ARTICLE IN PRESS

Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Parallel implementation of geometrical shock dynamics for two dimensional converging shock waves

Shi Qiu, Kuang Liu, Veronica Eliasson *

Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089-1191, USA

ARTICLE INFO

Article history:
Received 2 February 2016
Received in revised form
18 May 2016
Accepted 8 June 2016
Available online xxxx

Keywords:
Geometrical shock dynamics
Parallel computing
Converging shock
Symmetric boundary conditions

ABSTRACT

Geometrical shock dynamics (GSD) theory is an appealing method to predict the shock motion in the sense that it is more computationally efficient than solving the traditional Euler equations, especially for converging shock waves. However, to solve and optimize large scale configurations, the main bottleneck is the computational cost. Among the existing numerical GSD schemes, there is only one that has been implemented on parallel computers, with the purpose to analyze detonation waves. To extend the computational advantage of the GSD theory to more general applications such as converging shock waves, a numerical implementation using a spatial decomposition method has been coupled with a front tracking approach on parallel computers. In addition, an efficient tridiagonal system solver for massively parallel computers has been applied to resolve the most expensive function in this implementation, resulting in an efficiency of 0.93 while using 32 HPCC cores. Moreover, symmetric boundary conditions have been developed to further reduce the computational cost, achieving a speedup of 19.26 for a 12-sided polygonal converging shock.

1. Introduction

Focusing of shock waves can generate extreme conditions, such as high pressure and temperature, at the focal region. Shock focusing occurs in a variety of man-made and naturally occurring events, for example in extracorporeal shock wave lithotripsy [1], inertial confinement fusion [2] and collapse of cavitation bubbles [3]. The first researcher to study shock focusing was Guderley [4], who developed analytical solutions for converging cylindrical and spherical shock waves. Following his work, numerous authors have investigated on this area. Basically, there are two major numerical methods used to study shock wave propagation: one is the Navier–Stokes equations, and the other is the inviscid Euler equations if viscosity in the shocked medium can be neglected. The advantage of these two methods is that a full flow field can be accurately obtained. However, the strength of the shock front in the focusing area can be much higher than that of the initial shock front, resulting in smaller time scales to maintain the Courant–Friedrichs–Lewy (CFL) condition. In addition, the length of the shock front in the focusing area can be much smaller than that of the initial shock front. In order to resolve all small scales close to, and in, the focal region, high resolution in both time and space are required, which can make the computational task very expensive. Whitham proposed an alternative method [5], named Geometrical Shock Dynamics (GSD), that describes the motion of shock waves in a different way. Unlike the Navier–Stokes equations and the Euler equations, this theory avoids dealing with the flow field around the shock and only focuses on the curvature of the shock wave itself. As a result, solving a shock focusing event with GSD instead of the Navier–Stokes equations or the Euler equations, the computational complexity is reduced by solving a lower dimensional problem. In addition, the actual computational cost may be reduced by more than an order of magnitude depending on the required grid resolution when dealing with higher dimensional problems. In GSD, the shock front is discretized into small elements. Between each element, orthogonal trajectories are introduced as rays so that each shock front element can be approximated to propagate down a tube whose boundaries are defined by the rays, a so-called ray tube. The main assumption in GSD is that the motion of the shock only changes with the variation of the ray tube area. Then, instead of solving the full Euler equations, the motion of the shock can be predicted by deriving the relation between shock strength, which can be represented by the Mach number, M, and the area of the ray tube, A. This is the so-called Area-Mach

* Corresponding author.
E-mail address: eliasson@usc.edu (V. Eliasson).

http://dx.doi.org/10.1016/j.cpc.2016.06.003
0010-4655/

number (A–M) relation,

$$\frac{1}{A(x)}\frac{dA}{dM} = -g(M), \tag{1}$$

where

$$g(M) = \frac{M}{M^2-1}\left(1+\frac{2}{\gamma+1}\frac{1-\mu^2}{\mu}\right)\left(1+2\mu+\frac{1}{M^2}\right), \tag{2}$$

$$\mu^2 = \frac{(\gamma-1)M^2+2}{2\gamma M^2-(\gamma-1)}. \tag{3}$$

Here, γ represents the adiabatic index, x denotes the distance in the ray tube and the cross sectional area $A(x)$ is a function of x, see Ref. [6] for additional details.

There exists a variety of algorithms to implement GSD numerically. Here, only a part of those works are listed. The method of characteristics was used by Bryson and Gross [7] to analyze shock diffractions. Decades later, a front tracking approach was developed by Henshaw [8] in two dimensions and Schwendeman [9] in three dimensions. A conservative finite difference method was formulated by Schwendeman [10]. Most recently, a fast-marching like algorithm was developed by Noumir et al. [11]. Among all these algorithms, the front tracking method is the most used one due to its computational accuracy and simplicity of implementation. For example, Schwendeman applied this method to study shock motion in non-uniform media [12]. Best modified it to investigate underwater explosions [13,14], and Apazidis and Lesser utilized it to compare with the shock converging experiments [15]. However, in order to utilize the front tracking method to study shock motion for large scale applications, the computational cost is still a large bottleneck, which can be addressed through parallel computing techniques.

In our work, the front tracking method has been implemented on parallel computers and symmetric boundary conditions has been developed in order to reduce the computational expense dramatically.

2. Numerical methods

2.1. Serial scheme

In this study, the numerical implementation of GSD is based on previous front tracking methods [8,13,14]. A short review about implementation of the numerical scheme is given as follows. In a two dimensional condition, the shock front is discretized into N points denoted by $\boldsymbol{x_i}(t)$, where $i = 1, 2, \ldots, N$. The shock front propagates along the direction of its normal vector, and the speed is determined by the A–M relation. The motion of the shock is described by

$$\frac{d\boldsymbol{x_i}(t)}{dt} = a_0 M_i(t)\boldsymbol{n_i}(t), \quad i = 1, \ldots, N, \tag{4}$$

where $M_i(t)$ and $\boldsymbol{n_i}(t)$ denote the Mach number and shock front normal at $\boldsymbol{x_i}(t)$, respectively. The speed of sound in ambient air is fixed as $a_0 = 1$ in all of our calculations. A fourth-order Runge–Kutta scheme is adopted to numerically integrate equation (4).

The Mach number, $M(t)$, is calculated by combining the A–M relation from Eq. (1) with the shock front relation $d_t A = a_0 M d_x A$. Thus, Eq. (1) can be converted into

$$\frac{dM}{dt} = \frac{M}{-g(M)}\frac{A'}{A}. \tag{5}$$

In Eq. (5), A' denotes $d_x A$ and A'/A can be expressed explicitly as [13],

$$\frac{A'}{A} = \frac{\partial \boldsymbol{x}(s(t),t)}{\partial s(t)}\frac{\partial \boldsymbol{n}(s(t),t)}{\partial s(t)}, \tag{6}$$

where, as is shown below, the arclength $s(t)$ represents the geometry of the shock front and $\boldsymbol{n}(s(t),t)$ indicates the norm vector of the shock front,

$$s_i(t) = \begin{cases} 0, & \text{if } i = 1; \\ s_{i-1}(t) + |\boldsymbol{x_i}(t) - \boldsymbol{x_{i-1}}(t)|, & \text{if } i = 2, \ldots, N; \end{cases} \tag{7}$$

$$\boldsymbol{n}(s(t),t) = \left(\frac{\partial y(s(t),t)}{\partial s(t)}, -\frac{\partial x(s(t),t)}{\partial s(t)}\right). \tag{8}$$

Following [8], there are two extra procedures. One is a point insertion and deletion approach that maintains the resolution of the shock front and the CFL condition. Additionally, a time-step reduction scheme [16] should be applied to replace this approach for a particular converging shock scenario. The other scheme is a two-step smoothing procedure to reduce the high frequency errors. The main focus of this study is to implement the algorithm for converging shocks. Thus, here the time-step reduction scheme is adopted. However, the parallel implementation can be coupled with the point insertion and deletion approach to tackle other shock wave simulations.

The time-step reduction scheme can be expressed by the following equation:

$$\Delta t < f(R(t), M(t)), \tag{9}$$

where $f(R(t), M(t))$ is a function of shock radius $R(t)$ and $M(t)$ at time t. More details of this scheme can be found in [16]. If this condition fails, Δt will be reduced as $\Delta t_{new} = 0.75\Delta t$ and Eqs. (4) and (5) are restarted by using Δt_{new}.

The two-step smoothing procedure is given as follows:

$$\boldsymbol{x_i}(t) = \frac{1}{2}(\boldsymbol{x_{i-1}}(t) + \boldsymbol{x_{i+1}}(t)), \tag{10}$$

and it is applied every n_s iterations, where n_s depends on the time increment, Δt, and the average shock arclength between each discrete point, Δs_{avg}. When $\Delta s_{avg} = 0.01$, n_s is usually set between 10 and 50.

A schematic flow chart shown in Fig. 1 illustrates how this algorithm works. In general, the algorithm consists of four functions during the time marching process. At time t, A_i'/A_i is calculated first by Eq. (6). The two components $\partial \boldsymbol{x}(s(t),t)/\partial s(t)$ and $\partial \boldsymbol{n}(s(t),t)/\partial s(t)$ in Eq. (6) can be obtained by applying a cubic spline interpolation method using discrete data $s_i(t), \boldsymbol{x}_i(t)$ and $s_i(t), \boldsymbol{n}_i(t), i = 1, \ldots, N$ under adequate boundary conditions, which vary according to the task settings. This step is named *updateAda*. The advantage of setting *updateAda* initially is that it can also be used to generate the norm vector of the shock front, see Eq. (8), which is required for updating the shock location in the third step. Next, a fourth-order Runge–Kutta scheme is employed, see Eq. (5), to update the Mach number at the current time iteration. This step is named *updateM*. In the next function, called *updateX*, the shock location for the next time iteration is calculated by using Eq. (4) so that $\boldsymbol{x}(t+\Delta t)$ is obtained. Later, the time-step reduction scheme is applied to maintain the CFL condition, see Eq. (9). The next function, *updateS*, computes $s(t+\Delta t)$ through Eq. (7). In addition, the smoothing procedure, see Eq. (10), is implemented after *updateS*. At the end of each iteration, a terminate condition is given to determine whether the program should be aborted. In order to run this algorithm, a set of coordinates, $\boldsymbol{x_i}(t_0)$, and Mach number, $M_i(t_0)$, representing the initial location and strength of the shock front are given as initial conditions. In addition, *updateX* and *updateS* need to be called before the time marching to obtain $s_i(t_0+\Delta(t))$, which is the arclength for the first iteration. These initial steps are implemented in the parameter initialization, shown in Fig. 1.

Please cite this article in press as: S. Qiu, et al., Parallel implementation of geometrical shock dynamics for two dimensional converging shock waves, Computer Physics Communications (2016), http://dx.doi.org/10.1016/j.cpc.2016.06.003