Numerical Investigation of Flow Control by a Virtual Flap on Unstructured Meshes

Sungyoon Choi* and Oh Joon Kwon

Division of Aerospace Engineering
Korea Advanced Institute of Science and Technology(KAIST)
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701 Republic of Korea

Key Words: Flow Control, Unstructured Meshes, Virtual Flap

Recently, numerous researches have been conducted for the improvement of the aerodynamic characteristics of existing airfoils using flow control. Flow control method is divided into two large catagories depending on whether it requires to supply energy for control or not. Passive control uses devices such as flap and slat, while active control utilizes blowing, suction, or synthetic jet actuator. Gurney flap is one of the passive flow control device which is a simple flat plate attached perpendicular to airfoil surface at the bottom of the trailing edge. The effect of Gurney flap on flow control have been previously investigated by several researchers[1, 2] and it was shown that the extra downward turning flow is generated behind the airfoil and this downward turning flow induces higher circulation which results in a pressure decrease on the suction side and a pressure increase on the pressure side as compared to a clean airfoil. This pressure difference between upper and lower surfaces results in the increment of the maximum lift, decrement of the zero-lift angle of attack, and increment the the nose-down pitching moment, showing that Gurney flap increases the effective camber of the airfoil[2]. As the size of Gurney flap increases, the amount of lift increment becomes higher, but when the flap hight exceeds the boundary layer thickness, the drag also increases substantially. Therefore the flap hight should be determined such that the maximum lift-to-drag ratio, which is one of the criteria of the airfoil aerodynamic efficiency, can be acquired.

In the present study, an active flow control method was used to numerically simulate the Gurnay flap by imposing a continuous jet in the vertical direction from the airfoil lower surface at the trailing edge to enhance the high-lift performance of the airfoil and to relieve the defects of using Gurney flap at the same time. The Governing equations are the two-dimensional, compressible Reynolds-averaged Navier-Stokes equations. The Spalart and Allmaras one-equation turbulence model was employed to evaluate the turbulent viscosity. A vertex-centered finite-volume method on unstructured meshes was used for the spacial discretization of the convective fluxes. A second-order accurate upwind-biased formula was adopted to evaluate the convective flux derivatives. The viscous terms were computed using second-order accurate central differences. The time advancement was achieved using a second-order accurate implicit time integration based on a point Gauss-Seidel relaxation scheme and dual time-step subiteration. The characteristic boundary condition was imposed at the far-field boundary. The no-slip iso-thermal boundary condition was applied at the solid surface. Unstructured hybrid meshes consisted of quadrilateral cells inside the viscous boundary layer and isotropic triangular cells for the rest of computation domain was used.

To simulate the jet, the velocity along the control slot on the airfoil surface was imposed as follows:

$$u = V_j \cos(\theta + \Phi),\ v = V_j \sin(\theta + \Phi) \qquad (1)$$

where θ is the boundary surface angle; Φ is the control jet angle; and V_j is the control jet velocity. The control jet velocity is obtained from the non-dimensional control momentum coefficient, $C_\mu (C_\mu = < J > / cq_\infty)$. Here, $< J >= \rho V_j^2 H_j$ is the control momentum; q_∞ is the dynamic pressure; H_j is the jet slot width; and c is the chord length.

To reduce the large computational time required for the unsteady time-accurate solution, a parallel computation algorithm was adopted by splitting the computational domain into local sub-domains using the MeTis library. Because MeTis library assigns each vertex to parallel processors uniquely, parallel communication boundaries are set along the edges of the control volume (dual edges). This type of communication boundary is not efficient for the current time integration method because it requires more communication overhead to get the information from the neighbour vertices surrounding each vertex for the evaluation of residuals and flux Jacobians. In the present study, to avoid this excessive communication time, information about all vertices needed for flux integration was retained in each subdomain at the cost of slightly larger storage resource. Then only two communications are required at each stage of the time integration up to second-order accurate calculation, one for the evaluation of the solution gradients for high-order reconstruction and the other for updating the solution at the boundary vertices at the end of the time integration. Fig. 1 shows the vortices retained for boundary communication under the current vertex partitioning. Mesh partitioning, sorting of the additional vertices for each subdomain, and the generation of the information needed for communication are completed in the pre-processing step and stored.

In order to check the efficiency and the accuracy of the parallel computation, a transonic inviscid flow around an NACA 0012 airfoil was tested. The prescribed freestream Mach number was $M_\infty = 0.8$ and the angles of attack was set at $1.25°$. In Fig. 2(a), comparison of the convergence characteristics between single processor and parallel processors is presented. It shows that independent to the number of processors used, the convergence characteristics of the parallel computation is consistent with that of the single processor when communication was executed at every iteration of the Gause-seidal relaxation. However, when the communication was only performed at the first iteration, the conversence characteristics were deteriorated, and this degradation of convergence was amplified as the number of processors increased. Fig. 2(b) represents the relative speedups achieved for the number of processors used. The results show that good scalability was obtained both with and without communication in the Gause-Seidal relaxation up to eight processors. However, as the number of processors increased, the speedup became poorer with full communication, while the opposite was obtained without full communication.

Fig. 3 shows comparison of the velocity vectors in the vicinity of the trailing edge at $\alpha = 0°$ between the flow without blowing and that with blowing for $V_j = 0.2, L_j = 99.5\%c, W_j = 0.5\%c$. It shows that the perpendicular blowing worked as a virtual flap similar to the Gurney flap, and induces downward turing flow.

In Fig. 4(a), the increment of lift is shown for various blowing jet strengths (prescribed non-dimensional jet velocity in the figure) with $L_j = 99.5\%c, W_j = 0.5\%c$. At a given angle of attact, perpendicular blowing increased the lift without changing the lift curve slope, and the amount of lift increment became larger as the blowing jet strengh increased. Compared to the case without bowing at $\alpha = 4°$, the lift increment was 1.3 times, 1.54 times, and 2.05 times higher for $V_j = 0.05, V_j = 0.1, V_j = 0.2$, respectively. However, blowing was not as effective to suppress the airfoil stall as the Gurney flap, and the stall angle of attack remained similar even though the blowing jet strength increased. Fig. 4(b) represents the effect of blowing jet strength on drag coefficient. When the jet strength increased, the drag coefficient was reduced at the given angle of attack, and this tendency is different from that of the Gurney flap showing drag increase as the flap hight increases. Compared to the drag coefficient without bowing at the $\alpha = 4°$, 0.99 times, 0.96 times, and 0.79 times lower drag coefficients were obtained for

$V_j = 0.05$, $V_j = 0.1$, $V_j = 0.2$, respectively. This drag reduction was the major factor of increasing the lift-to-drag ratio as shown in Fig. 4(d). Fig. 4(c) shows the effect of blowing on the moment coefficient about the quater chord. As for the Gurney flap, there was an increment in the node-down pitching moment as the blowing jet strengh became larger, mostly due to the circulation increment in the vicinity of the trailing edge induced by the downward turning flow.

In the full paper, detailed description about the numerican method and the parallel computation algorithm will be presented. More calculation cases for the virtual flap and the comparison with both Gurney flap and upper surface blowing will also be presented.

REFERENCES

[1] C. S. Jang, J. C. Ross, and R. M. Cummings, (1992). "Computational Evaluation of an Airfoil with a Gurney Flap", AIAA 92-2708.

[2] R. Myose, M. Papadakis, and I. Heron, (1998). "Gurney Flap Experiments on Airfoils, Wings, and Reflection Plane Model", Journal of Aircraft, Vol. 35, pp. 206-211.

[3] V. Venkatakrishnan, (1994). "Parallel Implicit Unstructured Grid Euler Solvers", AIAA Journal, Vol. 32, pp. 1985-1991.

[4] D. J. Mavriplis, (2000). "Viscous Flow Analysis Using a Parallel Unstructured Multigrid Solver", AIAA Journal, Vol. 38, pp. 2067-2076.

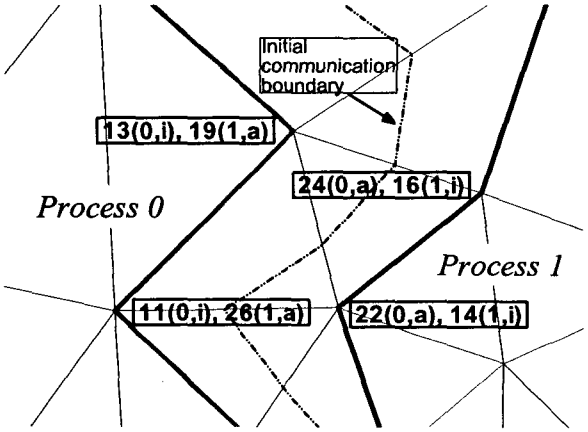
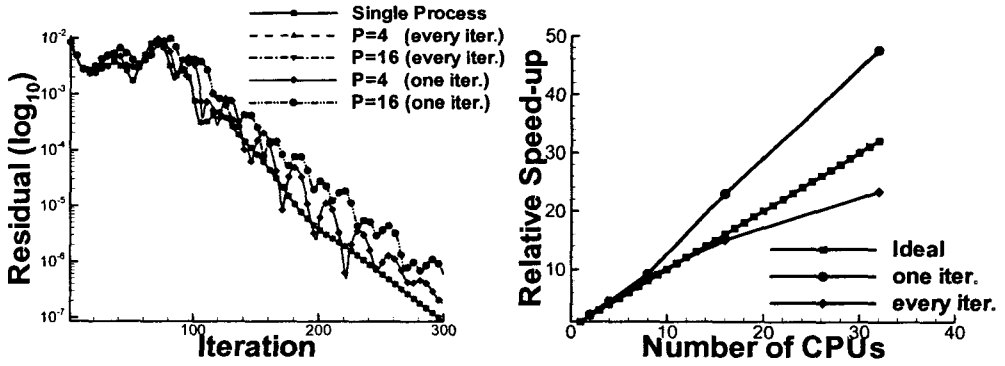


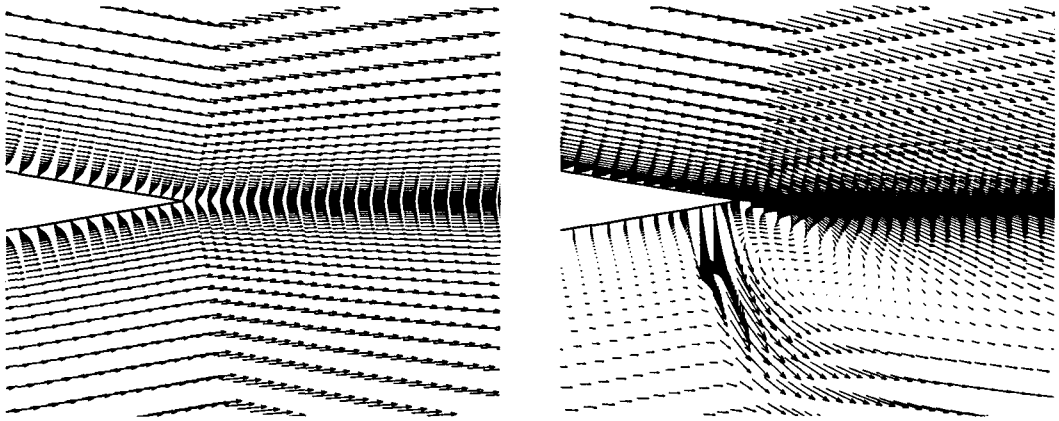
Fig. 1 Vertex partitioning at communication boundary.



(a) Convergence characteristics between single processor and multiple processors

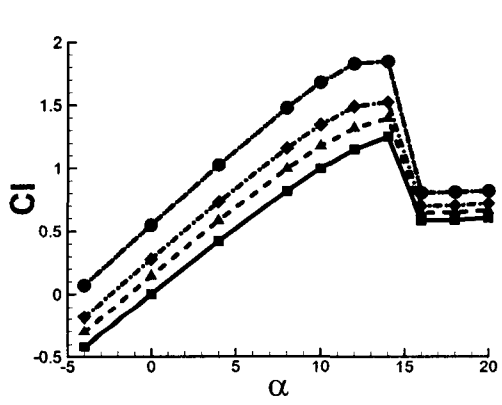
(b) Relative speedups for different number of processor

Fig. 2 Performance check of parallel computation.

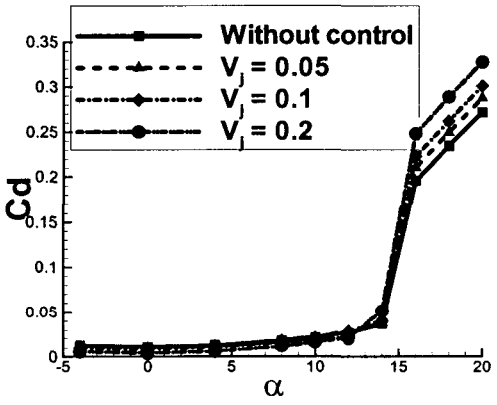


(a) without blowing (b) with perpendicular blowing

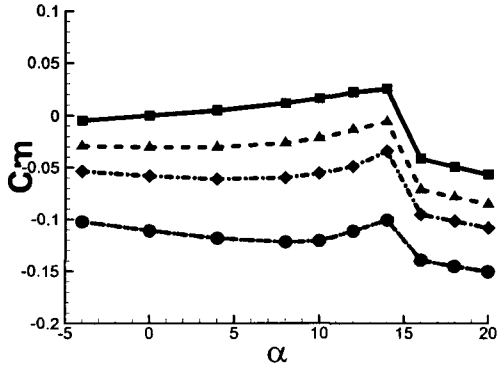
Fig. 3 Comparison of velocity vectors near the trailing edge between flows with and without blowing.



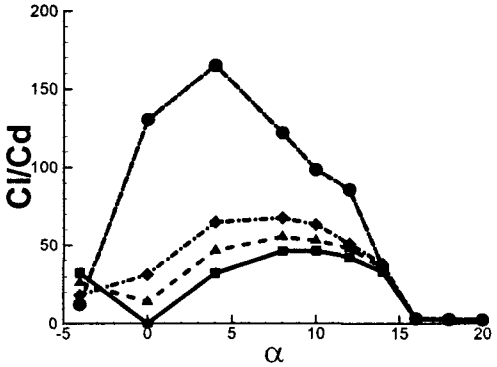
(a) Lift coefficient vs. angle of attack



(b) Drag coefficient vs. angle of attack



(c) Moment coefficient vs. angle of attack



(d) Cl/Cd vs. angle of attack

Fig. 4 Comparison of force and moment coefficients between flows with and with blowing.