Efficient high Performance Computing Techniques for Multi-disciplinary Optimisation

Mohamed Hassanine Aissa

Turbomachinery and Propulsion Department, von Karman Institute for Fluid Dynamics, Belgium, aissa@vki.ac.be

Supervisor: Tom Verstraete

Assistant Professor, Turbomachinery and Propulsion Department, von Karman Institute for Fluid Dynamics, Belgium, tom.verstraete@vki.ac.be.

Abstract

The multidisciplinary design optimization (MDO) of turbomachinery components relies nowadays on performance estimations through simulation tools. These tools have to include mainly: aerodynamical, structural and thermodynamical considerations. The accuracy of the multidisciplinary aspect requires a high computational power available nowadays only by High-Performance-Computing systems (HPC).
One of the newest HPC hardware and most powerful shared-memory systems is Graphics processing Unit (GPU). Its actual widespread is encouraged by a higher computational power and limited by a different hardware architecture compared to conventional Central Processing Units (CPU). The aim of this work is therefore to utilize the GPU computational power to develop and validate one of the firsts viable GPU code for multidisciplinary design optimization on turbomachinery application. The current step is to boost the time consuming CFD solver - Euler and Navier-Stokes - of the in-house MDO system by adapting it for GPU. A second target envelops a larger use of GPU on the MDO algorithm. Preliminary results have been achieved with the verification and porting to GPU of the Euler version of the in-house CFD solver with explicit time stepping. The GPU saved 15% of the simulation run-time. The preexisting CPU-based Navier-Stokes solver has been validated against the VKI LS89 turbine guide vane. These first results pave the way for the porting and optimization of the full CFD simulation and therefore achieving better overall speedups.

Keywords: Multidisciplinary design optimization, graphics processing units, RANS Navier-stokes , Turbomachinery, LS89 turbine , GAMM test case,

1. Introduction

Multidisciplinary Design Optimization (MDO) is a key technology in the further improvement of the energy efficiency of aero engines. Turbomachinery design is highly based on multidisciplinarity. Structural analysis, aerodynamics and heat transfer are some of the essential disciplines to be coupled in the design process in order to achieve a global, constrained design optimality. The flow equations are solved to find the structural load. This load acts on the blades and influences their shape, thus altering again the flow equations.

Most of internal flows in turbomachinery applications are related to viscosity and turbulence. To simulate such a flow solving the Euler equations is not accurate enough. Theses equations are indeed able to capture shocks but neglects viscous effects. In order to resolve boundary layer occurring on the hub, shroud and on blades the full Navier-Stokes equations have to be solved. Solving the full Navier-Stokes equation requires a very high computational power even for low Reynolds number. Therefore some methods have been developed that models the turbulence or only part of its scales. Reynolds-Averaged Navier-Stokes approaches models all scales of turbulence. Large Eddy Simulation solves the low frequency part and models the rest. Direct Numerical Simulation solves all turbulence but the computational power required is scaling as Re^3[1] which stay beyond the today computing power.

In order to increase the time performance even RANS equations are run on High Performance systems. The GPU constitutes one of the most powerful shared-memory systems with hundreds of cores and is therefore a new alternative in HPC.

Graphics Processing Units (GPU) have been used from the eighties to display data on screen. As their function was clear and predefined the users had no control on the procedure of data preparation and displaying. The concept was based on a closed *graphic pipeline*. A multitude of small computation units calculate simultaneously the right color of pixels using information send by the Central Processing Unit.

With the evolution of GPU techniques users had more and more possibilities to control the running functions on the pipeline. With the appearance of CUDA-enabled GPUs in 2007 [2] the users started to write their own function on GPUs with a C-similar programming language called CUDA C [9].
Scientists showed indeed interests on GPU even before a comfortable programming language like CUDA appeared. They translated their CFD problems into graphics functions to have the GPU solving them in parallel. This interest for GPU is motivated by the relatively high computational power of GPUs. The large number of computation units (CUDA cores) is able to compute CFD variables of a numerical cell inside a grid analogous to a pixel inside a picture. GPU designers are increasing the number of CUDA cores and the interest is much bigger since the programming burden is reduced.

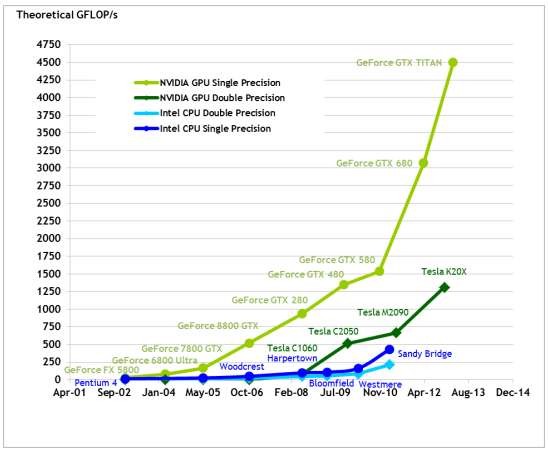

Figure 1: Floating-Point Operations per Second for the CPU and GPU

The computational power is measured on float operation per second (FLOPs). Figure 1 taken from the CUDA programming guide [9] shows the fast increasing gap on computational power between CPUs and GPUs over last decade on single and double precision.

GPU-cores are indeed very efficient for intensive arithmetical operation but present a weakness when it comes to data transfers. The GPU code has therefore to cope with the memory bandwidth as a bottle neck following a new programming approach. The challenge is then to write traditional sequential algorithms in a parallel way giving different cores the possibility to compute simultaneously.

This paper focus on the presentation of the MDO system CADO host of all optimization efforts related to this work and the CFD simulation of the CADO system to be optimized. Preliminary results are also presented.

2. Multidisciplininary Design Optimization

2.1. Optimization Algorithm

The multidisciplinary optimization tool CADO [5] developed at the Von Karman Institute for Fluid Dynamics (VKI) is conceived to improve the performance of turbomachinery components. The CADO algorithm presented in figure 2 involves a high fidelity performance evaluation based on computational fluid dynamics analysis (CFD) and Computational Solid Mechanics analysis (CSM). A metamodel assisted low-fidelity performance estimation reduces the computational costs and accelerates the design optimization process. The CFD analysis, performed in the actual version by commercial packages, takes much longer than the CSM analysis. Therefore the focus is on the optimization of the CFD simulation runtime. The saved run-time could decrease the computation costs or serves to perform more designs analysis and so increase the accuracy of the optimization algorithm.

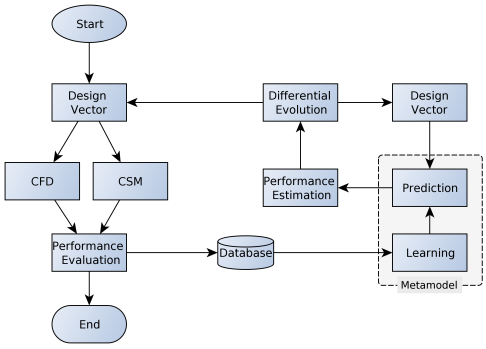

Figure 2: VKI optimization Algorithm (CADO)

GPUs have been already used to accelerate CFD simulation. Pullan et. al[3] used this hardware to accelerate a turbomachinery code having an over all speedup of one order of magnitude. Wang et Al. adapted his high-order accurate CFD method to GPU accomplishing a maximum speed up of 70x against single CPU core [4].

As a first step in the optimization of the CFD simulation an Euler-solver has been rewritten with CUDA C and so adapted for GPU. The CPU-based and the GPU-based Euler-solvers have been kept working side to side during the developing process to ensure the same output of both solvers given the same initialization data. The exact behavior of both solver has been verified automatically. The function per function porting of the CPU code relied on the *verification* of each ported piece of code against the CPU-based one.

A wider use of GPU on CADO is to be investigated. Running the metamodel on GPU with single precision could boost the process since GPU have an optimized hardware and function sets for single precision arithmetic[9]. New MDO algorithm are also to be investigated.

2.2. Flow Analysis

The flows considered in turbomachinery are viscous and turbulent.

2.2.1. Governing equations

The conservative integral form of the governing equations for internal flow in turbomachinery [1] are :

$$\frac{\partial}{\partial t}\int_\Omega \vec{W} d\Omega + \oint_{\partial\Omega} (\vec{F_c} - \vec{F_v}) dS = \int_\Omega \vec{Q} d\Omega \qquad (1)$$

where the conservative variable vector $\vec{W}$ consist of those components in three dimensions:

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \qquad (2)$$

and the convective flux vector is defined as follows:

$$\vec{F_c} = \begin{bmatrix} \rho V \\ \rho u V + n_x p \\ \rho v V + n_y p \\ \rho w V + n_z p \\ \rho H V \end{bmatrix} \qquad (3)$$

With following *contravariant velocity* definition:

$$V \equiv \vec{v} \cdot \vec{n} = n_x u + n_y v + n_z w \qquad (4)$$

and the viscous flux is

$$\vec{F_v} = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\Theta_x + n_y\Theta_y + n_z\Theta_z \end{bmatrix} \qquad (5)$$

where

$$\begin{aligned} \Theta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x} \\ \Theta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y} \\ \Theta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z} \end{aligned} \qquad (6)$$

and the viscous stress tensor τ after application of stokes hypothesis ($\lambda + \frac{2}{3}\mu = 0$) reads in tensor notation:

$$\begin{aligned} \tau_{ii} &= 2\mu\left(\frac{\partial v_i}{\partial x_i} - \frac{1}{3} div\vec{v}\right) \\ \tau_{ij} &= \tau_{ji} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \end{aligned} \qquad (7)$$

with following source term Q:

$$\vec{Q} = \begin{bmatrix} 0 \\ \rho f_{e,x} \\ \rho f_{e,y} \\ \rho f_{e,z} \\ \rho \vec{f_e} \cdot \vec{v} + \dot{q}_h \end{bmatrix} \qquad (8)$$

In absence of external force and in non-rotating system the source term is equal to zero. In a rotating frame of reference the Coriolis and centrifugal forces have effects on the source term. The appropriate formulation in this case is to be found in [1].

2.2.2. Turbulence Modeling

An MDO systems requires a large number of designs evaluation partly by means of flow simulations to achieve an improved design. The time and memory consuming DNS and LES methods are therefore out of reach of actual MDO systems. In the Reynolds-Averaged Navier-Stokes equation (RANS) the turbulence is however fully modeled. The time efficient RANS equations are consequently widely used on MDO systems. To account for turbulence the in-house Navier-Stokes equations solver uses a modified version [6] of SpalartAllmaras (SA) one-equation turbulence model [7; 8]. The SA Model is based on a linear eddy viscosity. It defines the turbulent viscosity (eddy viscosity) as:

$$\mu_T = \rho\tilde{\nu} f_{v1} \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad \chi = \frac{\tilde{\nu}}{\nu} \tag{9}$$

the fully turbulent version of the transport equation for the variable $\tilde{\nu}$ is

$$\begin{aligned} \frac{D\tilde{\nu}}{Dt} &= \underbrace{c_{b1}\tilde{S}\tilde{\nu}}_{\text{production}} \\ &+ \underbrace{\frac{1}{\sigma}\{\nabla\cdot[(\tilde{\nu}+\nu)\nabla\tilde{\nu}] + c_{b2}(\nabla\tilde{\nu})^2\}}_{\text{diffusion}} \\ &- \underbrace{c_{w1} f_w(r)\left(\frac{\tilde{\nu}}{d}\right)^2}_{\text{destruction}} \end{aligned} \tag{10}$$

Full details about the used modified Spalart-Allmaras model can be found in [6].

The SA model is linked to the Navier-Stokes equation by the extension of the dynamic viscosity μ on equation 7 [1]:

$$\mu = \underbrace{\mu_L}_{\text{laminar viscosity}} + \underbrace{\mu_T}_{\text{turbulent viscosity}} \tag{11}$$

3. Results

The first result concerns the verification and porting for GPU of the in-house CFD solver for inviscid Euler-governed flows using the GAMM circular bump testcase [10]. The second result is the validation of in-house CPU-based solver for turbulent viscous flows against the VKI turbine inlet guide vane testcase [11].

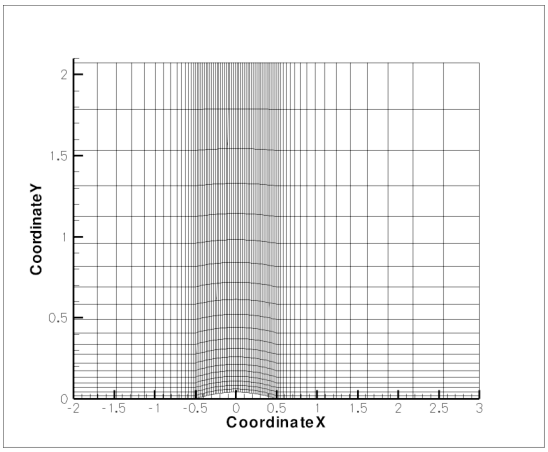

Figure 3: 72x21 Mesh of the GAMM [10] test case

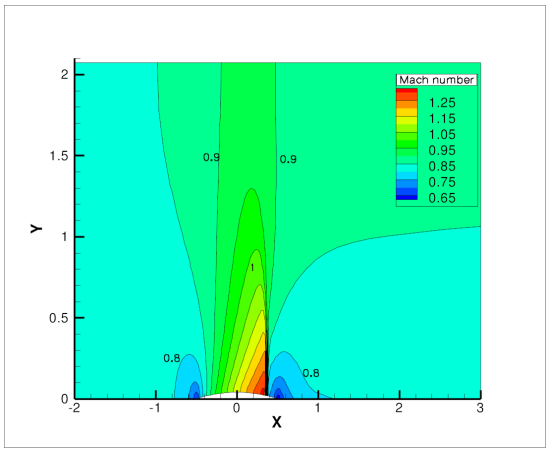

Figure 4: Mach contours for output isentropic Mach number of 0.85

3.1. GAMM circular arc "bump"

The test case is an internal transonic flow through a parallel channel having a 4.2 % thick circular bump on the lower wall as shown in Figure 3. The boundary condition concerns the exit pressure giving an isentropic Mach number of 0.85. The Mach number contours presented in figure 4 shows a supersonic pocket on trailing edge of the bump ended by a normal shock. The solver uses a central scheme for the space discretization. For the sake of verification the same test case with same setting is run by Numeca FineTurbo CFD solver with Spalart Allmaras turbulence model. Comparison of the pressure coefficient c_p of both simulations is shown in figure 5. The two distributions are very close and predict the shock at the same position on the bump. The Numeca solver predicts however a higher pre-shock Mach number.

Starting from a low level of complexity the performance of the ported Euler equations solver has been measured on this test case. A relative small speedup of 1.15 has been achieved, which is a reasonable start for a GPU program not making use of most of the GPU optimization techniques.

Table 1: Performance comparison on GAMM inviscid testcase

Hardware	Number of cores	Memory Bandwidth [GB/s]	Speedup
Xeon(R) E3	1	25.6	1
Tesla C2070	448	144	0.6
Geforce 780	2304	288.4	1.15

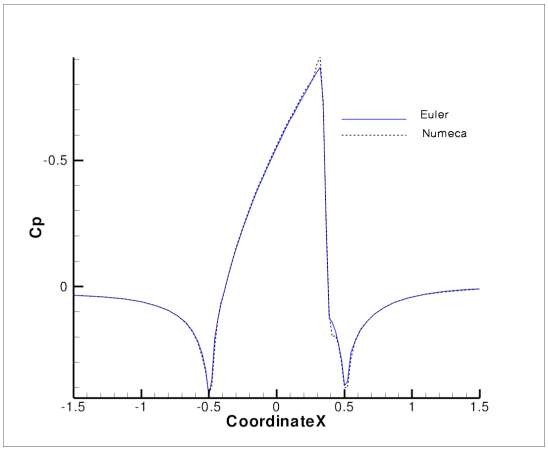

Figure 5: Cp distribution along the channel lower wall for different solvers (M2i=0.85)

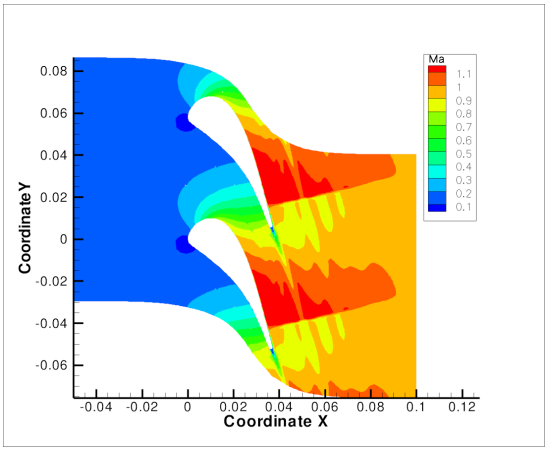

Figure 7: Mach contours of the transonic LS89 [11] turbine guide vane test case

3.2. *LS89 Transonic Turbine Guide Vane*

Figure 6 shows the multi-block mesh used on the testcase. An O-mesh block with small near wall cell width surrounds the turbine guide vane profile for a better resolution of the boundary layer.

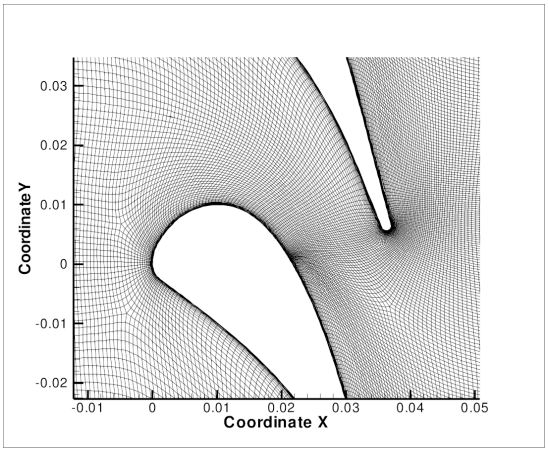

Figure 6: Multiblock Mesh of the LS89 VKI turbine guide Vane cascade [11]

The boundary condition concerns the exit isentropic Mach number, which leads to a subsonic or a transonic testcase. Figure 7 shows the Mach number contours of the transonic testcase revealing a normal shock occurring on the suction side of the profile.

The isentropic Mach number is computed from local static pressure using a constant total pressure equal to the inlet total pressure. The isentropic Mach number is obtained by following equation:

$$M_{is} = \sqrt{\frac{2}{\gamma - 1}((\frac{p_0}{p})^{\frac{\gamma-1}{\gamma}} - 1)} \qquad (12)$$

For the validation, numerical isentropic Mach number has been compared to the experimental one. Figure 8 and 9 shows the turbine blade surface isentropic Mach number distributions of the experiment and the numerical calculation with a subsonic and a transonic exit isentropic Mach number. The calculation results are in a fair agreement with the experimental data although the transonic case is better resolved

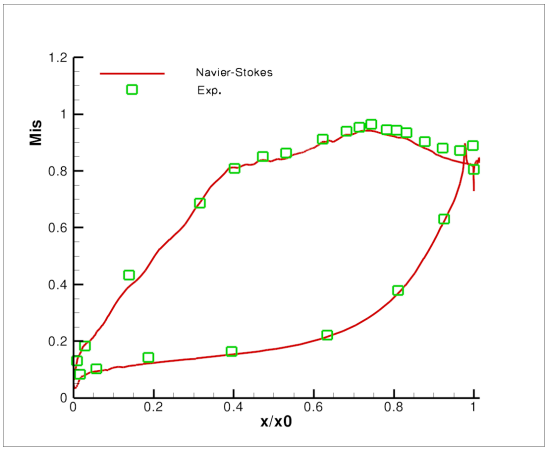

Figure 8: Mach-number distribution on LS89 turbine subsonic case : M_{is}=0.84 and P_{01}= 1.435 bar

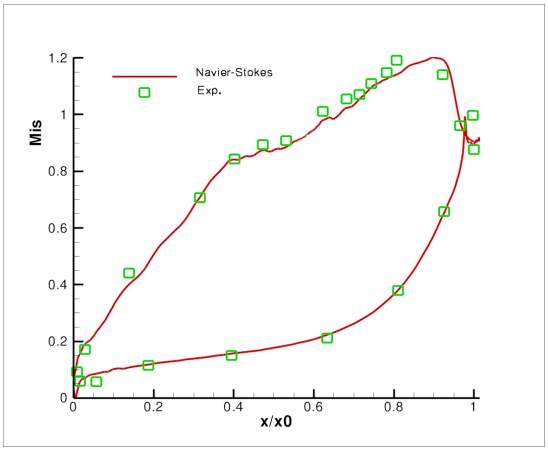

Figure 9: Mach-number distribution on LS89 turbine subsonic case : M_{is}=1.02 P_{01}= 1.605 bar

4. Conclusion

This work showed the progress of the development toward a GPU-based multidisciplinary design optimization system. The focus is on the CFD solver of the MDO system. First performance optimization of 15% are performed on Euler-governed flows. The validated CPU version of the Navier-stokes solver will be ported and a wide variety of GPU optimization techniques will be used to achieve an interesting speedup able to compete with the implicit version of the same in-house flow solver. The acceleration of the CPU solver using multigrid techniques will be performed for a mature performance comparison with future GPU-version. The multigrid acceleration technic will be investigated on GPU also as it showed benefits on GPU clusters [12; 13].

Acknowledgments

This research activity is funded by a Marie Curie Action as part of the European union's Framework 7 research program (AMEDEO :Project No. 316394). This work is using the CPU-based Solver of Dipl.-Ing. Lasse Müller

References

[1] J. Blazek, Computational Fluid Dynamics:Principles and Applications, Elsevier, 2001.

[2] E. Lindholm , J. Nickolls, S. Oberman and J. Montrym , Nvidia Tesla: a Unified Graphics and Computing Architecture, Micro, IEEE 28 (2) (2008) 39–55.

[3] Brandvik, Tobias and Pullan, Graham , An Accelerated 3D NavierStokes Solver for Flows in Turbomachines, Journal of Turbomachinery 133 (2).

[4] B. Zimmerman and Z.J. Wang , The Efficient Implementation of Correction Procedure Via Reconstruction with GPU Computing, AIAA 2013-2692.

[5] Verstraete T., CADO: a computer aided design and optimization tool for turbomachinery applications, 2nd International Conference on Engineering Optimization.

[6] J. R. Edwards and S. Chandra, Comparison of Eddy Viscosity-Transport Turbulence Models for Three-Dimensional, Shock-Separated Flowfields, AIAA Journal.

[7] Spalart P. R. and Allmaras S. R., A one-equation turbulence model for aerodynamic flows, AIAA-Paper 92-0439.

[8] Allmaras S. R., Johnson F. T. and Spalart P. R., Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model, ICCFD7-1902.

[9] Nvidia, CUDA C Programming Guide, PG-02829-n 001 v5.5, 2013.

[10] A. Rizzi and H. Viviand, Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves, Notes on Numerical Fluid Mechanics (3).

[11] T. Arts, M. Lambert de Rouvroit and A.W. Rutherford, Aerothermal Investigation of a highly loaded transonic linear Turbine Guide Vane Cascade, TN 174, von Karman Institute for Fluid Dynamics (1990).

[12] D. Göddeke, Fast and accurate finite-element multigrid solvers for PDE simulations on GPU clusters, Ph.D. thesis, Technische Universität Dortmund, Fakultät für Mathematik (May 2010).

[13] D. Goddeke, R. Strzodka, J. Mohd-Yusof, P. McCormick, H. Wobker, C. Becker, S. Turek, Using gpus to improve multigrid solver performance on a cluster, Int. J. Comput. Sci. Eng. 4 (1) (2008) 36–55.