A Grid Enrichment and Movement Strategy for A Posteriori Error Analysis in Viscous Flows

T. Zervogiannis, P.I.K.D. Liakopoulos, D.I. Papadimitriou and K. C. Giannakoglou

Laboratory of Thermal Turbomachines, National Technical University of Athens,
P.O. Box: 64069, Athens 157 10, Greece,   e-mail: kgianna@central.ntua.gr

ABSTRACT

This paper presents an adjoint-based grid adaptation algorithm which, in turbomachinery *CFD*, is capable of computing integral functionals with desired accuracy. Unstructured grid enrichment, moving and smoothing strategies are employed, driven by a sensor that is based on the functional error estimates. Their combined use allows the computation of integral quantities where the error is less than a user defined threshold, on grids with a "reasonable" number of nodes. For the demonstration of the proposed method two integral functionals are used: the first is the entropy generation field integral accounting for profile losses in a cascade and the second stands for the total pressure losses between the cascade inlet and outlet. The studies presented in the results section are concerned with losses associated with the laminar flow through a compressor cascade. Turbulent flow cases are expected to behave similarly as long as the exact discrete adjoint equations are used but this is beyond the scope of this paper.

INTRODUCTION

The role of the adjoint method in the optimization of aerodynamic shapes is widely known. The formulation and numerical solution of adjoint equations provides the objective function's gradient with respect to the design variables, which drives descent algorithms to the optimal solution. A different application of the adjoint method is related to error estimation in computing functional integrals by post processing the outcome of the numerical solution of partial differential equations on a given grid. This error can be used as a sensor for optimal grid refinement methods leading to adapted grids on which the functional output has the desirable accuracy.

The method which is known as posteriori error estimation for functional correction and grid adaptation was proposed by Giles (1998); see also Giles and Pierce (1999, 2002), Giles and Suli (2002), Pierce and Giles (2004) in external aerodynamics. Another variant of the method has been introduced by Venditti and Darmofal (1999). They proposed the combined use of two grids, namely a coarse grid on which the flow and adjoint equations are solved and a fine one on which the functional is estimated. The method was first applied to quasi-one dimensional flows, Venditti and Darmofal (2000), to compute the integral of the static pressure. Two dimensional applications can be found in Venditti and Darmofal (2001, 2002), where the functional of interest is the airfoil lift or drag. Viscous effects have been taken into account, Giles et al. (1997), Venditti and Darmofal (2003), and the method was extended to 3D problems by Park (2002).

In the present paper, an extension of the method to turbomachinery applications is proposed. Two functionals which quantify losses in turbomachinery blades are used. They correspond to the entropy generation due to profile losses and the averaged total pressure losses between the inlet to and the outlet from the cascade. Though the physical significance of both functionals is the same, they lead to different adjoint problems; entropy losses are expressed as a field integral, in contrast to the total pressure losses which are given by a boundary integral along the inlet and outlet cascade boundaries. So, the difference between the functionals lies on the source terms appearing in the adjoint equations.

The programmed software was applied to the error estimation of the two functionals and the subsequent grid adaptation over a 2D compressor cascade, in laminar flow. The extension to turbulent flows is straightforward as long as an additional adjoint equation is formulated and solved, taking into account the variation in turbulent viscosity (under progress). To the authors' opinion, however, testing the proposed method on laminar flows is adequately illustrative.

The flow solver used for the present work is a time–marching Euler/Navier–Stokes equations solver for unstructured grids, based on the finite–volume technique and an upwind formulation, employing the Roe' s approximate Riemann solver (Roe (1981)) with variables extrapolation to account for second-order accuracy and the use of the van Leer–van Albada limiter (van Albada et al. (1982)). The point–implicit Jacobi scheme is used for both the flow and adjoint equations.

A POSTERIORI ERROR ANALYSIS

Our objective is to compute an integral quantity $f(U)$, where U is the vector of the flow variables, with the desired accuracy and the lowest possible CPU cost, i.e. using a grid of minimum size. As proposed by Venditti and Darmofal (1999), an iterative scheme utilizing two computational grids per cycle can be used: a *coarse* one (index H; U_H for instance, denotes the flow variables computed on the coarse grid nodes), on which flow equations $R_H(U_H) = 0$ and functional $f_H(U_H)$ are computed with low accuracy and at low cost, and a *fine* one (index h) on which we refrain from performing computations. Instead, whatever needed on the fine grid is interpolated from the coarse one using the prolongation operator I; either linear or quadratic interpolation can be used based on nodal information from the segment vertices to interpolate data on the new segment midnode. The quadratic operator interpolates data on the midnode using nodal values and gradients. The operator which was chosen for the present work was quadratic due to higher accuracy. Thus $f_h(U_I) = f_h(IU_H)$. From

$$f_h(U_h) \approx f_h(U_I) + \frac{\partial f_h}{\partial U_h}|_{U_I}(U_h - U_I) \tag{1}$$

$$R_h(U_h) \approx R_h(U_I) + \frac{\partial R_h}{\partial U_h}|_{U_I}(U_h - U_I) = 0 \tag{2}$$

where $\frac{\partial f_h}{\partial U_h}|_{U_I}$ and $\frac{\partial R_h}{\partial U_h}|_{U_I}$ are computed using U_I, which leads to the functional estimate

$$f_h(U_h) \approx f_h(U_I) - \frac{\partial f_h}{\partial U_h}|_{U_I} \left[\frac{\partial R_h}{\partial U_h}|_{U_I}\right]^{-1} R_h(U_I) = f_h(U_I) - \Psi_h^T R_h(U_I) \tag{3}$$

where

$$\left[\frac{\partial R_h}{\partial U_h}|_{U_I}\right]^T \Psi_h = \left(\frac{\partial f_h}{\partial U_h}|_{U_I}\right)^T \tag{4}$$

To circumvent the numerical solution of eq. 4 on the fine grid, we solve

$$\left[\frac{\partial R_H}{\partial U_H}\right]^T \Psi_H = \left(\frac{\partial f_H}{\partial U_H}\right)^T \tag{5}$$

on the coarse grid and, then, prolongate Ψ_H on the fine grid, by means of a prolongation operator J, to get $\Psi_J = J\Psi_H$. By doing so, the functional is approximated by

$$\tilde{f}_h(U_h) \approx f_h(U_I) - \Psi_J^T R_h(U_I) \tag{6}$$

FUNCTIONALS FOR TURBOMACHINERY APPLICATIONS

(a) The first functional expresses the entropy generation over the flow domain Ω

$$f(U) = \int_{\Omega} \rho u_i \frac{\partial s}{\partial x_i}\, d\Omega = \int_{\Omega} \frac{1}{T} \tau_{ij} \frac{\partial u_i}{\partial x_j}\, d\Omega \tag{7}$$

where τ_{ij} is the stress tensor. In contrast to functionals used in previous similar works (such as the lift or drag), this functional is a field integral over the whole flow domain.

(b) The second functional accounts for total pressure ($_t$) losses and is expressed as the difference in area averaged p_t between the cascade inlet and outlet. We define

$$f(U) = \int_{S_{in}} p_t dS - \int_{S_{out}} p_t dS \tag{8}$$

but since the inlet p_t is fixed and known, the first integral is eliminated.

GRID ADAPTATION

This section presents two adaptation strategies employed during the numerical computations. The first one is grid embedding adaptation (h-refinement) according to the sensor described in the subsequent subsection; it is the same sensor that drives the grid moving adaptation strategy, used as either alternative or complementary adaptation technique.

Grid Embedding Adaptation

Starting from eq. 3, the estimated error in the functional $f(U)$ can be related to the local residual of the primal ($R(U)$) or the adjoint ($R^{\Psi}(\Psi)$) solutions, as follows

$$f_h(U_h) - f_h(U_I) = \Psi_J^T R_h(U_I) + (\Psi_h - \Psi_J)^T R_h(U_I) \tag{9}$$

or

$$f_h(U_h) - f_h(U_I) = \Psi_J^T R_h(U_I) + \left(R_h^{\Psi}(\Psi_J)\right)^T (U_h - U_I) \tag{10}$$

where the residual of the adjoint equations on the fine grid is

$$R_h^{\Psi}(\Psi) \equiv \left[\frac{\partial R_h}{\partial U_h}|_{U_I}\right]^T \Psi - \left(\frac{\partial f_h}{\partial U_h}|_{U_I}\right)^T \tag{11}$$

The last terms in eqs. 9 and 10 stand for the error in the computed correction of the functional. Venditti and Darmofal (2002, 2003) recommend the following adaptation parameter $\eta_k = \frac{\varepsilon_k}{\bar{e}_o}$ where $\bar{e}_o = \frac{e_0}{N_h}$, e_0 is a user-defined threshold for the integral output error, N_h is the number of fine grid nodes and ε_k is the adjoint based estimate of the error, computed over the fine grid nodes as follows

$$\varepsilon_k = \frac{1}{2}\left\{ |\, [Q_h\Psi_H - L_h\Psi_H]_k^T \, [R_h(L_hU_H)]_k \,| + |\, [Q_hU_H - L_hU_H]_k^T \left[R_h^{\Psi}(L_h\Psi_H)\right]_k |\right\} \tag{12}$$

with L_h and Q_h being linear and quadratic extrapolation operators, accordingly. In a grid enrichment method, the nodal η_k is scatter added to the grid edges and those marked with $\eta_k > 1$ should be refined. The h-refinement used is a triangle embedding technique presented by Giotis et al. (2001) and terminates when the global error in the computed correction drops below the predefined threshold.

Grid Moving Adaptation

Apart from grid embedding, the computational accuracy of $f(U)$ can be improved using adaptation based on a fixed number of grid points and fixed connectivity among them, i.e. the so-called grid moving adaptation. With the available η_k values, grid nodes are moved using a weighted modified Laplacian filter (Liakopoulos and Giannakoglou (2006)). For each internal node, a surrounding polygon is defined according to the shortest edge emanating from this node, fig. 1. The position vectors of the polygon vertices are given by

$$\overrightarrow{r}_{j'} = \overrightarrow{r}_i + \rho_{i,min} \frac{\overrightarrow{r}_{ij}}{|\overrightarrow{r}_{ij}|}, \quad \rho_{i,min} = min(|\overrightarrow{r}_{ij}|) \tag{13}$$

Each grid node moves according to the following expression (ω=relaxation factor)

$$\overrightarrow{r^*}_i = (1-\omega)\overrightarrow{r}_i + \omega \frac{\sum_{j=1}^{n_i} \eta_k \overrightarrow{r}_{j'}}{\sum_{j=1}^{n_i} \eta_k} \tag{14}$$

towards the centroid of its surrounding polygon; its vertices act as point sources with intensity proportional to η_k so as to reduce the distance between nodes in areas where the error is high. Due to eq. 14, grid smoothing is implicit to the previously described grid moving adaptation.

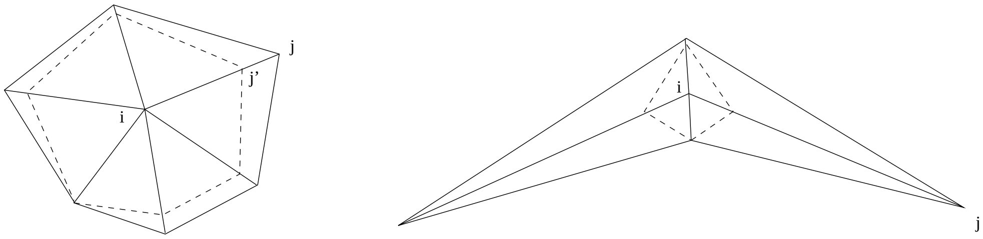

Figure 1: Grid moving adaptation. The new constrained surrounding polygon (dashed line). The feasible area lies within the surrounding polygon even in extreme cases.

METHOD APPLICATION

The use of the proposed algorithm is demonstrated in the numerical prediction of the subsonic flow in a compressor cascade ($\alpha_{in} = 47^\circ$, $M_{out,is} = 0.37$). In our attempt to get results which are not affected by errors due to the omitted adjoints to the turbulent model equations (this omission is actually, the standard practice), the flow is considered to be laminar ($Re = 1000$). The algorithm's behavior is examined using different values for the error threshold and four different scenarios for the adaptation. In conformity with figure captions, these are:

- **Scenario A**: grid adaptation is restricted to grid embedding, i.e. the η_k based *h*–refinement.
- **Scenario B**: grid adaptation includes alternating use of grid moving and *h*–refinement adaptation between cycles.
- **Scenario C**: stands for the most "aggressive" approach, where both grid moving and *h*–refinement are applied at each cycle.
- **Scenario D**: similar to Scenario A, after employing the smoothing technique proposed by Liakopoulos and Giannakoglou (2006) to the initial grid.

- **Uniformly Refined**: the flow equations are solved directly on a sequence of uniformly refined grids, each generated from the previous after splitting each parent triangle into four triangles.

In order to assess the gain in CPU cost, figs. 2 and 3 compare results obtained using scenarios A,B,C and the uniformly refined grids with the entropy generation acting as the integral functional. Convergence plots and results with different preset error thresholds are shown. Since, apart from CPU cost, memory requirements are also of concern, let us state that the adjoint adaptation procedure has higher memory requirements than the numerical solution of the flow equations, increased by a factor of approximately 1.9.

In addition, Richardson extrapolation was applied to the two values obtained by the last two uniformly refined grids so as to get a reference value to compare with. The same procedure was also followed for the calculation of total pressure losses and the associated errors. The functional values plotted in figs. 2 and 3 include the correction imposed by eq. 6, the effect of which can be clearly seen in fig. 4, where the corrected values come closer to the asymptotic limit. Applying h-refinement and grid moving adaptation alternatively seems to stagnate the convergence curve since, at the grid moving cycle, the solution of the adjoint equations imposes additional CPU cost without significantly improving the accuracy. However, applying both h-refinement and grid moving adaptation at all cycles (Scenario C) yields faster convergence than scenarios A or B.

As expected, the use of low error threshold results in a more accurate estimate of the functional, though CPU and memory requirements increase accordingly. Even for higher accuracy ($e_0 = 5 \times 10^{-5}$) the method proves to be better than solving on a very fine grid, both in terms of memory and CPU cost. Indicatively, computing the functional with an error tolerance of 10^{-4} costs about 2300 CPU seconds and yields a finally adapted grid of 22968 nodes. Also, the cost for meeting an error tolerance of 5×10^{-5} is 2800 CPU seconds and the final grid possesses 32075 nodes. Note that, in all figures, the horizontal axis is in logarithmic scale. Comparing the three adaptation approaches (Scenarios A,B and C) in terms of final grid size does not lead to definite conclusions about the superiority for any of them.

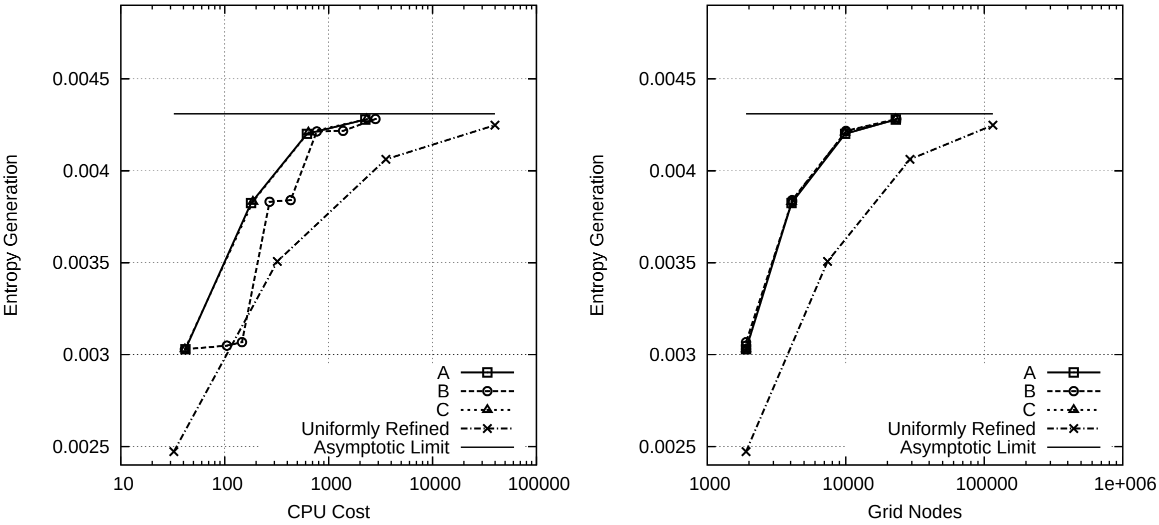

Figure 2: Integral functional : corrected field integral of entropy generation – error threshold $e_0 = 10^{-4}$. Error convergence plots for scenarios A,B and C versus CPU cost and grid size. Here and in all subsequent figures as well as the ensuing discussion, A,B and C correspond to the corrected functional values according to eq. 6, if not stated differently.

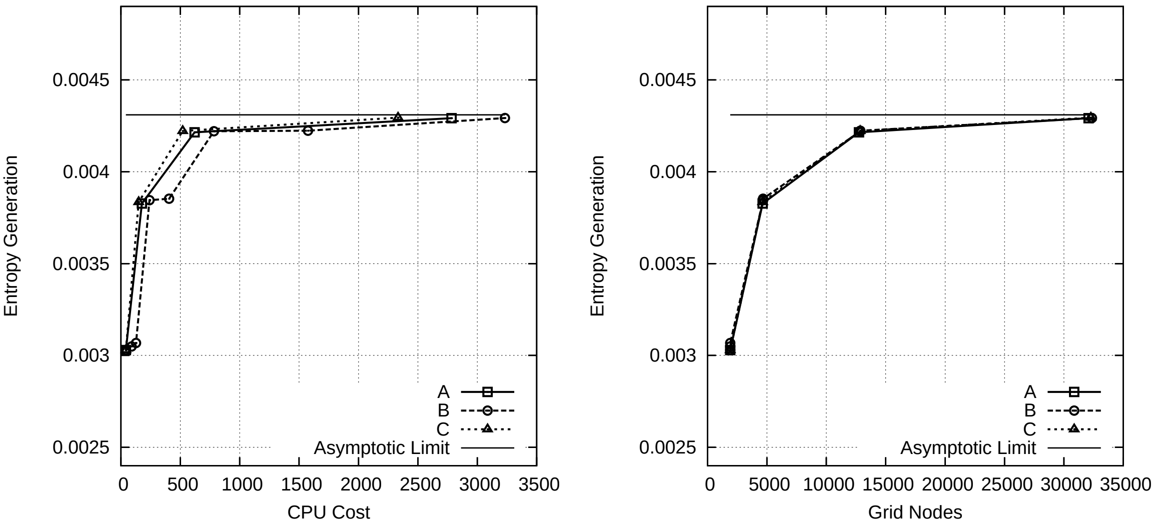

Figure 3: Integral functional : corrected field integral of entropy generation – error threshold $e_0 = 5 \times 10^{-5}$. Error convergence plots for scenarios A,B and C versus CPU cost and grid size.

A second series of tests are concerned with the method application to the prediction of total pressure losses in the same cascade, figs. 5 and 6. In all cases, the final grid size obtained by the adaptation method is about four times less than that of the uniformly refined grids which provides the reference - asymptotic solution. The CPU cost follows the same trend, falling four to six times below that required by the uniformly refined grids. In this case, Scenario B behaves poorly and converges to a value which is notably different than the asymptotic limit.

Finally, when the initial grid was smoothed prior to its adaptation, the method is faster and the computed value is closer to the asymptotic limit, regardless of the threshold used ($e_0 = 10^{-4}$ or $e_0 = 5 \times 10^{-5}$, fig. 7)

In figure 8 the starting grid and two of the final grids, properly adapted to reduce the error in entropy generation and total pressure losses prediction, are presented. In both cases, the proposed method causes intense adaptation of the grid inside the boundary layer and wake.

In all cases the memory requirements (taking into account the 1.9 factor) of the a posteriori adaptation starting from a coarse grid is less than the memory required to solve the flow equations on the corresponding fine grid, which has four times as many elements.

A comment on how the so-called asymptotic limit values are approached, in all cases examined, is due. Note that the starting grid is not adequate for laminar flow computations; this grid is unacceptably coarse close to the airfoil contour and in the wake. So, for the initial grid as well as those resulted from the very first adaptation cycles, although entropy generation constantly increases (figs. 2, 3, 4), total pressure losses decrease (figs. 5, 6). This is attributed to (a) numerical solution errors due to the locally coarse grid (b) integration errors for the same reason. Once the adapted grid becomes “acceptable” for laminar flow computations, the convergence curves for the two functionals behave similarly (both entropy and total pressure losses increase; see for instance figs. 5 and 6, after the second adaptation cycle).

CONCLUSIONS

A posteriori error analysis (which, thus far, has almost exclusively been used in external aerodynamics) together with h-refinement and grid movement was successfully applied to the accurate prediction of two integral functionals, related to the flow developed within a 2D compressor cascade. These functionals are the entropy generation due to profile losses and the total pressure losses. The

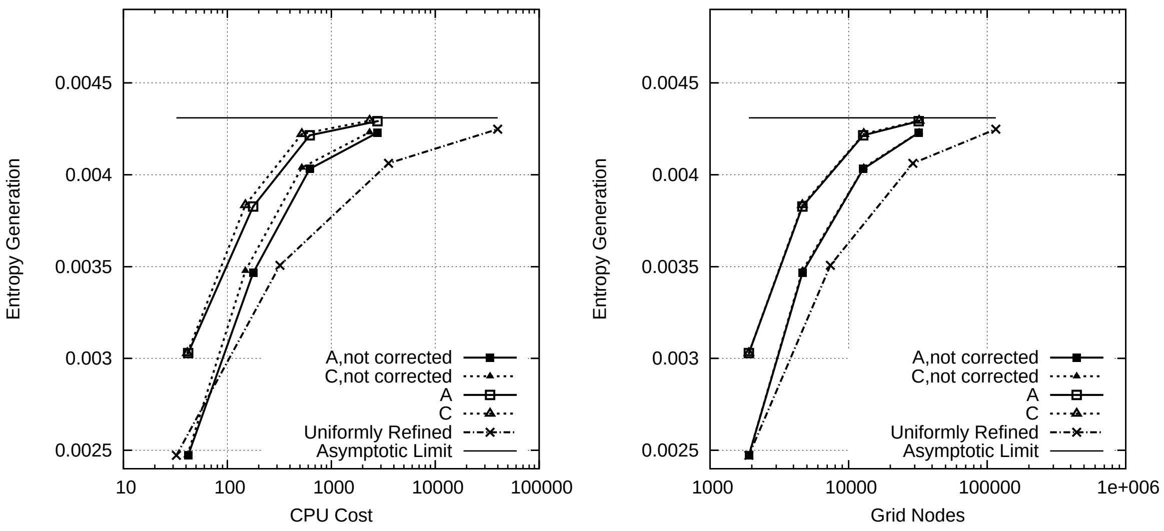

Figure 4: Integral functional : corrected field integral of entropy generation – error threshold $e_0 = 5 \times 10^{-5}$. Effect of correction on the estimated functional error.

solution of the adjoint equations provided corrections to the estimated functionals, while further improvements in accuracy were achieved through the combined use of grid embedding and moving. Similar beneficial effects were obtained by smoothing the initial grid prior to the adaptation procedure. The method's extension to turbulent and three–dimensional flows is under progress.

ACKNOWLEDGEMENTS

Parts of the current research were supported by grants from the Beneficial Foundation Alexandros S. Onasis and the National Scholarship Foundation of Greece. This work was also partially funded by the PENED01 program (Measure 8.3 of the Operational Program Competitiveness, of which 75% is European Commission and 25% national funding) under project number 01ED131.

REFERENCES

M.B. Giles. On adjoint equations for error analysis and optimal grid adaptation. *Frontiers of Computational Fluid Dynamics*, pages 155–170, 1998.

M.B. Giles and N.A. Pierce. Adjoint error correction for integral outputs. *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, pages 47–96, 2002.

M.B. Giles and N.A. Pierce. Improved lift and drag estimates using adjoint euler equations. AIAA Paper 99-3293, 1999.

M.B. Giles and E. Suli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, pages 145–236, 2002.

M.B. Giles, J.M. Larson, M.G. Levenstam, and E. Suli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. Report NA-97/06, Oxford University Computing Laboratory, 1997.

A. P. Giotis, P. I. K. Liakopoulos, D. G. Koubogiannis, and K. C. Giannakoglou. A fast ga-based data partitioner for use in parallel cfd. EUROGEN 2001 Conference, Athens, Sept. 19-21, 2001.

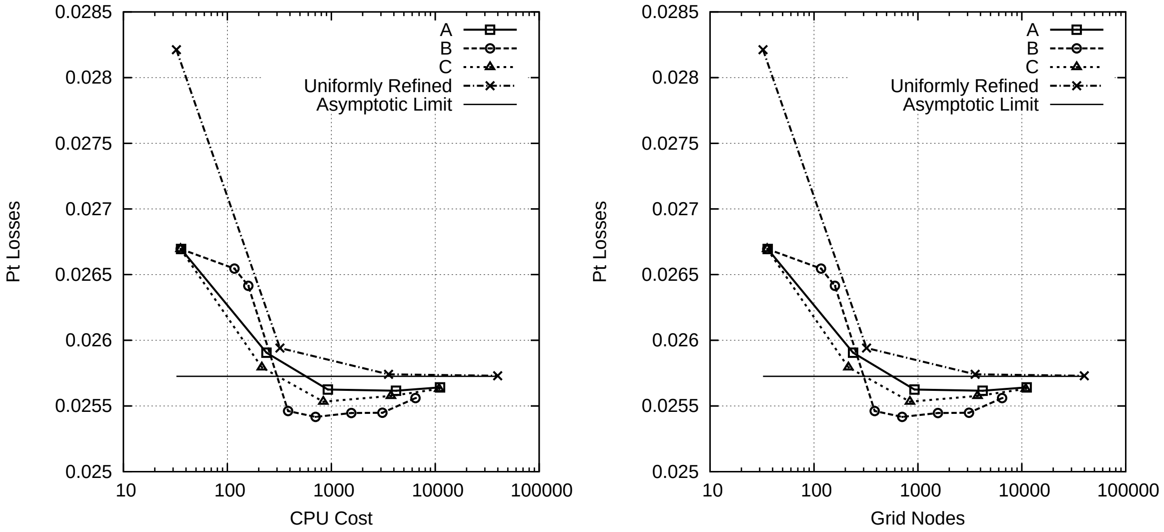

Figure 5: Integral functional : corrected total pressure losses – error threshold $e_0 = 10^{-4}$. Error convergence plots for scenarios A and C versus CPU cost and grid size.

P. I. K. Liakopoulos and K. C. Giannakoglou. Unstructured remeshing using an efficient smoothing scheme approach. ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, 2006.

M.A. Park. Adjoint-based, three dimensional error prediction and grid adaptation. AIAA Paper 2002-3286, 2002.

N.A. Pierce and M.B. Giles. Adjoint and defect error bounding and correction for functional estimates. *Journal of Computational Physics*, 200:769–794, 2004.

P. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–371, 1981.

G.D. van Albada, B. van Leer, and W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astron. Astrophys.*, 108:76–84, 1982.

D.A. Venditti and D.L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal of Computational Physics*, 164:204–227, 2000.

D.A. Venditti and D.L. Darmofal. A grid adaptive methodology for functional outputs of compressible flow simulations. AIAA Paper No. 01-2659, 2001.

D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: Application to 2-d inviscid, compressible flow. *Journal of Computational Physics*, 176:40–69, 2002.

D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22–46, 2003.

D.A. Venditti and D.L. Darmofal. A multilevel error estimation and grid adaptive strategy for improving the accuracy of integral outputs. AIAA Paper No. 99-3292, 1999.

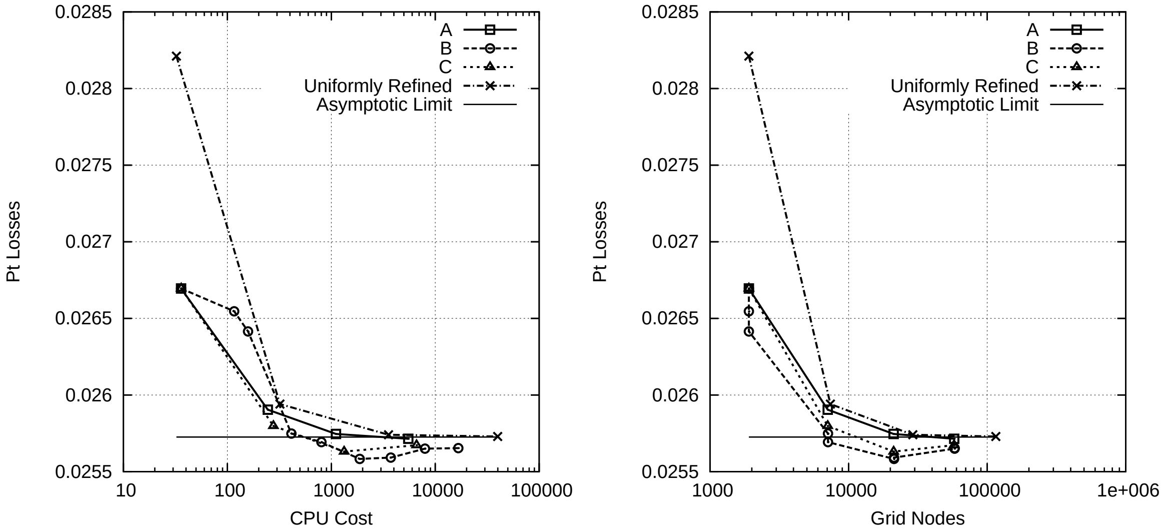

Figure 6: Integral functional : corrected total pressure losses – error threshold $e_0 = 5 \times 10^{-5}$. Error convergence plots for scenarios A and C versus CPU cost and grid size.

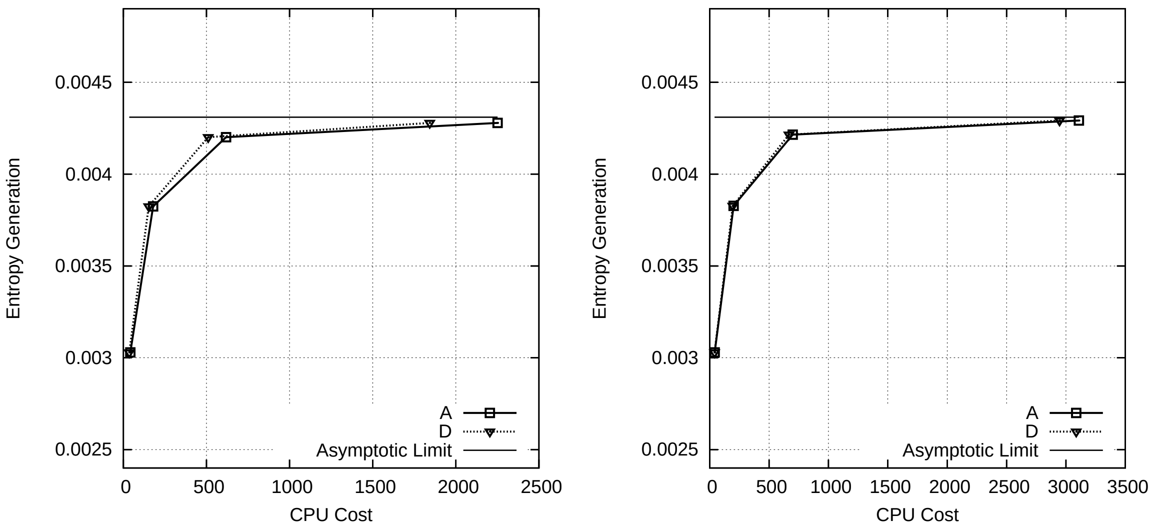

Figure 7: The effect of smoothing the starting grid prior to the adaptation procedure. The threshold used is $e_o = 10^{-4}$ for the estimation of the entropy generation and $e_o = 10^{-5}$ for the total pressure losses.

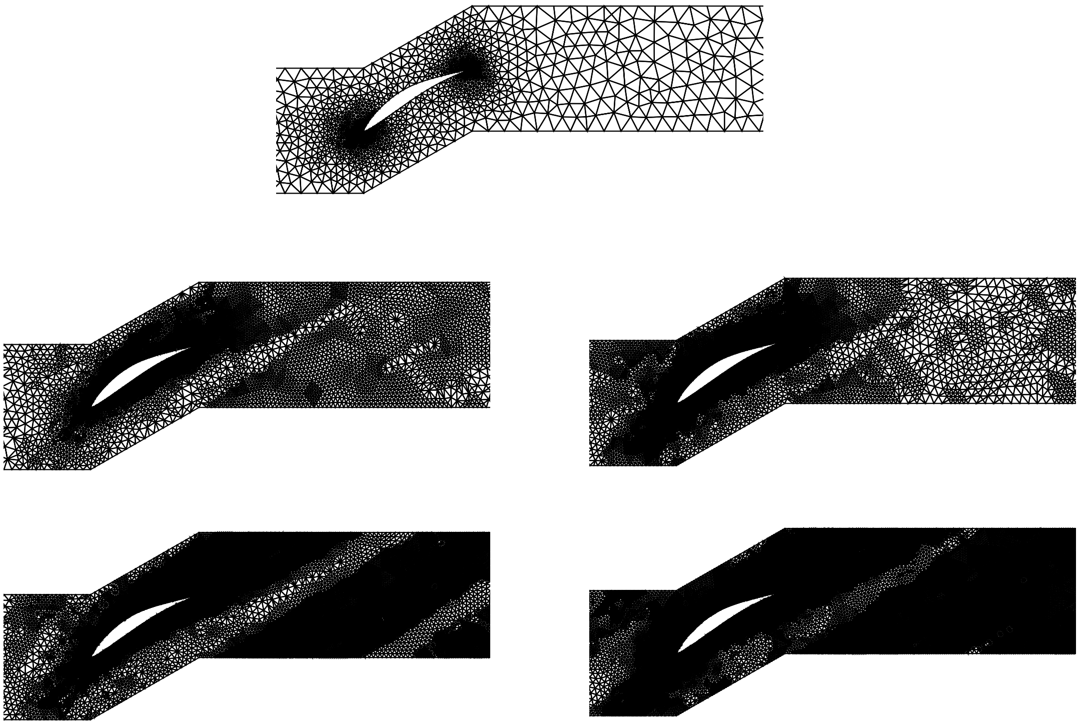

Figure 8: Starting grid (top), intermediate and finally adapted grids for accurately predicting entropy generation (left), p_t losses (right). Note that the starting grid is far from being adequate for the numerical solution of the laminar cascade flow.

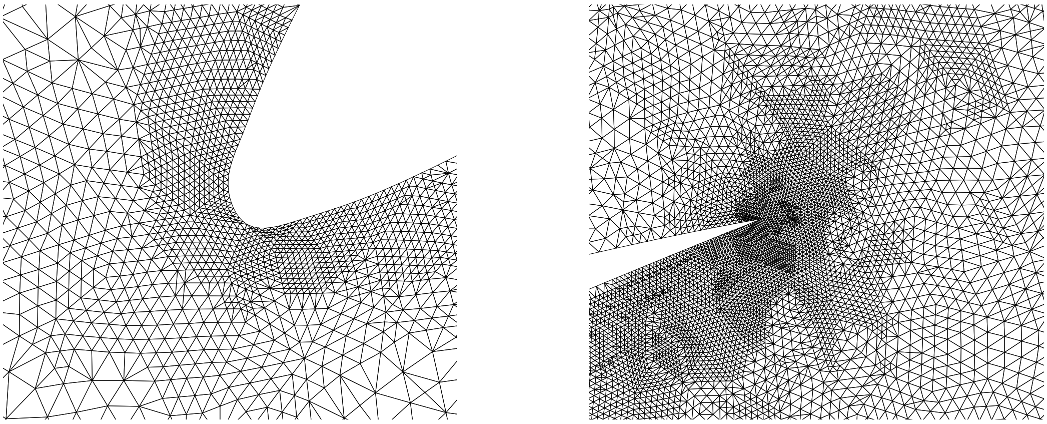

Figure 9: Close-up view of the final grid, close to the leading and trailing edge of the airfoil; this is the outcome of the total pressure losses based adaptive algorithm.