基于CPU-GPU的超音速流场N-S方程数值模拟

卢志伟* 张皓茹 刘锡尧 王亚东 张卓凯 张君安
西安工业大学机电工程学院,西安,710021

摘要:为深入分析超音速流场的特性并提高数值计算效率,设计了一种高效的加速算法。该算法充分利用中央处理器-图形处理器(CPU-GPU)异构并行模式,通过异步流方式实现数据传输及处理,显著加速了超音速流场数值模拟的计算过程。结果表明:GPU并行计算速度明显高于CPU串行计算速度,其加速比随流场网格规模的增大而明显提高。GPU并行计算可以有效提高超音速流场的计算速度,为超音速飞行器的设计、优化、性能评估及其研发提供一种强有力的并行计算方法。

关键词:超音速流场;中央处理器-图形处理器;异构计算;有限差分
中图分类号:V211.3
DOI:10.3969/j.issn.1004-132X.2025.09.005 **开放科学(资源服务)标识码(OSID)**:

Numerical Simulation of N-S Equations for Supersonic Flow Fields Based on CPU-GPU

LU Zhiwei* ZHANG Haoru LIU Xiyao WANG Yadong ZHANG Zhuokai ZHANG Jun'an
School of Mechatronic Engineering, Xi'an Technological University, Xi'an, 710021

Abstract: To thoroughly analyze the characteristics of supersonic flow fields and enhance the efficiency of numerical computations, an efficient acceleration algorithm was designed. The algorithm herein fully leveraged the CPU-GPU heterogeneous parallel architecture and achieved data transmission and processing through asynchronous streaming, significantly accelerating the computational processes of supersonic flow field numerical simulations. The results demonstrate that the computational speed of GPU parallel processing is markedly faster than that of CPU serial processing, and the speedup ratio exhibits a pronounced increasing trend as the scale of the flow field grid expands. GPU parallel computing may effectively improve the computational speed of supersonic flow field simulations, providing a robust parallel computing method for the design, optimization, performance evaluation, and development of supersonic aircrafts.

Key words: supersonic flow field; central processing unit - graphics processing unit(CPU-GPU); heterogeneous computing; finite difference

0 引言

近年来,超音速技术在军事(如反舰导弹和巡航导弹)和航空航天领域(如超音速飞机和高超音速飞机)得到广泛应用,为国防和民用提供了重要支持。由于在现实中对超音速飞行器和导弹进行实物试验难度大且成本高,通常会采用数值模拟技术模拟实物在不同条件下的性能[1-3]。随着流场计算模型和规模的不断增大,传统上通过在中央处理器(CPU)上开展并行计算已经无法满足日益扩大的计算需求[4-5]。为了提高计算效率,研究人员利用图形处理器(GPU)众核架构强大的浮点运算能力和高速存储带宽高效并行地求解流体力学基本方程[6-7],这为超音速技术的研究和发展提供了重要的支持。

通常采用CPU/GPU异构并行计算模式来实现更高效的并行算法[8-9]。在此模式下,CPU负责控制逻辑和流程,GPU进行大规模数据并行计算。通过充分发挥CPU和GPU的各自优势,可以提高计算效率和精度,实现流场数值模拟的高效计算[10-13]。

国内外学者对基于异构并行计算模式实现流体力学方程求解开展了大量研究。YE等[14]将曲线坐标上采用高阶有限差分方法的内部多块结构化CFD求解器移植到GPU平台上,提出了一套硬件感知技术来优化CPU与GPU之间的数据传输效率并提高GPU之间的通信效率,测试结果表

收稿日期:2024-07-15
基金项目:国家自然科学基金(52301101);陕西省科技厅项目(2025ZG-JBGS-007);陕西省教育厅项目(23JC040)

明：对不同的GPU可以取得明显的加速效果。GHIOLDI等[15]通过CPU/GPU异构并行采用五阶龙格-库塔方法实现可压缩纳维-斯托克斯(N-S)方程的求解，加速计算超音速条件下有限速率化学反应，结果表明：异构并行模式可以显著提高求解器的执行速度，比在相同规模128个CPU上运行相同求解器快9.3倍。张东飞等[16]采用GPU加速高阶谱差分方法实现对三维非定常可压缩N-S方程的求解，测试了GPU对静止计算域和旋转计算域的加速效果，结果表明：相比于单个CPU，采用运动网格的GPU计算加速比为13.9。

基于以上研究，本文提出利用CPU-GPU异构并行模式并通过异步流方式实现数据传输及处理，显著加速流场计算。

1　控制方程及数值计算方法

1.1　控制方程

本文采用数值计算方法研究超音速单平板流场，其理论基础是流体力学基本方程N-S方程，包括连续性方程、x方向动量方程、y方向动量方程、能量方程，用张量形式表示如下：

连续性方程

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x}(\rho u)+\frac{\partial}{\partial y}(\rho v)=0 \tag{1}$$

式中：ρ为流体密度；t为时间；u、v分别为流场在x方向和y方向的速度分量。

x方向动量方程

$$\frac{\partial}{\partial t}(\rho u)+\frac{\partial}{\partial x}(\rho u^2+p-\tau_{xx})+\frac{\partial}{\partial y}(\rho uv-\tau_{yx})=0 \tag{2}$$

式中：p为流场压力；τ_{xx}、τ_{yx}分别为流场的正应力和切应力。

y方向动量方程

$$\frac{\partial}{\partial t}(\rho v)+\frac{\partial}{\partial x}(\rho uv-\tau_{xy})+\frac{\partial}{\partial y}(\rho v^2+p-\tau_{yy})=0 \tag{3}$$

式中：τ_{yy}为流场的正应力；τ_{xy}为流场的切应力，与τ_{yx}值相等。

能量守恒方程

$$\frac{\partial}{\partial t}(E_t)+\frac{\partial}{\partial x}[(E_t+p)u+q_x-u\tau_{xx}-v\tau_{xy}]+\frac{\partial}{\partial y}[(E_t+p)v+q_y-u\tau_{yx}-v\tau_{yy}]=0 \tag{4}$$

式中：E_t为每单位体积流体动能与内能e之和；q_x、q_y分别为x方向和y方向的热传导矢量。

用向量形式表示的控制方程更适合进行数值计算，对这些方程进行汇总，发现它们有相同的通用形式：

$$\frac{\partial U}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=0 \tag{5}$$

其中，U、E、F都为列向量，这里仅展示U的表示方式，E和F列向量的表示方式类似。U表示解向量，U的分量表示为

$$U=(\rho,\rho u,\rho v,E_t)^{\mathrm{T}} \tag{6}$$

1.2　数值计算

本文对控制方程进行有限差分方法离散，采用MacCormack数值计算方法进行求解。MacCormack方法在时间和空间上都具有二阶精度，主要由预估步和校正步两个步骤构成。

在t时刻，通过每个网格点处的流场数值可以计算出这些网格点在$t+\Delta t$时刻的流场变量值。如图1所示，节点i与节点$i+1$之间的空间步长为Δx，节点j与节点$j+1$之间的空间步长为Δy。

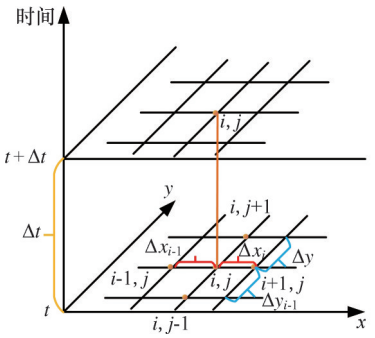


图1　时间推进网格

Fig.1　Time-progressing grid

为了实现这一目标，本文采用以下步骤。

1)预估步。在t时刻，根据已知的流场数值对空间导数进行向前差分，得到$t+\Delta t$时刻节点(i,j)预估值，其表达式为

$$\overline{U}_{i,j}^{(t+\Delta t)}=U_{i,j}^{(t)}-\Delta t(E_{i+1,j}^{(t)}-E_{i,j}^{(t)})/\Delta x-\Delta t(F_{i,j+1}^{(t)}-F_{i,j}^{(t)})/\Delta y \tag{7}$$

式中：$\overline{U}$为U的预估值。

2)校正步。对预估值进行解码计算出流场变量p、u、v等，并计算列向量E、F的预估值，将其代入式(5)中对空间导数进行向后差分，得到$t+\Delta t$时刻流场变量的校正值，其表达式为

$$U_{i,j}^{(t+\Delta t)}=\frac{1}{2}\left[U_{i,j}^{(t)}+\overline{U}_{i,j}^{(t+\Delta t)}-\frac{\Delta t}{\Delta x}(\overline{E}_{i,j}^{(t+\Delta t)}-\overline{E}_{i-1,j}^{(t+\Delta t)})-\frac{\Delta t}{\Delta y}(\overline{F}_{i,j}^{(t+\Delta t)}-\overline{F}_{i,j-1}^{(t+\Delta t)})\right] \tag{8}$$

式中：$\overline{E}$、$\overline{F}$分别为E和F的预估值。

3)时间推进。重复以上步骤，将预估步和校正步交替进行，直到达到所需的时间推进步数，每一步的时间间隔Δt趋近于某一个值，即便再进行时间推进也几乎没有变化，即所求的定常解。约束条件为CFL(Courant-Friedrichs-Lowry)准则，其表达式为

$$\Delta t_{i,j}=K\left[\frac{|u_{i,j}|}{\Delta x}+\frac{|v_{i,j}|}{\Delta y}+a_{i,j}\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}}+2v'_{i,j}\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}\right)\right]^{-1} \tag{9}$$

$$v'_{i,j}=\max\left(\frac{\frac{4}{3}(\gamma\mu_{i,j}/Pr)}{\rho_{i,j}}\right) \tag{10}$$

式中：$a_{i,j}$为声速；$u_{i,j}$、$v_{i,j}$分别为每个网格点在x方向和y方向的分量；$\Delta t_{i,j}$为每个网格点每一步的时间间隔值，应小于声波在两个网格点之间传播时间，对于每一个网格点的Δt是不相同的，为保证求解稳定，要选取其中最小的Δt；Pr为普朗特数；γ为比热容比；μ为动力黏度。

2 异构并行加速计算关键过程

2.1 CPU/GPU异构并行计算

本文采用CPU/GPU异构并行模式设计该并行算法，充分发挥CPU和GPU各自优势以提高计算效率。图2展示了CPU/GPU异构并行模式。

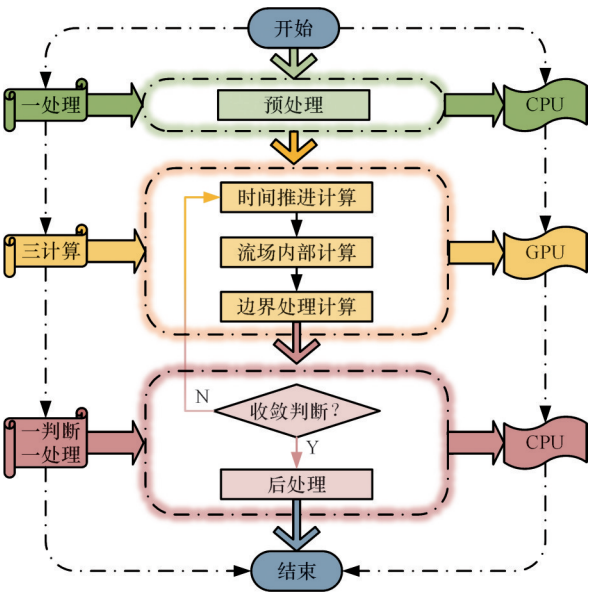


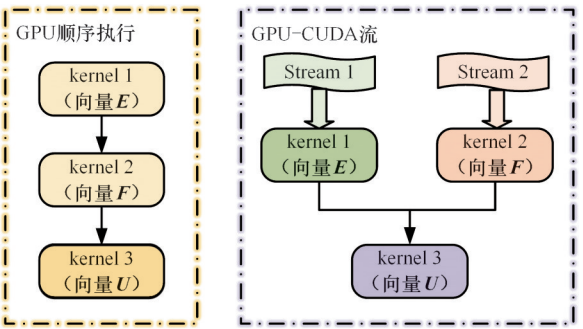
图2 CPU/GPU异构并行模式

Fig.2 CPU/GPU heterogeneous parallelism model

从图2中可以清晰地看出，在超音速流场计算中，可以将任务划分为6个模块：流场预处理、时间推进计算、流场内部计算、边界处理、收敛判断、流场后处理。在流场预处理、收敛判断以及后处理阶段，需要进行参数设置和流场初始化等逻辑处理，这些任务放置在CPU上进行。在GPU端仅进行大量流场数据的并行计算。

2.2 CUDA流并发执行及异步数据传输

CUDA流(Streams)是CUDA的一种多任务管理机制，在CUDA流的管理机制下，CPU可以发布多个计算任务，将这些计算任务放入不同的流中，由于流与流之间没有顺序执行要求，所以在GPU资源足够时这些任务可以并发执行。在设计算法时采用多流多内核执行，使并行算法的加速效果更好。例如，在MacCormack数值计算时，求解向量$\boldsymbol{U}$依赖于向量$\boldsymbol{E}$和$\boldsymbol{F}$的计算结果。但是向量$\boldsymbol{E}$和$\boldsymbol{F}$之间不存在数据依赖关系，可以并发执行。本文将向量$\boldsymbol{E}$和$\boldsymbol{F}$的计算任务分配给不同的CUDA流，每个流处理不同的任务，进一步提高MacCormack算法的加速效果。图3a为GPU顺序执行内核函数示意图，图3b所示为CUDA流异步执行。



(a) 顺序执行内核函数　　(b) CUDA流异步执行

图3 kernel函数的异步执行

Fig.3 Asynchronous execution of kernel function

由于每迭代一步都需要将计算的变量由设备端(Device)传输至主机端(Host)，传输量大，因此本文提出采用CUDA流异步执行数据传输，图4所示为CUDA流并行传输数据。从图4中可以看到，密度、压力、速度等变量传输可以通过不同CUDA流同时执行，压缩传输时间。

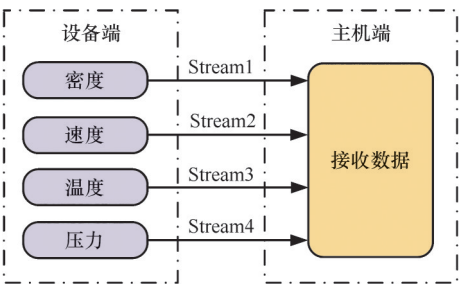


图4 CUDA流并行传输数据

Fig.4 CUDA streams for parallel data transfer

2.3 共享存储器访问方式设置

在扩大网格规模计算流场时，计算规模会相应增大。为提高计算效率，通过优化对共享存储器的访问方式来加速程序计算。图5所示为网格规模64×64网格的线程结构，可以看到，在线程块内部每次线程通信都需要通过共享存储器，因此可以通过共享存储器访问优化来加速程序运行速度。

共享内存被分为32个同样大小的内存体

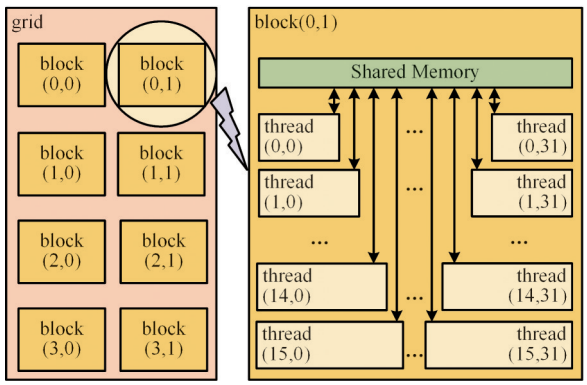


图5　网格规模64×64的线程结构

Fig.5　Thread structure with a 64×64 grid size

(banks)，对应线程束(warp)中32个线程。通过合理布置数据在存储器中的位置，可以避免bank冲突提高计算速度。例如，计算黏性项q_y，要输入来流参数温度T，并且该参数会在计算时不断应用到，将该参数从全局内存复制到共享存储器中，定义其数组大小为32×16，即

```
__shared__ double T_shared[32][16]
```

将其存储在图6所示的32个bank中，二维32×16数组的每行正好是32个数，当一个warp的32个thread同时读取一行时不会产生bank冲入。

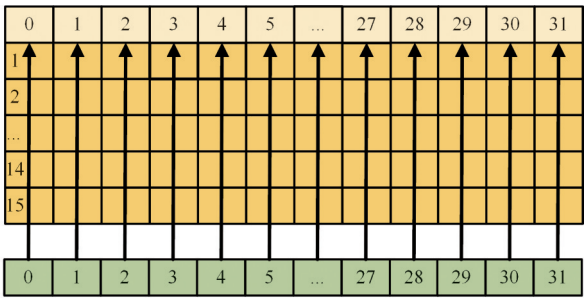


图6　共享存储器访问方式优化过程

Fig.6　Optimization process of shared memory access patterns

通过优化共享存储器访问方式可以极大提高算法的计算效率。

3　计算流体力学的GPU并行执行过程

3.1　执行内核函数参数配置

在CUDA编程中，线程被组织成线程块和网格，因此在调用GPU内核函数前必须进行参数配置来提高程序运行效率，即确定线程块数(grid)、每个线程块中的线程数(block)。本文采用数据类型dim3来表示grid和block，仅设置x、y方向，z方向默认为1。在设置block维度时应选32的倍数，设置dimBlock为(32,16)，再根据block维度及网格大小计算grid维度dimGrid。表1所示为不同计算网格内核函数参数。

表1　内核函数参数配置

Tab.1　Configuration of parameters for kernel functions

计算网格	内核配置(grid维度)	数值
64×64	(2,4)	4096
128×128	(4,8)	16384
256×256	(8,16)	65536
320×320	(10,20)	102400
512×512	(16,32)	262144

3.2　执行内核函数参数配置

在设计并行算法时，最主要的是设计GPU代码，一个完整的GPU代码包括六部分：第一部分是启动设备端，使用函数cudaSetDevice()获取可用的GPU设备；第二部分是分配GPU显存，利用函数cudaMalloc()为GPU上的数据传输分配足够的显存空间；第三部分是进行主机端到设备端数据传输，通过函数cudaMemcpy()将主机内存中的数据复制到GPU设备的显存中；第四部分是执行内核函数kernel；第五部分是将设备端的数据传输到主机，同样使用函数cudaMemcpy()；最后通过函数cudaFree()释放GPU显存。

结合以上GPU设计代码过程以及CPU/GPU异构并行模式，设计基于MacCormack的算法流程图，如图7所示。在图7中，CPU和GPU框图部分表示基于CPU/GPU异构并行模式的并行实现过程，在CPU上进行前处理后，启动GPU开始数值计算，调用函数将计算数据从CPU传递至GPU执行内核函数。之后将收敛判断所需数据由GPU传递至CPU，如果收敛，则将所有用于研究的数据全部传输至CPU端进行后处理；如果不收敛，则继续进行迭代计算。图7中kernel框图部分表示MacCormack内核函数在GPU上的执行过程。该内核函数包括时间步长计算、原始变量转换为守恒变量、预估步计算以及校正步计算。

在预估步中，首先进行各个方向的黏性应力和热流量的计算。然后，根据这些计算结果，计算出原始变量的预估值U_{1p}、U_{2p}、U_{3p}、U_{5p}。接下来，通过解码和边界条件计算出变量u、v、T、p的预估值。在校正步中，同样进行各个方向的黏性应力和热流量的计算。然后，根据计算结果和原始变量的预估值计算出原始变量的校正值U_1、U_2、U_3、U_5。最后，通过解码和边界条件计算出变量u、v、T、p的校正值。

4　算例验证及结果分析

4.1　物理模型

超音速流过平板算例是一个复杂的问题，涉

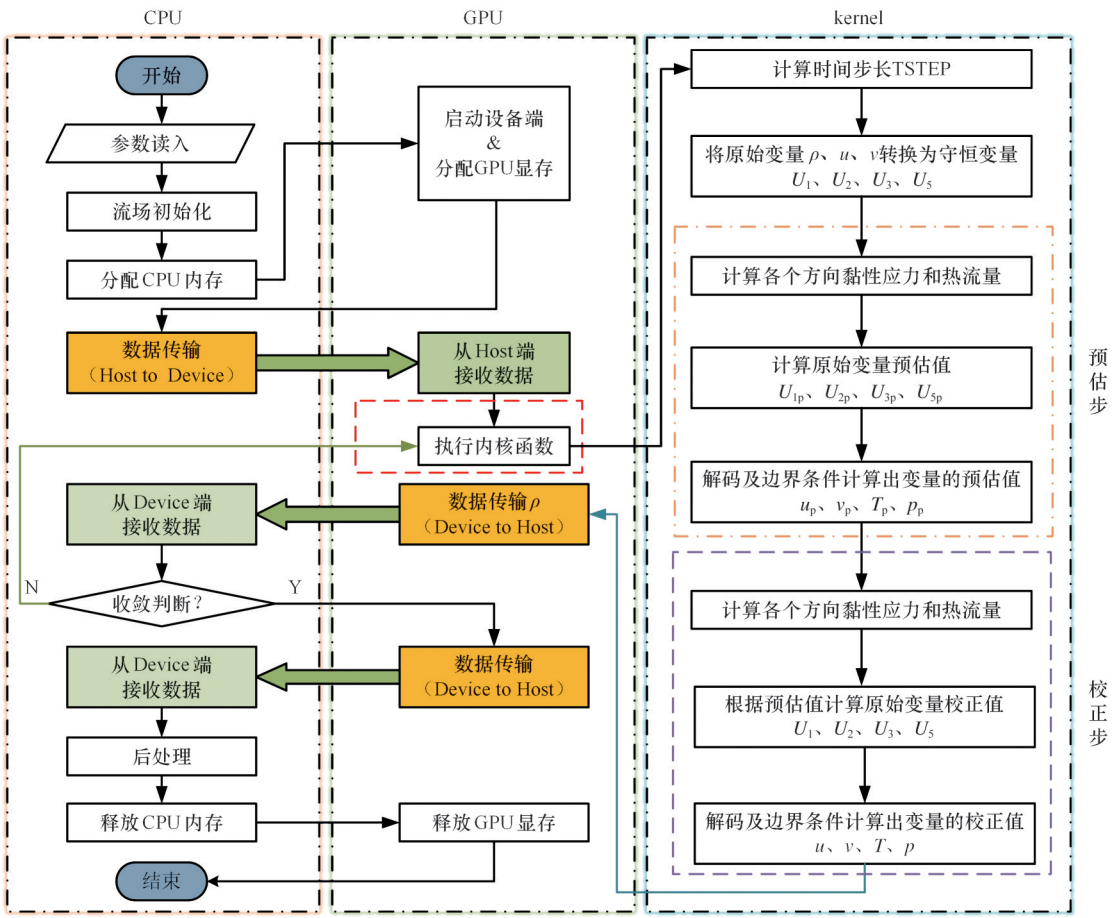


图7　基于MacCormack的并行算法设计

Fig.7　Design of parallel algorithm based on MacCormack

及流体力学、热力学和气体动力学等多个领域，主要研究的是气流以超过介质中声速的速度运动时平板上方的气流行为。在超音速流动中，任何扰动都表现为波的形式，激波就是由于超声速流体速度突变而形成的一种不连续波动界面。这种现象在实际应用中非常重要，例如在设计超声速飞行器时，需要考虑激波和膨胀波等现象对飞行器性能的影响。本文提出的超音速单平板物理模型如图8所示。在图8中，一个长度为L的尖前缘薄平板在攻角为零的状态下进行超声速绕流，平板前缘出现黏性边界层使平板产生了弯曲，从远处观看该平板就带有一定曲率的激波。图中压力p、温度T、声速a、马赫数Ma均为来流参数。图中用1、2标出了两个不同的流向位置，本文将对两个站位的流场进行分析。

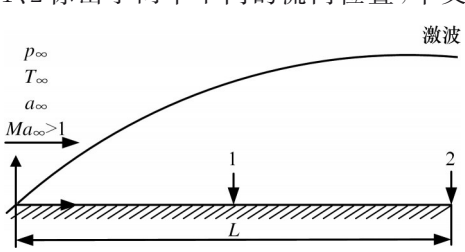


图8　超音速单平板物理模型

Fig.8　Physical model of supersonic single flat plate

4.2　流场边界条件及初始参数条件

本文所研究和计算的是超音速单平板流场，气流从平板上方流过。如图9所示，本文设定的计算域长度为平板长度L，高度H为五倍的边界层厚度。该计算域的边界条件包括：①入口边界条件分别给定来流气体x方向速度u、气体压力p以及温度T，y方向的速度v为0；②上边界条件与入口边界设定一致；③物面边界条件为壁面无滑移条件，即$u=0$，$v=0$，$T=T_w$（其中T_w为壁面温

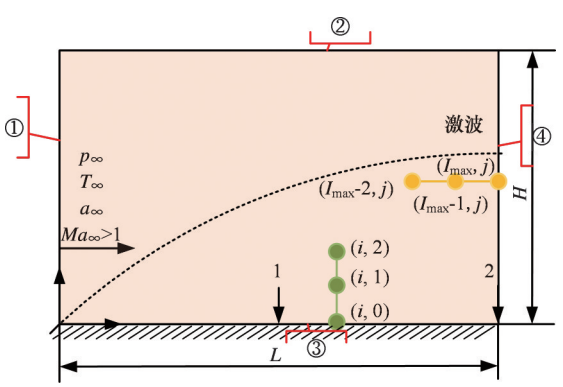


图9　计算域示意图

Fig.9　Diagram of computational domain

度），压力p由外推法计算，即$p(i,0)=2p(i,1)-p(i,2)$；④出口边界u、v、p、T均由外推法得到，u的计算公式为$u(I_{max},j)=2u(I_{max}-1,j)-u(I_{max}-2,j)$，其中，$I_{max}$为$x$方向最大网格数。

为了求解流场结果，需要从初始条件开始推进，因此，需要设置流场每个网格点在初始时刻的参数，如表2所示。

表2　流场初始参数

Tab.2　Initial parameters of the flow field

参数	数值
平板长度L/μm	10
声速a_∞/(m·s^{-1})	340.28
压力p_∞/Pa	101 325.0
温度T_∞/K	288.16
比热容比γ	1.4
普朗特常数Pr	0.71
理想气体常数R/(J·kg^{-1}·K^{-1})	287

4.3　流场计算结果对比

首先将基于70×70网格的CPU计算结果与文献[17]中平板表面的压力分布结果进行对比。本文在进行流场分析时，对来流参数量纲一化，例如p/p_∞、u/u_∞。由图10可以看出，本文的CPU计算结果与文献[17]的等温壁物面压力分布计算结果基本一致，从而证明了本文的CPU程序计算结果是正确的。

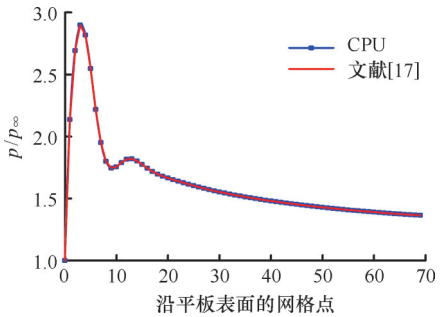


图10　CPU结果正确性对比图

Fig.10　CPU result correctness comparison diagram

图11为CPU与GPU的流场温度等值线分布图。通过对比可知，GPU和CPU都能够精确地模拟出流场内部的变化情况，都显示出了高度的计算准确性和细致的描绘能力。

采用边界层坐标绘制剖面图，定义如下：

$$\bar{y}=\frac{y}{x}\sqrt{Re} \tag{11}$$

其中，x方向上均匀分布的坐标以步长Δx递增，直到平板长度L；y方向上均匀分布的坐标以步长Δy递增，直到计算域高度H；Re为雷诺数。

从图12中可以看出，无论是GPU还是CPU，都能准确反映平板站位1和站位2的速度变化情况，两者都成功地捕捉到了平板速度的突变情况，并且GPU和CPU的计算结果大体一致。

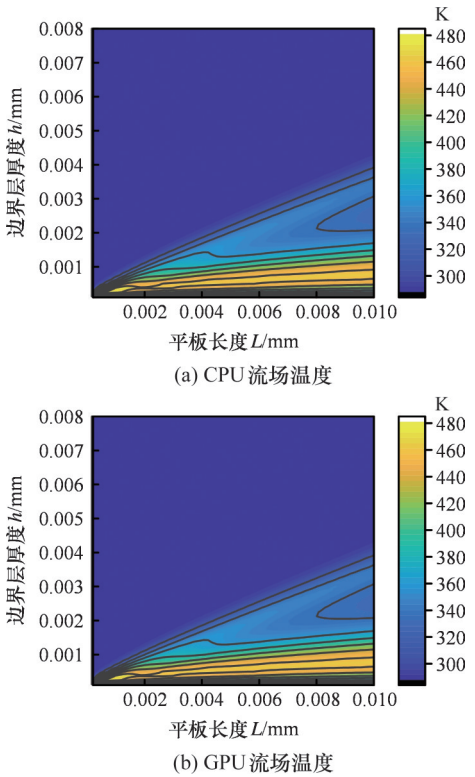


图11　CPU与GPU的流场温度等值线分布图

Fig.11　Flow field temperature contour distribution diagram of CPU and GPU

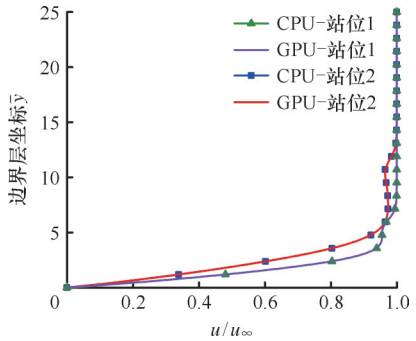


图12　GPU与CPU平板速度计算结果

Fig.12　Flat plate velocity calculation results of GPU and CPU

图13所示为GPU和CPU在压力剖面站位1和站位2的计算结果。图13清晰地展示了GPU和CPU对平板边界层的压力，并且可以观察到两者的计算结果几乎完全一致。

对计算结果进行对比后发现，GPU与CPU的计算结果基本一致，但是仅从肉眼来判断其结果的一致性不够严谨，本文以CPU的计算结果为基准计算不同网格规模的流场计算偏差，如表3所示。可以看出，两者的平均误差都在1.0×10^{-7}以下，最大误差都在1.0×10^{-4}以下，因此GPU可以代替CPU进行数值计算。

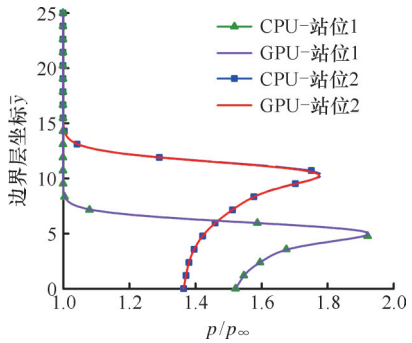


图13 GPU与CPU压力剖面计算结果

Fig.13 Pressure profile calculation results of GPU and CPU

表3 CPU与GPU压力计算结果偏差

Tab.3 Deviation of pressure calculation results of CPU and GPU

计算网格	最大偏差/Pa	平均偏差/Pa
64×64	4.327×10^{-6}	3.523×10^{-9}
128×128	6.534×10^{-6}	5.637×10^{-9}
256×256	5.247×10^{-6}	4.823×10^{-9}
320×320	2.547×10^{-5}	7.654×10^{-8}
512×512	7.245×10^{-5}	5.852×10^{-8}

4.4 加速效果分析

本文采用CUDA-GPU代替CPU进行数值计算求解超音速平板绕流问题，目的是提高程序计算速率。对比64×64网格采用CPU和GPU计算的迭代过程，如图14所示。结果显示，在收敛曲线一致且迭代步数相同的情况下，使用GPU进行计算的收敛速度明显优于CPU的计算收敛速度，所用时间短。

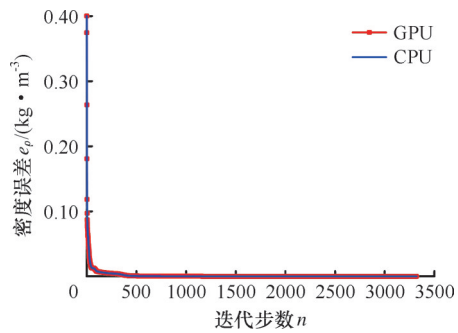


图14 GPU与CPU收敛过程迭代步

Fig.14 Iteration steps in convergence process of GPU and CPU

图15所示为GPU与CPU收敛过程迭代用时，可以看到，在迭代步数以及收敛曲线一致的情况下，使用GPU平台进行计算的收敛速度显著优于传统的CPU计算收敛速度，用时更短。这一结果表明GPU在处理大规模并行计算任务时的优越性。

表4所示为不同网格规模下GPU与CPU的流场计算时间，以及GPU并行程序的加速比。可观察到，在所有测试的网格规模中，GPU的计算时间都显著少于CPU的计算时间。随着网格规模的增大，GPU的加速比也随之提高。在较小的网格规模下，GPU中只有少数计算单元参与运算，大部分计算单元处于空闲状态，因此获得的加速比较小。当网格规模增大时，可以充分发挥GPU计算潜力，从而大幅缩短计算时间，展示了GPU在处理大规模并行任务方面的优势。

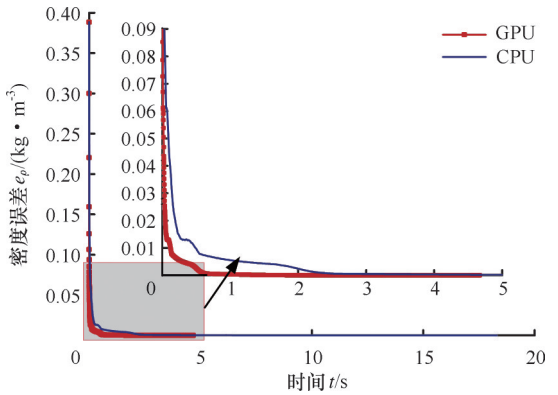


图15 GPU与CPU收敛过程迭代用时

Fig.15 Iteration time in convergence process of GPU and CPU

表4 CPU与GPU计算时间及加速比

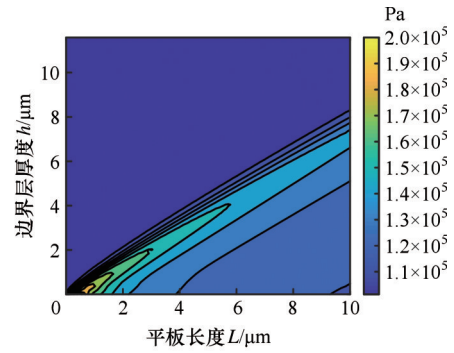
Tab.4 CPU and GPU computation time and acceleration ratio

计算网格	进程数	CPU用时/s	GPU用时/s	加速比
64×64	4096	18.32	4.68	3.91
128×128	16384	160.39	30.75	5.22
256×256	65536	1506.01	247.90	6.08
320×320	102400	4486.02	573.11	7.83
512×512	262144	30 238.02	2988.18	10.12

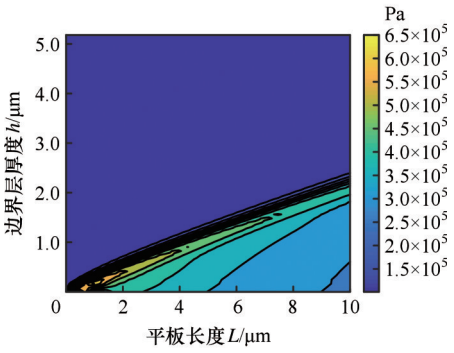
4.5 马赫数对GPU计算结果的影响

图16为网格规模64×64的不同马赫数的压力等值线分布图。由图16a（Ma=2）和图16b（Ma=10）可知，两者都清晰地捕捉到激波，Ma=2时在边界层坐标8 μm处达到最大，Ma=10时在边界层坐标2 μm处达到最大，并且激波更加靠近平板。

增大马赫数时激波不断靠近平板表面，平板表面的压力也会发生变化。图17是在网格规模不变的条件下，增大马赫数时平板表面压力计算结果对比图。可以看出，随着马赫数的增大，平板表面的压力与标准压力的比值也会增大。这一趋势表明，在高速流动中，激波的形成会导致气体流动的不连续性，从而在激波前后形成较大的压力梯度。随着马赫数的增大，这种压力梯度会变得更为明显，进而导致平板表面的压力增大。因此



(a) $Ma=2$



(b) $Ma=10$

图16　不同马赫数的压力等值线分布图

Fig.16　Pressure contour distribution at different Mach numbers

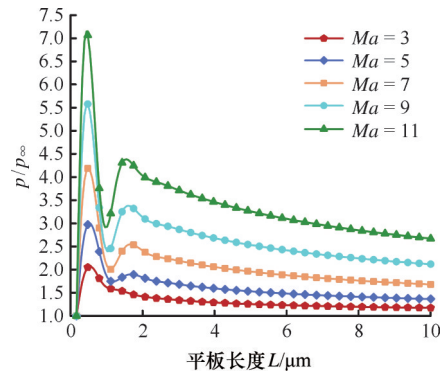


图17　不同马赫数时平板表面压力的计算结果

Fig.17　Calculation results of surface pressure on the flat plate at different Mach numbers

得出结论：随着马赫数的增大，平板表面的压力与标准压力的比值会逐渐增大。

5　结论

1）在GPU平台上实现的计算结果与CPU一致，充分验证了GPU计算结果的准确性，同时表明GPU平台具有良好的稳定性和可靠性。

2）采用GPU并行计算不仅能够充分利用硬件的并行计算能力，还能够显著加速计算过程，节省时间和成本，因此，在处理大规模计算任务时，推荐使用GPU并行计算来获得更优的计算效率。

3）随着马赫数的增大，激波会越来越靠近平板。这是因为在高速流动中，气体的压缩性效应更加显著，导致激波的形成位置向物体表面靠近，压力梯度变得更明显，从而增大平板表面压力。

参考文献：

[1] 卢志伟，张君安，刘波．多孔集成节流空气静压轴承数值计算与性能研究[J]．兵工学报，2019，40(10)：2151-2160.
LU Zhiwei，ZHANG Jun'an，LIU Bo. Numerical Calculation and Performance Study of Aerostatic Bearing with Multi-hole Integrated Restrictor[J]. Acta Armamentarii，2019，40(10)：2151-2160.

[2] 刘深深，罗磊，韩青华，等．动量增升高升阻比飞行器横航向稳定性研究[J]．北京航空航天大学学报，2023，49(11)：3010-3021.
LIU Shenshen，LUO Lei，HAN Qinghua，et al. Study on Lateral-directional Stability of a Practical High Lift-to-drag Ratio Hypersonic Vehicle with Momentum Lift Augmentation[J]. Journal of Beijing University of Aeronautics and Astronautics，2023，49(11)：3010-3021.

[3] 许翔，张艺伦，梅铮，等．汽车环境舱流场的数值模拟与实验研究[J]．中国机械工程，2023，34(17)：2115-2123.
XU Xiang，ZHANG Yilun，MEI Zheng，et al. Numerical Simulation and Experimental Investigation of Flow Fields in Vehicle Climatic Chambers[J]. China Mechanical Engineering，2023，34(17)：2115-2123.

[4] 张峰，翟季冬，陈政，等．面向异构融合处理器的性能分析、优化及应用综述[J]．软件学报，2020，31(8)：2603-2624.
ZHANG Feng，ZHAI Jidong，CHEN Zheng，et al. Survey on Performance Analysis，Optimization，and Applications of Heterogeneous Fusion Processors[J]. Journal of Software，2020，31(8)：2603-2624.

[5] LIU Xu，SUN Mingbo，WANG Hongbo，et al. A Heterogeneous Parallel Algorithm for Euler-Lagrange Simulations of Liquid in Supersonic Flow[J]. Applied Sciences，2023，13(20)：11202.

[6] 翁跃，张献伟，张曦，等．面向计算流体力学的图形处理器资源管理[J]．国防科技大学学报，2022，44(5)：35-44.
WENG Yue，ZHANG Xianwei，ZHANG Xi，et al. Graphics Processing Unit Resource Management for Computational Fluid Dynamics[J]. Journal of National University of Defense Technology，2022，44(5)：35-44.

[7] PISCAGLIA F，GHIOLDI F. GPU Acceleration of CFD Simulations in OpenFOAM[J]. Aerospace，

2023，10(9)：792.

[8] 徐坤浩，聂铁铮，申德荣，等. 基于CPU-GPU异构体系结构的并行字符串相似性连接方法[J]. 计算机研究与发展，2021，58(3)：598-608.
XU Kunhao，NIE Tiezheng，SHEN Derong，et al. Parallel String Similarity Join Approach Based on CPU-GPU Heterogeneous Architecture[J]. Journal of Computer Research and Development，2021，58(3)：598-608.

[9] 陈玥丹，肖国庆，阳王东，等. 基于异构系统的多级并行稀疏张量向量乘算法[J]. 计算机学报，2024，47(2)：441-455.
CHEN Yuedan，XIAO Guoqing，YANG Wangdong，et al. Exploiting Hierarchical Parallelism for Sparse Tensor-vector Multiplication on Heterogeneous Parallel Systems[J]. Chinese Journal of Computers，2024，47(2)：441-455.

[10] 杨峰，罗世杰，杨江鸿，等. 基于GPU加速的等几何拓扑优化高效多重网格求解方法[J]. 中国机械工程，2024，35(4)：602-613.
YANG Feng，LUO Shijie，YANG Jianghong，et al. A GPU-accelerated High-efficient Multi-grid Algorithm for ITO[J]. China Mechanical Engineering，2024，35(4)：602-613.

[11] 黄国如，陈志威，曾博威. 城市洪涝模型及CPU-GPU异构并行计算技术研究进展[J]. 水利学报，2023，54(6)：654-665.
HUANG Guoru，CHEN Zhiwei，ZENG Bowei. Research Progress of Urban Flood Model and CPU-GPU Heterogeneous Parallel Computing Technology[J]. Journal of Hydraulic Engineering，2023，54(6)：654-665.

[12] 郑勇，芦鞟，马永强，等. 基于CUDA技术的先进组件中子学程序异构并行研究[J]. 核动力工程，2021，42(增刊2)：124-129.
ZHENG Yong，LU Wei，MA Yongqiang，et al. Study on CUDA-based Heterogeneous Parallel for Advanced Assembly Neutronics Program[J]. Nuclear Power Engineering，2021，42(S2)：124-129.

[13] 张健，李瑞田，邓亮，等. 面向多核CPU/众核GPU架构的非结构CFD共享内存并行计算技术[J]. 航空学报，2024，45(7)：128888.
ZHANG Jian，LI Ruitian，DENG Liang，et al. Shared-memory Parallelization Technology of Unstructured CFD Solver for Multi-core CPU/Many-core GPU Architecture[J]. Acta Aeronautica et Astronautica Sinica，2024，45(7)：128888.

[14] YE Chuangchao，ZHANG P J Y，WAN Zhenhua，et al. Accelerating CFD Simulation with High Order Finite Difference Method on Curvilinear Coordinates for Modern GPU Clusters[J]. Advances in Aerodynamics，2022，4(1)：7.

[15] GHIOLDI F，PISCAGLIA F. Acceleration of Supersonic/Hypersonic Reactive CFD Simulations via Heterogeneous CPU-GPU Supercomputing [J]. Computers & Fluids，2023，266：106041.

[16] 张东飞，高军辉. GPU加速高阶谱差分方法在风扇噪声中的应用[J]. 航空学报，2024，45(8)：128941.
ZHANG Dongfei，GAO Junhui. Application of GPU-accelerated High-order Spectral Difference Method in Fan Noise[J]. Acta Aeronautica et Astronautica Sinica，2024，45(8)：128941.

[17] ANDERSON J D. 计算流体力学基础及其应用[M]. 吴颂平，刘赵淼，译. 北京：机械工业出版社，2007.
ANDERSON J D. Computational Fluid Dynamics[M]. WU Songping，LIU Zhaomiao，trans. Beijing：Mechanical Industry Press，2007.

（编辑 王旻玥）

作者简介：卢志伟*（通信作者），男，1979年生，副教授、博士研究生导师。研究方向为流体润滑与并行计算。E-mail：luzhiwei@xatu.edu.cn。

本文引用格式：
卢志伟，张皓茹，刘锡尧，等. 基于CPU-GPU的超音速流场N-S方程数值模拟[J]. 中国机械工程，2025，36(9)：1942-1950.
LU Zhiwei，ZHANG Haoru，LIU Xiyao，et al. Numerical Simulation of N-S Equations for Supersonic Flow Fields Based on CPU-GPU[J]. China Mechanical Engineering，2025，36(9)：1942-1950.