Explosive Field. Journal of Beijing Institute of Technology, Vol.10, No.1. (2001) 39-44
[8] Jianguo Ning, Cheng Wang and Tianbao Ma: Numerical Analysis of The Shaped Charged Jet with Large Cone Angle. International Journal of Nonlinear Sciences and Numerical Simulation, Vol.7, No.1. (2006) 71-78
[9] Jianguo Ning, Tianbao Ma and Cheng Wang: 2-D Numerical Simulation of Explosion and Impact Problems. Proceeding of WCCM VI in Conjunction with APCOM'04. (2004) 515

Jianguo Ning received the B.S. and M.S. degrees in solid mechanics from Lanzhou University in 1985 and 1988, respectively and the PhD degree in solid mechanics from Taiyuan University of Technology in 1992. He worked in Peking University as a postdoctor from 1992 to 1994. At present, He serves as a professor and doctor supervisor in Beijing Institute of Technology, and his current research interests involve explosion mechanics, computational mechanics, and dynamic mechanical behavior of material.