

# 基于 CFD 系统辨识的气弹分析及 GPU 并行算法初探\*

黄灿<sup>†</sup> 赵永辉

(南京航空航天大学机械结构力学及控制国家重点实验室, 南京 210016)

**摘要** 通过求解 Euler 方程获得运动翼段的非定常流场,并用 CUDA 语言对流场求解器进行 GPU 并行计算. 使用 ARMA(auto-regressive-moving-average)模型对非定常气动力进行辨识,由系统辨识模型得到的结果与全阶 CFD 计算结果十分吻合. 基于降阶气动模型与结构的耦合,计算了具有 S 型颤振边界的气动弹性标准算例-Isogai Wing 的跨音速颤振. 本文给出的方法可以在保证气动弹性计算精度的前提下大幅提高计算效率.

**关键词** Euler 方程, ARMA 模型, 系统辨识, 气动弹性, 并行计算

DOI: 10.6052/1672-6553-2014-010

## 引言

颤振是飞行器气动弹性问题中的重要课题. 传统气弹研究在亚声速上使用涡格法和偶极子网格法等频域方法可以得到较为精确的气动力解. 而在跨声速时,以上方法不再适用. 非定常 CFD 的方法可以进行大迎角、跨声速等非线性计算,在跨声速领域目前只有 CFD 方法计算最为精确<sup>[1]</sup>. 然而, CFD 计算所耗时间非常长,计算效率十分低下.

针对 CFD 计算离散化的特点,利用 GPU 并行计算可以天然地将任务划分成多份同时进行,大大的提高计算效率. GPU 并行计算是近些年在计算领域快速发展的一项技术. GPU 与 CPU 相比有着更强的单精度浮点计算能力和更宽的内存带宽<sup>[2-3]</sup>. 在浮点运算上, GPU 比 CPU 有着相当大的优势,同时期的 GPU 单精度浮点计算能力可达 CPU 的十倍. CUDA 语言是 GPU 上的语言,可以通过 CUDA 命令调用 GPU 资源进行计算,实现计算的并行化<sup>[4-10]</sup>.

近些年系统识别也被应用到 CFD 计算中来,作为提高计算效率的另一种途径. 目前有多种模型降阶(ROM, reduced-order model)方法,包括频域模型、时域自回归滑动平均模型(ARMA, auto-regressive-moving-average)和离散时间状态空间模型等<sup>[11-13]</sup>. CFD 技术属于时域方法,本文选择 AR-

MA 模型来进行气动力建模. 建立精确的非定常时域气动力降阶模型可以绕过繁杂的 CFD 流场求解器,直接根据输入输出关系得出想要的结果,这样可以大大降低计算所耗的时间. 本文将这两种提高气动弹性分析效率的方法结合到了一起. 首先利用 GPU 进行并行 CFD 计算,得到非定常气动力. 然后利用非定常气动力识别出降阶模型. 最后对降阶模型的精度进行了考核.

## 1 计算方法

基于 CFD 系统辨识的气动弹性分析需要的流程如图 1 所示. 其中非定常求解器求解所耗时间最多,本文将对这一部分进行并行化处理以提高效率.

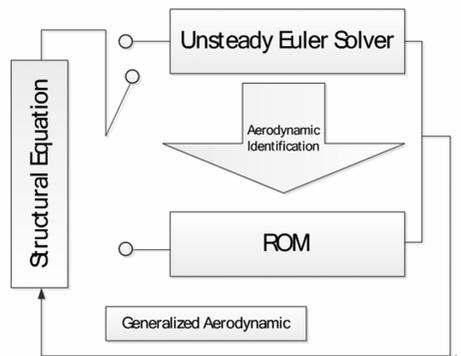


图 1 基于气动力辨识技术的气动弹性仿真流程图  
Fig. 1 Flow chart of aero-elastic simulation based on the identification of aerodynamic force

2013-10-22 收到第 1 稿, 2014-02-19 收到修改稿.

\* 国家自然科学基金资助项目(10972100)

<sup>†</sup> 通讯作者 E-mail: 772704776@qq.com

另外,对此部分进行系统辨识,得到降阶的气动力模型,再次提高计算效率.需要注意的是,不同的初始条件下辨识得到的非定常气动力降阶模型只能使用一次,初始条件改变则需要重新识别.例如,在广义气动力的计算中,不同马赫数的情况需要对应不同的降阶模型.

CFD 计算部分采用 Euler 方程,守恒形式的 Euler 方程如下:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W} d\Omega + \int_s (\mathbf{F} dy - \mathbf{G} dx) = 0 \quad (1)$$

其中  $x$  和  $y$  为笛卡尔坐标系坐标,  $\mathbf{W}$  为守恒变量:

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho U \\ \rho V \\ \rho E \end{bmatrix}$$

$\mathbf{F}, \mathbf{G}$  表示通量:

$$\mathbf{F} = \begin{bmatrix} \rho U \\ \rho U^2 + P \\ \rho UV \\ \rho UH \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho V \\ \rho UV \\ \rho V^2 + P \\ \rho VH \end{bmatrix}$$

其中,  $\rho, P, H$  和  $E$  分别表示密度、压强、单元总焓和单元总能量.  $U, V$  分别表示笛卡尔坐标系下向  $x$  和  $y$  向的速度矢量. 这些量由理想气体的单位体积的总能量  $E$  和总焓  $H$  互相联系,式中  $\gamma$  为比热比.

$$\rho E = P / (\gamma - 1) + \rho (U^2 + V^2) / 2 \quad (2)$$

$$\rho H = \rho E + P \quad (3)$$

本文采用 Jameson 中心格式的有限体积法进行空间离散,利用双时间推进法进行时间离散.物理时间采用显示格式,拟时间采用四步龙格-库塔推进格式.网格使用的是非结构网格,动网格技术使用了弹簧法<sup>[14-18]</sup>.

气动弹性控制方程为:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F} \quad (4)$$

式中  $\mathbf{M}$  为质量矩阵,  $\mathbf{K}$  为刚度矩阵,  $\mathbf{F}$  为广义气动力向量,  $\mathbf{q}$  为广义位移向量,它们分别可以表示为:

$$\mathbf{M} = \begin{bmatrix} 1 & x_{\alpha} \\ x_{\alpha} & r_{\alpha}^2 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} (\omega_h / \omega_{\alpha})^2 & 0 \\ 0 & r_{\alpha}^2 \end{bmatrix},$$

$$\mathbf{F} = \frac{1}{\pi k_c^2 \mu} \begin{Bmatrix} -C_l \\ 2C_m \end{Bmatrix} = \frac{1}{\pi} V_f^{*2} \begin{Bmatrix} -C_l \\ 2C_m \end{Bmatrix},$$

$$\mathbf{q} = \begin{Bmatrix} h/b \\ \alpha \end{Bmatrix}, k_c = \omega_{\alpha} \cdot b / V_{\infty}, \mu = \frac{m}{\pi \rho b^2}$$

其中  $\mu$  为质量比,  $C_l$  和  $C_m$  分别为升力系数和力矩系数,  $r_{\alpha}$  为机翼对刚心的无量纲回转半径,  $x_{\alpha}$  为重心与刚心的无量纲距离,  $\omega_{\alpha}$  和  $\omega_h$  分别为俯仰和浮沉两个自由度的固有频率,  $v_f^*$  为无量纲颤振速度.

为了便于时域求解,引入状态变量  $\mathbf{E} = (q_1, q_2, \dots, q_N, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_N)^T$ , 则方程(4)可以写成状态空间的形式:

$$\dot{\mathbf{E}} = \mathbf{f}(\mathbf{E}, t) = \mathbf{A} \cdot \mathbf{E} + \mathbf{B} \cdot \mathbf{F}(\mathbf{E}, t) \quad (5)$$

式中

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix}$$

在每个时间步内,将视为时间的单值函数,这样应用改进的龙格-库塔方法可将方程(5)展开为:

$$\begin{cases} \mathbf{E}_{n+1} = \mathbf{E}_n + (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) / 6 \\ \mathbf{k}_1 = \Delta t \cdot [\mathbf{A} \cdot \mathbf{E}_n + \mathbf{B} \cdot \mathbf{F}(t)] \\ \mathbf{k}_2 = \Delta t \cdot [\mathbf{A} \cdot (\mathbf{E}_n + \mathbf{k}_1 / 2) + \mathbf{B} \cdot \mathbf{F}(t + \Delta t / 2)] \\ \mathbf{k}_3 = \Delta t \cdot [\mathbf{A} \cdot (\mathbf{E}_n + \mathbf{k}_2 / 2) + \mathbf{B} \cdot \mathbf{F}(t + \Delta t / 2)] \\ \mathbf{k}_4 = \Delta t \cdot [\mathbf{A} \cdot (\mathbf{E}_n + \mathbf{k}_3) + \mathbf{B} \cdot \mathbf{F}(t + \Delta t)] \end{cases} \quad (6)$$

上式后几步中的  $\mathbf{F}(t + \Delta t / 2)$  和  $\mathbf{F}(t + \Delta t)$  可用前几个时刻 ( $\mathbf{F}(t), \mathbf{F}(t - \Delta t), \mathbf{F}(t - 2\Delta t)$ ) 的值插值得到.该方法计算所得的精度要高于传统的冻结气动力的方法,而且在效率上大大超过标准的龙格-库塔方法.

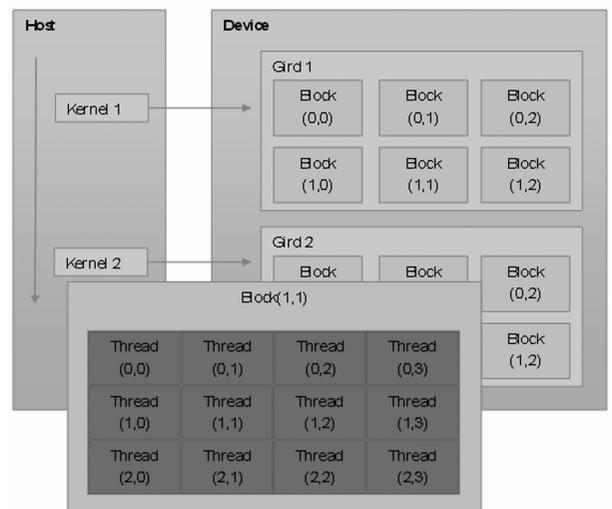


图2 GPU 任务划分原理图

Fig. 2 The principle of division of tasks on GPU

GPU 并行计算本文应用到 CUDA 语言和 NVIDIA 的 GPU. 并行计算的原理是将程序中的循环部分拆开,分别交给  $k$  (循环次数) 个计算单元同

时进行计算,再将计算结果读回主机端. GPU 做并行计算有着天然的优势,其最小计算单元是线程, GPU 中的线程数远远大于 CPU 中的核心数. 其工作原理如图 2:

并行计算要求各线程块和线程之间的计算互不干扰,于是随时间变化的迭代过程是不能够并行化的,因此只有适合的离散模型才能用 GPU 进行并行计算. 非定常 Euler 方程中,多处进行了按单元或者按边循环的计算,这种空间离散模型在循环计算时同一时间步内边与边的计算相互独立,可以用 GPU 进行并行计算. 随着网格单元数的上升, CPU 计算时间大幅增加,而对于 GPU 的计算时间却增加很小. 理论上并行部分的计算时间几乎不变,只是网格数的增加会增加一定的数据传输时间. 于是对这些部分进行并行是可行的. 其中主要包括通量和残值计算以及动网格计算的并行化.

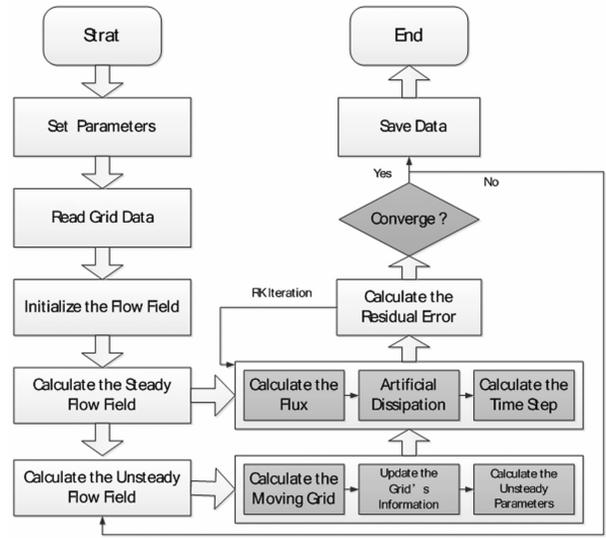


图3 CFD 计算流程

Fig. 3 Process of CFD

图3为CFD的计算流程,其中阴影部分均为可并行部分. 本文将算例的CFD计算部分并行化处理,大大提高计算效率. 本文计算的硬件环境为: Intel Core 2 Duo CPU, 2.94GHz, 4GB 内存; NVIDIA Geforce GTX550Ti, 192 个流处理器(SM), 1GB 显存. 软件环境为 Microsoft Windows7 SP1 操作系统, VS2008 编译器, CUDA 版本为 4.2.

系统识别采用 ARMA 模型. 多输入多输出的 ARMA 模型实际上是系统的离散差分模型,其表达式为:

$$y(k) = \sum_{i=1}^{na} A_i y(k-i) + \sum_{i=0}^{nb-1} B_i u(k-i) + e(k) \quad (7)$$

式中,  $y(k)$  为系统输出量(这里指广义气动力系数向量)的第  $k$  次观测值;  $y(k-1)$  为系统输出量的第  $k-1$  次观测值,以此类推;  $u(k)$  为系统第  $k$  个输入量(这里指广义位移向量);  $u(k-1)$  为系统第  $k-1$  个输入量,以此类推;  $e(k)$  为零均值的随机噪声;  $A_i$  和  $B_i$  为待辨识的参数矩阵;  $na$  和  $nb$  分别为输出和输入的延迟阶数.

设系统共采集了  $L$  组数据,即  $k=1, 2, \dots, L$ , 将式(7)化为如下方程:

$$y_L(k) = H_L(k)\theta + e_L(k) \quad (8)$$

式中:

$$y_L(k) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(L) \end{bmatrix}, \quad e_L(k) = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(L) \end{bmatrix}$$

$H_L(k) =$

$$\begin{bmatrix} -y(0) & \cdots & -y(1-na) & u(1) & \cdots & u(2-nb) \\ -y(1) & \cdots & -y(2-na) & u(2) & \cdots & u(3-nb) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(L-1) & \cdots & -y(L-na) & u(L) & \cdots & u(L+1-nb) \end{bmatrix}$$

$$\theta = [A_1 \ A_2 \ \cdots \ A_{na} \ B_0 \ B_1 \ \cdots \ B_{nb-1}]$$

根据最小二乘法,在随机噪声最小时,可得待识别参数矩阵  $\theta$  的估计为:

$$\hat{\theta} = (H_L^T \cdot H_L)^{-1} \cdot H_L^T \cdot y \quad (9)$$

本文选择了比较容易实现并且具有很宽频带宽度的“3211”速度输入. 通过调整训练信号的时间步长,可以将频带移到试验所希望激发的频带上去. 由于输入输出数据是一次性批处理的,故采用上文所述的一次性最小二乘估计,其精度较高. 参数辨识需要对系统进行稳态清零,使系统满足零输入-零输出特性.

## 2 算例

为了验证流场求解器的正确性,首先选用 NA-CA0012 翼型进行非定常气动力计算,用计算结果与实验结果对比. 以翼型的俯仰简谐振荡为例:

$$\alpha(t) = \alpha_0 + \alpha_m \sin(\omega t) \quad (10)$$

其中,初始攻角  $\alpha_0 = 0.016^\circ$ , 角度幅值  $\alpha_m = 2.51^\circ$ ,

角频率  $\omega = k \cdot V_\infty / b$ , 缩减频率  $k = 0.0814$ , 半弦长  $b = 0.5m$ ,  $Ma = 0.755$ . 计算结果如图 4 所示:

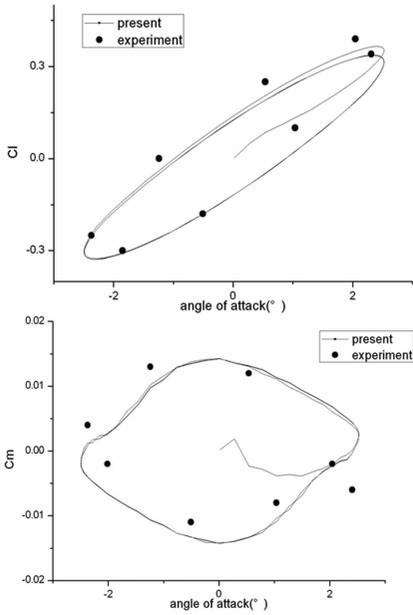


图 4 NACA0012 翼型非定常气动力系数与实验值对比  
Fig. 4 NACA0012 airfoil's unsteady aerodynamic force simulation compared with experiment

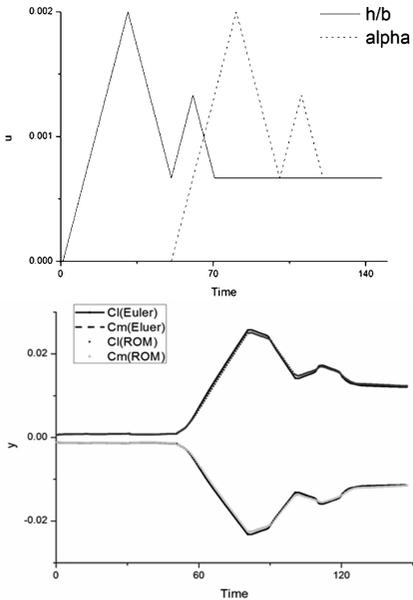


图 5 3211 信号输入下 Euler 解和降阶模型解比较  
Fig. 5 The Euler solution compared with the ROM solution on 3211 input signal

颤振计算中本文选择了 Isogai Wing, 它是二维跨声速气动弹性计算的标准算例, 其翼型为 NACA64A010 翼型. 二元翼段气动弹性系统的物理参数为:

$$\alpha = 0^\circ, a = -2.0, x_\alpha = 1.8,$$

$$r_\alpha^2 = 3.48, \omega_\alpha / \omega_h = 1.0, \mu = 60.$$

针对 NACA64A010 翼型, 本文首先使用“3211”输入信号进行激励, 通过调整输入信号的带宽和幅值将信号训练到合适的频带上来. 训练信号包含了俯仰和浮沉两个自由度的输入, 所得到的输出信号包括和两个系统输出.

图 5 中给出了“3211”信号输入下的 Euler 解和辨识模型解的比较. 在相同输入下辨识模型给出的输出结果与非定常 Euler 求解器的输出结果吻合很好. 可以看出, 该辨识模型在很宽的频带范围内对原系统都有很好的近似.

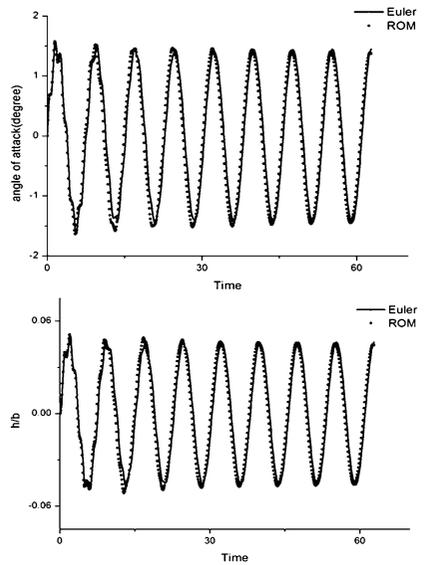


图 6 NACA64A010 翼型的颤振临界响应 ( $Ma = 0.825, v_f^* = 0.54$ )  
Fig. 6 The critical flutter response of NACA64A010 airfoil ( $Ma = 0.825, v_f^* = 0.54$ )

图 6 给出了  $Ma = 0.825, v_f^* = 0.54$  时, Euler 方程计算的俯仰和浮沉两个自由度的颤振响应与辨识模型计算的响应对比. 图中可以看出两种方法的颤振响应基本吻合.

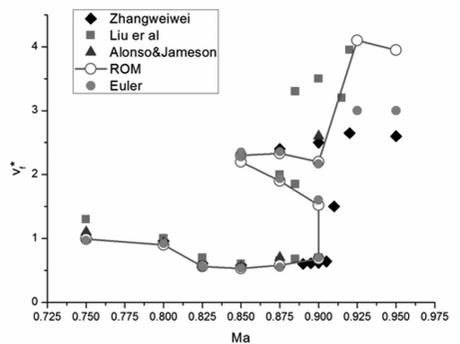


图 7 NACA64A010 翼型的颤振速度边界  
Fig. 7 The flutter velocity boundary of NACA64A010 airfoil

针对 Isogai Wing, 算例计算了其颤振速度随马赫数变化的边界. 图 7 中可以看出, 与众多文献中的结果相比, 基于非定常 Euler 方程的颤振速度边界和辨识模型计算的颤振边界在的状态下吻合的非常好.

CUDA 语言的并行程序和串行 C++ 程序计算的加速比约为 2.8 倍; ROM 降阶模型与全阶 CFD 模型计算颤振速度加速比约为 2.3 倍; 将并行程序和 ROM 降阶模型结合与全阶串行程序相比加速比约为 6.4 倍. 对于二维翼型的计算, 参考文献 [10] 中的结果来看, 其应用 CUDA 语言并行计算得到了 1.04 倍的加速比, 本文结果加速比更优.

表 1 各方法下计算与原始程序计算加速比

Table 1 The speedup of the program on different methods

Method	Speedup
CUDA parallel computing	2.8
ROM	2.3
CUDA + ROM	6.4

### 3 结论

本文结合了两种非定常 Euler 方程的加速方法: 基于 CFD 的非定常气动力采用 GPU 并行计算实现加速; 之后利用 CFD 计算结果, 得到了非定常气动力的 ARMA 模型. 仿真结果表明 ARMA 降阶模型对原非定常气动模型还原较好, 在保证了一定精度的情况下提升了计算效率. 此外, CUDA 并行程序受限于网格量并未发挥全部潜力, 但优于文献 [10] 的二维结果, 在三维问题上 GPU 并行计算对效率的提升将会非常可观. 本文将两者结合可大大减少基于 CFD 的气动弹性计算时间, 值得进一步深入研究.

### 参 考 文 献

- 张伟伟, 叶正寅. 基于非定常气动力辨识技术的气动弹性数值模拟. 航空学报, 2006, 27(4): 579 ~ 583 (Zhang W W, Ye Z Y. Numerical simulation of aeroelasticity basing on identification technology of unsteady aerodynamic loads. *Acta Aeronautica ET Astronautica Sinica*, 2006, 27(4), 579 ~ 583 (in Chinese))
- 张加乐. 基于 GPU 并行计算的非定常 Euler 方程算法研究[硕士学位论文]. 南京: 航空航天大学, 2012 (Zhang J L. Numerical studies of unsteady euler equations based on GPU parallel computing [Master Thesis]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2012 (in Chinese))
- 苗树明. NS 方程在 GPU 上的并行实现[硕士学位论文]. 上海: 上海交通大学, 2011 (Miao S M. Implementation of NS equations in parallel on GPU [Master Thesis]. Shanghai: Shanghai Jiao Tong University, 2011 (in Chinese))
- Corrigan A, Camelli F, Lohner R, Wallin J. Running unstructured grid based CFD solvers in modern graphics hardware. *AIAA*, 2009; 4001
- Antoniou A S, Karantasis K I, Polychronopoulos E D. Acceleration of a finite difference WENO scheme for large-scale simulations on many-core architectures. *AIAA*, 2010; 525
- Brandvik T, Pullan G. Acceleration of a 3D euler solver using commodity graphics hardware. *AIAA*, 2008; 607
- Shinn A F, Vanka S P, Hwu W W. Direct numerical simulation of turbulent flow in a square duct using a graphics processing unit (GPU). *AIAA*, 2010; 5029
- Elsen E, LeGresley P, Darve E. Large calculation of the flow over a hypersonic vehicle using a GPU. *Journal of Computational Physics*, 2008, 227: 10148 ~ 10161
- Hargen T R, Lie K A, Natvig J R. Solving the euler equations on graphic processing units. *Computational Science*, 2006, 3994: 220 ~ 227
- 张兵, 韩景龙. 基于 GPU 和隐式格式的 CFD 并行计算方法. 航空学报, 2010, 2: 249 ~ 256 (Zhang B, Han J L. Parallel computing methods for CFD using a GPU and implicit scheme. *Acta Aeronautica ET Astronautica Sinica*, 2010, 2: 249 ~ 256 (in Chinese))
- Daniella E Raveh. Identification of computational-fluid-dynamics based unsteady aerodynamic models for aeroelastic analysis. *Journal of Aircraft*, 2004, 41(3): 620 ~ 632
- Gupta K K, Bach C. Systems identification approach for a computational-fluid-dynamics based aeroelastic analysis. *AIAA*, 2007, 45(12): 2820 ~ 2827
- Lai K L, Won K S, Koh E P C, Tsai H M. Flutter simulation and prediction with CFD-based reduced-order model. *AIAA*, 2012; 2006 ~ 2026
- Stolcis L, Johnston L J. Solution of the euler equations on unstructured grids for two-dimensional compressible flow. *Aeronautical Journal*, 1990; 181 ~ 195
- John T Batina. Implicit flux-split euler schemes for unsteady aerodynamic analysis involving unstructured dynamic

- meshes. *AIAA*, 1991, 29(11):1836 ~ 1843
- 16 邓枫, 伍贻兆, 刘学强. 基于混合动网格的二维非定常粘性流动数值模拟. *南京航空航天大学学报*, 2007, 39(4):444 ~ 448 (Deng F, Wu Y Z, Liu X Q. Numerical simulation of two-dimensional unsteady viscous flow based on hybrid dynamic grids. *Journal of Nanjing University of Aeronautics & Astronautics*, 2007, 39(4):444 ~ 448 (in Chinese))
- 17 王军利, 白俊强, 詹浩. 基于非结构动网格的非定常气动力计算. *飞机设计* 2005, 9:24 ~ 29 (Wang J L, Bai J Q, Zhan H. Calculation of unsteady flow using dynamic unstructured grids. *Aircraft Design*, 2005, 9:24 ~ 29 (in Chinese))
- 18 Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. 10<sup>th</sup> *AIAA Computational Fluid Dynamics Conference*, 1991: 1596

## IDENTIFICATION OF CFD-BASED AEROELASTIC ANALYSIS AND GPU PARALLEL COMPUTING \*

Huang Can<sup>†</sup> Zhao Yonghui

(State Key Laboratory of Mechanics and Control of Mechanical Structures, Nanjing University of Aeronautics and Astronautics,  
Nanjing 210016, China)

**Abstract** The unsteady flow field around a moving airfoil section was obtained by solving the Euler equation. The CUDA programming language was used to provide a parallel computing capacity on GPU. The ARMA (auto-regressive-moving-average) model was used to identify the unsteady aerodynamic forces, and simulation results using the identified aerodynamic model of the system identification agree well with those using the full ordered CFD model. Furthermore, the transonic flutter of the Isogai Wing with an Stype flutter boundary, which is the standard example in aero-elasticity, was calculated. The approach presented can greatly improve the efficiency of computing without loss of accuracy.

**Key words** Euler equation, ARMA model, system identification, aero-elasticity, parallel computing