Development of Flutter Analysis Tool using Next-Generation CFD (Computational Fluid Dynamics) Algorithms

1st Takaaki Yumitori+1, 2nd Katsutoshi Ishikawa+1,
3rd Keizo Takenaka+1 and 4rd Toshihiko Azuma+2
+1Mitsubishi Heavy Industries, Ltd., 10, Oye-cho, Minato-ku, Nagoya 455-8515, Japan
+2Japan Aircraft Development Corporation/ Churyo Engineering Co.,Ltd.,
Hibiya Kokusai Bldg. 7F, 2-2-3, Uchisawai-cho, Chiyoda-ku, Tokyo, 100-001, Japan

Both high accuracy and high efficiency are strongly required for flutter analysis. To achieve these requirements, MHI developed a flutter analysis tool based on the BCM (Building-Cube Method), which is a next-generation CFD (Computational Fluid Dynamics) algorithm using an advanced Cartesian mesh method.

The BCM has several advantages such as rapid mesh generation, simple data structure for handling complex geometry, adaptive space-time resolution, and easy parallelization with ideal load balance.

The developed tool is validated with the NASA ARROW WING model. Computational time is greatly reduced to less than one–third by applying adaptive inner time iteration technique. Predicted flutter boundary agrees well with experimental data.

Keyword: flutter, CFD, cartesian mesh

1. INTRODUCTION

Flutter must be avoided for safe operation of aircraft, not to cause the structure destruction. Recently, the structure weight saving is necessary for performance improvement, operational and manufacturing cost reduction in aircraft design. This weight saving reduces the structural margin of safety and increases the risk of flutter. It is necessary to carry out the trade-off between structure weight saving and flutter prevention design with high accuracy. Also, since the flutter prevention design is linked with both aircraft aerodynamic shape design and structural design closely, high efficiency is required in design development cycles.

Flutter analysis is a coupled analysis between aerodynamics and structural vibration. The current primary analysis tool is NASTRAN[1)] which is widely applied in the field of aircraft design, and is accepted by the U.S. Federal Aviation Administration (FAA) in type certification. NASTRAN is high efficiency because it uses a linear analysis for aerodynamics, and required computer resource is personal PC level. On the other hand, they cannot evaluate nonlinear aerodynamics such as shockwave and flow separation in transonic regime including cruise condition. Flutter velocity drop caused by nonlinear aerodynamics is widely known as transonic dip, and linear aerodynamic analysis of NASTRAN cannot estimate accurately. Consideration of the transonic dip in actual aircraft design is achieved with an empirical safety margin to the analytical outputs based on experience. Application of the safety margin leads to the weight penalty of the aircraft.

The influence of nonlinear aerodynamics is now evaluated by the flutter wind tunnel test using small scale model. There are so many difficulties in the flutter wind tunnel test. It takes a lot of time and cost for model design, manufacturing and tunnel operation. The small scale model is required to simulate aerodynamic shape, mass and stiffness of the aircraft simultaneously, based on the similarity law, which is not easy task. The required number of the model is large, since the model is going to be broken by flutter during the test.

In recent years, CFD has become matured reasonable to evaluate nonlinear aerodynamics. Steady state analysis to deal with time-averaged flow has become a primary aerodynamic shape design tool. However, unsteady, time-dependent analysis even with a shape deformation, such as flutter, has still huge computational

[+1] takaaki_yumitori@mhi.co.jp

cost, then, there is a problem in the design application in terms of analysis efficiency. Although the computational time can be decreased by using a large-scale parallel computer, pre-processing (Mesh generation, mesh deformation, etc.) and post-processing (visualization, data extraction, etc.) is expected to become a bottleneck.

To solve the problem, the authors developed a flutter analysis tool based on the BCM (Building-Cube Method)[2], which is a next-generation CFD (Computational Fluid Dynamics) algorithm using an advanced Cartesian mesh method.

In this paper, the developed flutter analysis tool and its validation results are explained.

2. OVERVIEW OF ANALYSIS TOOL

BCM based on block-structured Cartesian mesh method is applied to flutter analysis.

BCM has several advantages by its essential strategies. These strategies employed building-up of sub-domains, called ‘Cube’, which has various sizes and equal number of cells. Fig. 1 shows schematic of BCM mesh.

Various sizes cubes enable to adapt to the geometry and flow features. Equal number of cells enables rapid mesh generation, simple data structure for handling complex geometry, and easy parallelization with ideal load balance.

BCM solves three-dimensional Euler and Reynolds averaged Navier–Stokes equations using a cell-centered finite volume method. The HLLEW (Harten-Lax-van Lee-Wada)[3] is used for the numerical flux computations. The third spatial accuracy is realized by Venkatakrishnan’s limiter[4] and MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) scheme. For time integration, the second-order accurate LU-SGS (Lower/Upper Symmetric Gauss-Seidel) implicit method[5] with dual-time stepping using the three-point Euler backward difference is applied. The flow equations, coupled with a second-order linear dynamic solver, are solved using the fully implicit second-order accuracy scheme[6] developed by Melville.

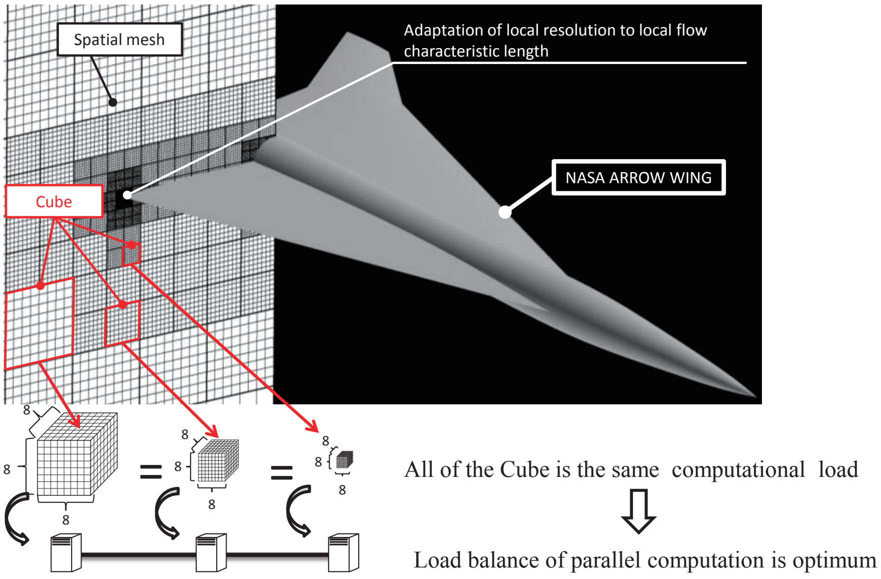

Spatial mesh
Adaptation of local resolution to local flow characteristic length
NASA ARROW WING
Cube
8
8
8
=
8
8
8
=
8
8
8
All of the Cube is the same computational load
⇩
Load balance of parallel computation is optimum

Figure 1: BCM mesh

(1) Treatment of wall boundary condition for BCM

a) Wall boundary condition

Immersed boundary method[7] using Ghost Cell (GC) and Image Point (IP) on the wall boundary is applied. Fig.2 shows the schematic of interpolation to GC from IP. GC is defined as cells in the solid that have at least one neighbor in the fluid, and IP is defined from the wall in the normal direction to 1.75 times of the length of the smallest cell size (larger than cell diagonal length). IP is interpolated using geometric weighting function[8] from 3 x 3 x 3 fluid cells around IP. Interpolation from IP to GC used formulation eq. 1 and eq. 2. Slip velocity and zero wall normal pressure gradient is applied.

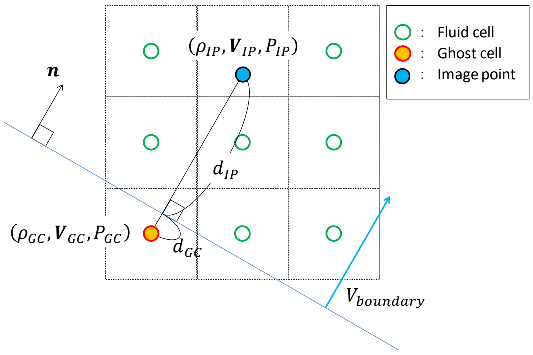


Figure 2: Schematic of interpolation to GC from IP

$$\boldsymbol{V}_{GC} \quad = \boldsymbol{V}_{IP} - \left\{1 + \left(\frac{d_{IP}}{d_{GC}}\right)\right\}(\boldsymbol{V}_{IP} \cdot \mathbf{n})\mathbf{n} \tag{1}$$

$$\left.\frac{\partial P}{\partial \mathbf{n}}\right|_{WALL} = 0 \tag{2}$$

Where $\boldsymbol{V}_{GC}$ is Velocity of Ghost cell , $\boldsymbol{V}_{IP}$ is Velocity of Image point, d_{IP} is distance between wall boundary and Image point, d_{GC} is distance between wall boundary and Ghost cell, and $\mathbf{n}$ is normal vector of wall boundary.

b) Moving wall boundary condition

Deforming mesh at each CFD time-step for unsteady simulation like moving wall can cause inefficiency and troubles such as mesh cross-over or over-skewed meshes. In order to avoid the problem, transpiration boundary method is applied for moving wall boundary. It was reported to predicted flutter boundary successfully[9),10),11)].

Transpiration boundary method is a simple technique to update boundary velocity, acceleration and normal vector for a thin wing small moving or deformation. The boundary condition can be written as follows.

$$\boldsymbol{V}_{GC} \quad = \left(\boldsymbol{V}_{IP} - \boldsymbol{V}_{boundary}\right) - \left\{1 + \left(\frac{d_{IP}}{d_{GC}}\right)\right\}\left\{\left(\boldsymbol{V}_{IP} - \boldsymbol{V}_{bondary}\right) \cdot \mathbf{n}\right\}\mathbf{n} \tag{3}$$

$$\left.\frac{\partial P}{\partial \mathbf{n}}\right|_{WALL} = -\rho_{\mathrm{IP}}\left(\boldsymbol{a}_{boundary} \cdot \mathbf{n}\right) \tag{4}$$

$$\rho_{GC} \quad = \rho_{IP}\left(\frac{P_{GC}}{P_{IP}}\right)^{\frac{1}{\gamma}} \tag{5}$$

Where ρ_{IP} is density of Image Point, $\boldsymbol{a}_{boundary}$ is acceleration of Wall boundary, ρ_{GC} is density of Ghost cell, P_{IP} is pressure of Image Point, P_{GC} is pressure of Ghost cell, and γ is ratio of specific heat.

(2) Adaptive iteration techniques

Aerodynamic and structural equations are coupled as fully implicit second-order scheme. Fig. 3 shows aerodynamic and structural equations coupling method

Aerodynamics and structure are solved independently in a decoupled way. By using inner time iteration, the phase difference between aerodynamics and structure can be eliminated. The approach allows for the use of independent solvers for the aerodynamics and structural equations, but accuracy and efficiency of the solution is affected by the inner time iteration.

In this study, two adaptive inner iteration techniques are developed and applied to improve efficiency of the analysis.

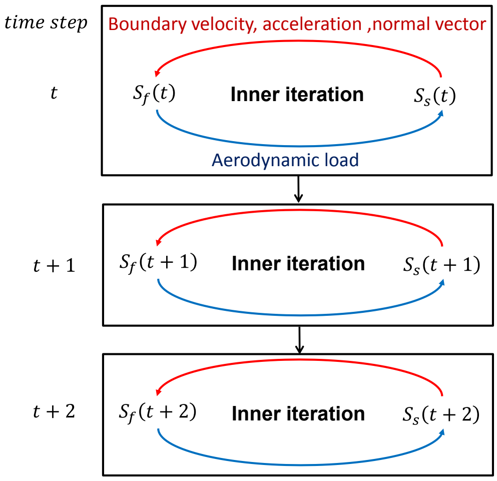


Figure 3: Aerodynamic and structural equations coupling method.

a) Reducing number of inner time iteration

Fig. 4(a) shows schematic of adaptive inner iteration.

We tried to improve the efficiency of calculation by adaptive inner time iteration using the convergence criteria based on the residual of generalized aerodynamics force is applied. Generalized aerodynamics force is shown as follows.

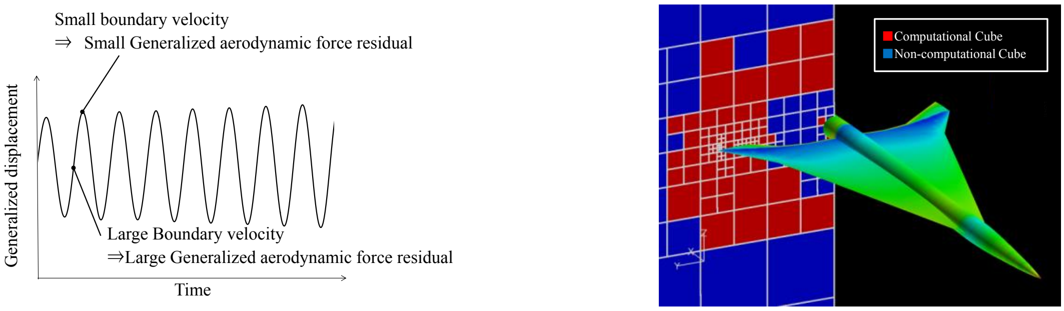
$$\boldsymbol{Q} = \boldsymbol{\Phi}^{\boldsymbol{T}} \boldsymbol{F}(\boldsymbol{\omega}, \boldsymbol{\Phi}) \tag{6}$$

Where $\boldsymbol{Q}$ is generalized aerodynamics force, $\boldsymbol{F}$ is aerodynamics force, $\boldsymbol{\omega}$ is frequency vector, $\boldsymbol{\Phi}$ is modal matrix, and $\boldsymbol{\Phi}^{\boldsymbol{T}}$ is transpose of modal matrix.

When generalized aerodynamic force residual is sufficiently small during inner iteration, BCM generates correct approximate solution, and inner iteration is stopped. Generalized aerodynamic force residual has a characteristic that when boundary velocity is small, generalized aerodynamics force becomes small. So, we introduced stopping tolerance to improve efficiency but also maintain accuracy by taking advantage of this characteristic.

b) Reducing number of computational cubes

Each cube of BCM has different convergence characteristics by its own flow field. Far field cubes have little flow change and converges rapidly. Therefore, BCM cubes are automatically divided into computational cubes and non-computational cubes during inner time iteration, based on residual values of flow variables. Fig. 4(b) shows schematic of computational cube groups and non-computational cube groups.



(a) Reducing number of inner time iteration (b) Reducing number of computational cubes

Figure 4: Adaptive iteration techniques

3. VALIDATION RESULT (2D PITCHING AIRFOIL)

BCM was utilized to simulate a transonic NACA 0012 pitching airfoil[12]. Fig.5 shows BCM mesh around the NACA 0012 airfoil.

As the airfoil oscillates, the shock transitions occur between the upper and lower surfaces of the airfoil. This hysteresis loop is evident in the Cl history plotted in Fig. 6(a). After an initial transient of approximately 1/2 cycle, the solution becomes periodic. At a given angle of attack, the Cl is multi-valued, depending on whether the airfoil is pitching up or down. Fig. 6(b) shows pitching down Cp distribution as an example.

The computed Cl hysteresis loop and Cp distribution are in good agreement with the experimental data.

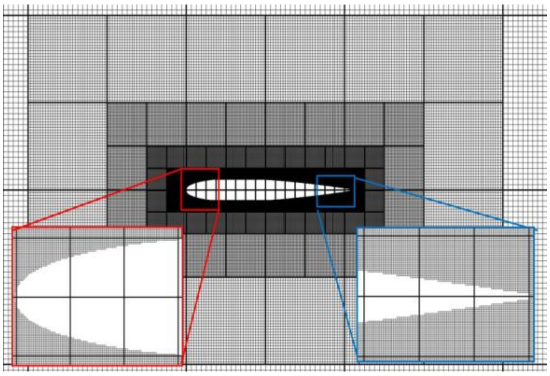
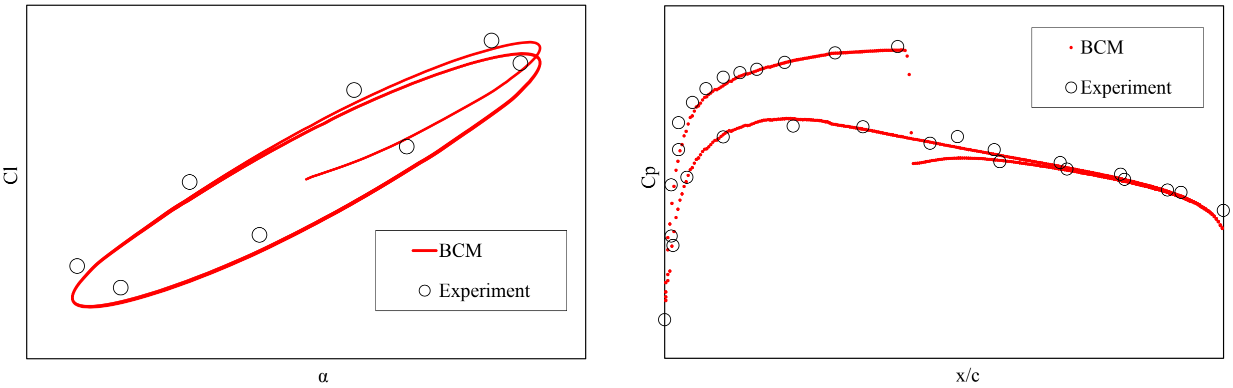


Figure 5: BCM mesh around the NACA 0012 airfoil



(a) Cl hysteresis loop (b) pitching down Cp distribution

Figure 6: Example of validation analysis on a pitching airfoil

4. VALIDATION RESULT (3D FULLTER ANALYSIS)

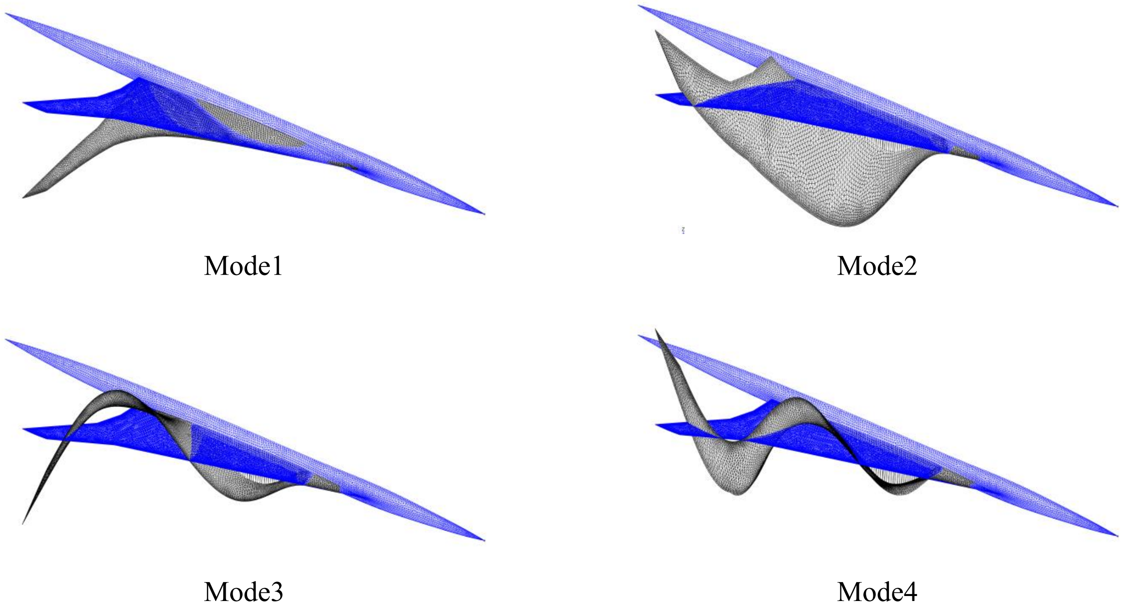
Validation example with NASA ARROW WING[13) is shown in this section.

Fig. 7 shows the first four modal shapes, and Fig. 8 shows BCM cubes around NASA ARROW WING.

Fig. 9 shows the time history of generalized displacement. BCM fixed inner time iteration (fine) was sufficiently converged. On the other hand, BCM fixed inner time iteration (coarse) was not sufficiently converged. BCM adaptive iteration showed good agreement with BCM fixed inner time iteration (fine) by improving efficiency treatments.

Fig.10 shows the flutter boundary. BCM fixed inner time iteration (fine) and BCM adaptive iteration showed good agreement with the experimental flutter boundary. BCM fixed inner time iteration (coarse) was lower than the experimental flutter boundary around 1.0Mach.

Fig.11 shows calculation time. BCM adaptive iteration reduces to less than one-third of the total calculation times.



Mode1

Mode2

Mode3

Mode4

Figure 7: Mode shapes of NASA ARROW WING

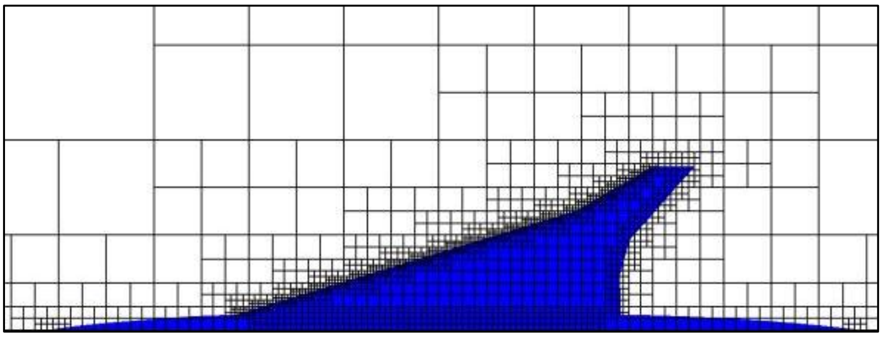


Figure 8: BCM cubes around NASA ARROW WING

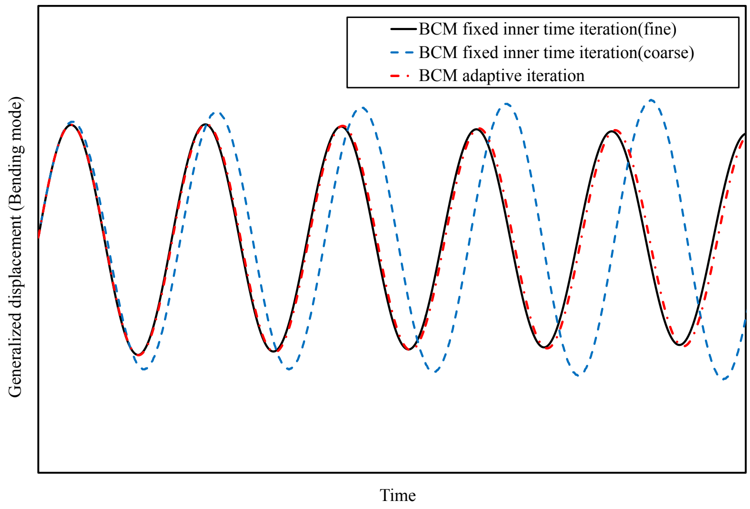


Figure 9: Generalized displacement time history

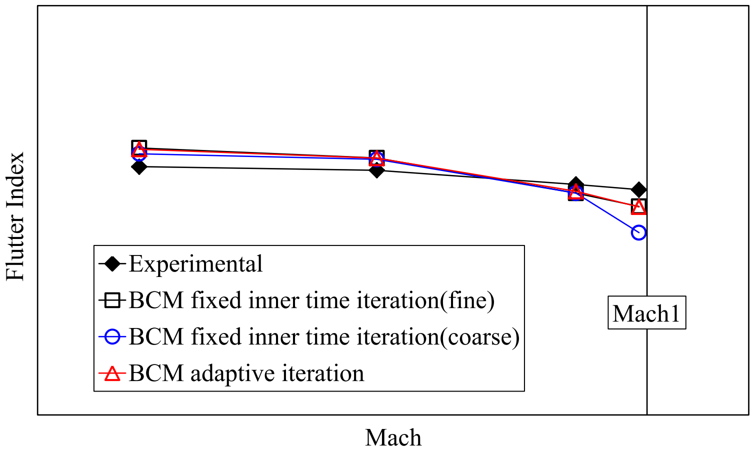


Figure 10: Flutter boundary

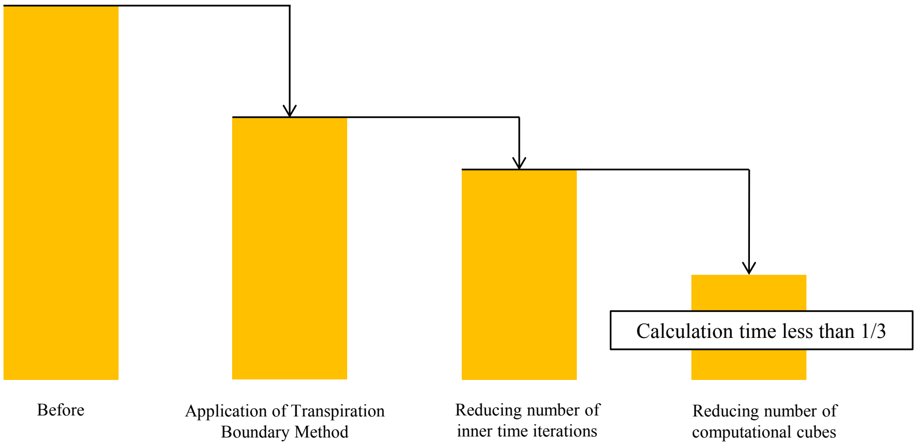


Figure 11: Improving efficiency flutter boundary analysis

5. CONCLUSIONS

The developed tool is validated with the NASA ARROW WING model. Computational time is greatly reduced to less than one–third by applying adaptive inner time iteration technique and reducing number of computational cubes. Predicted flutter boundary agrees well with experimental data.

ACKNOWLEDGMENT

This research has been conducted under auspice of JADC(Japan Aircraft Development Corporation) within HSTP (High Speed TransPort) Program.

REFERENCES

1) MSC/NASTRAN Aeroelastic Analysis USER'S GUIDE.
2) Nakahashi, K. and Kim, L. S. : Building-Cube Method for Large-Scale, High Resolution Flow Computations, AIAA Paper 2004-0423, 2004.
3) Obayashi, S. and Guruswamy, G. P. : Convergence acceleration of a navier-stokes solver for efficient static aeroelastic computations, *AIAA Journal*, 35(6):1134–1141, 1995.
4) Venkatakrishnan, V. : Preconditioned Conjugate Gradient Methods for the Compressible Navier - stokes Equations, *AIAA Journal*, Vol. 29, No.7, pp 1092-1099, 1991.
5) Yoon, S. and Jameson, A. : Lower-Upper Symmetric-Gauss Seidel Method for the Euler and Navier-Stokes Equations, *AIAA Journal*, Vol. 26 No. 9, pp. 1025-1026, .
6) Melville, R. B., Morton, S. A., and Rizzetta, D. P. : Implementation of a Fully-Implicit, Aeroelastic Navier-Stokes Solver, AIAA Paper 97-2039,1997.
7) Mittal, R., Dong, H., Bozkurttas, M., Najjar, F. M., and von Loebbecke, A. : A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *Journal of Computational. Physics*. 227(10), 4825-4852, 2008.
8) Franke, R. : Scattered data interpolation tests of some methods, Math. Compt. 38 181-200, 1982.
9) Stephens, C. H., Aerna, A. S. Jr, and Gupta, K. K. : Application of the Transpiration Method for Aeroservoelastic Prediction Using CFD, AIAA Paper 98-2071, 1998.
10) Yang, S., Zhang, Z., Liu, F., Luo, S., Tsai, H. -M., and Schuster, D. M. : Time-Domain Aeroelastic Simulation by a Coupled Euler and Integral Boundary-Layer Method, AIAA Paper 2004-5377, 2004.
11) Chen, P. C., Zhang, Z., Sengupta, A., and Liu, D. D. : Overset Euler/Boundary-Layer Solver with Panel-Based Aerodynamic Modeling for Aeroelastic Applications, *Journal of Aircraft*, Vol. 46, No. 6 , pp. 2054-2068,2009
12) Landon, R. H. : Compendium of Unsteady Aerodynamic Measurements, AGARD Report No. 702, 1982.
13) Keller, D. F., and Bullock, E. P. : Span Reduction Effects on Flutter Characteristics of Arrow-Wing Supersonic Transport Configurations, NASA Technical Paper 3077,1991.