Computers & Fluids 110 (2015) 9–18

Contents lists available at ScienceDirect

Computers & Fluids

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

Analysis of the second order accurate uniform equilibrium flux method and its graphics processing unit acceleration

Yen-Chih Chen [a], Matthew R. Smith [a,*], Alexander Ferguson [b]

[a] Dept. of Mech. Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC
[b] Beca Limited, Level 5, 825 Ann Street, Fortitude Valley, QLD, Australia

ARTICLE INFO

Article history:
Received 6 December 2013
Received in revised form 15 September 2014
Accepted 27 November 2014
Available online 20 December 2014

Keywords:
Computational fluid dynamics
CFD
Parallel computing
Graphic processing units
GPU
CUDA
Finite volume method
FVM
High resolution
Total variable diminishing
TVD

ABSTRACT

The extension of the Uniform Equilibrium Flux Method (UEFM) to second order accuracy in space is presented. This extension is made possible through the recasting of the original UEFM flux expressions from a volumetric flux to a surface flux, allowing for reconstruction through a Taylor series expansion of the resulting split surface fluxes at the cell interfaces. By doing so, we avoid the difficulties associated with integration of the gradient terms over velocity and physical space as required by the original UEFM fluxes. Analysis of the dissipative qualities of the renewed direction split UEFM flux expressions demonstrate that the numerical dissipation is a function of Mach number, with increasing amounts of dissipation present with increasing Mach numbers. Following this analysis, the higher order UEFM fluxes are applied to large scale parallel computation using Graphics Processing Units, or GPUs, through the use of CUDA. The vector split nature of the UEFM fluxes are well suited to GPU computation due to the high degree of locality. This parallelization is performed using a cell-based parallel paradigm through the creation of several key CUDA kernels for the calculation of split fluxes, gradient of split fluxes and state related computations. The algorithm is executed entirely on the GPU device, with the host remaining idle during the computation stage. The GPU accelerated UEFM algorithm is then applied to the solution of several two dimensional benchmark problems. Speedup of approximately 200 and 171 times for first order accuracy and second order accuracy respectively is demonstrated when using an Nvidia Tesla C2075 computing GPU compared to that of a single core of an Intel Xeon E5-2760 CPU.

1. Introduction

The Finite Volume Method (FVM) has evolved into one of the cornerstone methods of modern Computational Fluid Dynamics (CFD). In the Finite Volume Method, the flow region is discretized into finite volumes, known as cells, and fluxes of conserved quantities – notably mass, momentum and energy – are computed at each cell surface. Application of the divergence theorem to the governing partial differential equations provides us a simple set of equations for updating the cells state (i.e. conserved quantities) in terms of these fluxes across cell surfaces. Since each interface has a unique flux based on the conditions on either side of the interface, and the time evolution of the state for each cell is conservative.

The calculation of fluxes across any given interface – known as the Riemann problem – has been thoroughly investigated over the history of the Finite Volume Method. Several varieties of methods exist for the computation of these fluxes, ranging from analytical approaches, integral balance methods and methods centered on the specific behavior of the problem at hand. One of these such approaches employs the Kinetic Theory of Gases to compute the fluxes across a cell interface by taking moments over the particle velocity distribution function at the interface. Perhaps the first of such methods was introduced by Pullin [1] in the Equilibrium Flux Method (EFM) – using the Maxwell–Boltzmann equilibrium velocity probability distribution function, moments were taken around the distribution functions either side of the cell interface. This method has been refined and improved over the course of the last couple of decades [2], however the core concept remains more or less unchanged.

Despite being computationally efficient when compared to several analytical schemes [2], the EFM fluxes contain several expensive function evaluations, namely the exponential and error functions, which must be evaluated approximately. Hence, several strategies have been used to increase the efficiency of the EFM scheme. The Quiet Direct Simulation (QDS) [3–5] can be shown to be an approximation to the EFM scheme which works by

* Corresponding author.
E-mail address: msmith@mail.ncku.edu.tw (M.R. Smith).

http://dx.doi.org/10.1016/j.compfluid.2014.11.031
0045-7930/© 2014 Elsevier Ltd. All rights reserved.

replacing the moment integrals with a numerical approximation of the integral using Gauss–Hermite quadrature. Following this, Ferguson et al. [6] developed the Uniform Equilibrium Flux Method (UEFM) – an approximation to Smith et. al.'s TDEFM [2] – in which the governing equilibrium velocity distribution probability function was replaced by a series of Heaviside step functions, which was integrated over the cell volume and again over velocity space, leading to an increase in accuracy and efficiency over the QDS scheme.

One of the original challenges with the true directional fluxes obtained by TDEFM was the extension to higher order accuracy. Traditional approaches to obtaining higher order spatial accuracy in Finite Volume based approaches is through the incorporation of gradients in primitive and conserved quantities. The inclusion of gradients into the UEFM and TDEFM during the velocity and spatial integration phases is problematic. No analytical solution exists for the inclusion of energy gradients using the TDEFM approach, and while this is possible in UEFM, the resulting equations are very complex. Hence, as opposed to employing a true direction (volumetric) flux, we propose the use of a traditional surface flux for the computational of higher order fluxes. In other words, the fluxes may be computed by a single moment around velocity at the cell interface as opposed to the dual volume/velocity moments previously employed. Following the calculation of first order fluxes, the higher order expansion is obtained by using a Taylor series expansion around the split first order fluxes at the cell interfaces, allowing us to more accurately estimate the value of the forward and reverse fluxes. To maintain positivity, we need to limit these gradients using a slope limiting function.

The resulting forward and backward direction split fluxes are highly localized. Each cell, with its own unique conditions, requires computation of split fluxes in each spatial dimension simulated, with these fluxes being independent of conditions in nearby cells. Hence, such a vector split approach is ideally suited toward an SIMD (cell-based) or vector-based parallelization paradigm. This form of parallelization is not new; however, this research field is evolving rapidly – tools such as Advanced Vector eXtensions (AVX), the logical successor of the SIMD Streaming Extensions (SSE), and Graphics Processing Units (GPU's) for parallel computation mean that parallelization of CFD algorithms is rapidly becoming mainstream. Here, we focus on the use of Nvidia's CUDA [7] to aid in the parallel computation of the various elements of UEFM. The computation of split fluxes, split flux gradients and change of states are chosen as key CUDA kernels.

2. Direction decoupled uniform equilibrium flux method

Previously, the Uniform Equilibrium Flux Method (UEFM) was derived using a volume-to-volume flux computation for the sake of reducing the errors associated with direction decoupling [2,6]. The starting point for the UEFM method was the particle equilibrium velocity distribution function, which in this case, due to the assumption of equilibrium, was the Maxwell–Boltzmann probability distribution function. In its one dimensional form, this can be written as:

$$F(v) = \frac{1}{\sqrt{2\pi s^2}} \exp\left[\frac{-(v-\bar{v})^2}{2s^2}\right] \tag{1}$$

where v is the particle velocity, s is the standard deviation of the distribution function ($s = RT^{1/2}$) and the bulk velocity is indicated by the overscore. The original Uniform Equilibrium Flux Method (UEFM) aimed to replace this function with a composite distribution function consisting of a series of uniform step functions, as demonstrated in Fig. 1. In a stricter mathematical sense, we approximate the continuous velocity probability distribution function with the summation of continuous probability distribution functions over discontinuous (discrete) bounds, i.e.:

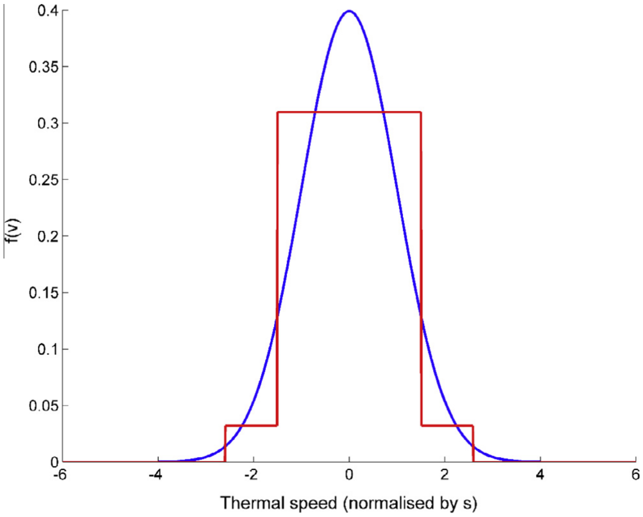

Fig. 1. Comparison of the continuous velocity distribution function (Maxwell–Boltzmann) to its composite approximation through the use of two uniform step functions. In this case, the weights $w = [1/6, 5/6]$ and thermal speeds $a = [3/2(3)^{0.5}, 3/2]S$ where used in the approximating composite function.

$$f(v) \approx \sum_{i=1}^{N} w_i f(a_i) \tag{2}$$

The true directional flux – in the case of UEFM – was found by integrating the particle probability distribution function over both velocity space and physical space. Assuming particle velocities are subject to a uniform distribution function bound by velocities V_R and V_L, the re-distribution of mass (per unit source mass) having moved for a period Δt is:

$$G(x_0, x, V_L, V_R, \Delta t) = \frac{1}{(V_R - V_L)\Delta t}[H(x0, x, V_L) - H(x0, x, V_R)] \tag{3}$$

where $H(x0, x, V) = H[-x0 + x - V\Delta t]$ is the Heaviside unit step function;

$$H(x) = \begin{cases} 0, x < 0 \\ 1, x \geqslant 0 \end{cases} \tag{4}$$

The function G represents a spatial probability distribution function describing how particles are distributed over space x from starting location x_0 as shown in Fig. 2. Taking the moment of G around x_0 over the width of the source region provides the source cell average value of G, written as $\bar{G}_{MASS}$ with units of (m^{-1}):

$$\bar{G}_{MASS} = \frac{1}{(x_R - x_L)} \int_{x_L}^{x_R} G dx_0 \tag{5}$$

The integral of this over all possible values of x returns unity – meaning that all mass must land in some finite destination region. The net flux of mass (per unit source mass) from the source region between x_L and x_R and destination region between x and $x + \Delta x$ is given by $F_{MASS} = \bar{G}_{MASS}\Delta x$. Complete integration of this expression (with respect to destination location x) across a destination region (x_l, x_r) then gives the total flux of mass (per unit source mass) from the source to the destination region:

$$F_{MASS}(x_l, x_r, x_L, x_R, V_L, V_R) = \int_{x_l}^{x_r} \bar{G}_{MASS} dx \tag{6}$$

Further details of the complete UEFM flux derivation can be found in [6]. This approach succeeds in creating split volumetric