

## 3차원 오일러 방정식의 계산 효율성 증대를 위한 Adaptive Wavelet 기법의 적용

조 동 욱,<sup>1</sup> 박 경 현,<sup>2</sup> 강 형 민,<sup>3</sup> 이 도 형\*<sup>1</sup>

<sup>1</sup>한양대학교 기계공학과

<sup>2</sup>삼성전기 중앙연구소 PMD 모터제어그룹

<sup>3</sup>한국항공공우주연구원 공력성능팀

### IMPLEMENTATION OF ADAPTIVE WAVELET METHOD FOR ENHANCEMENT OF COMPUTATIONAL EFFICIENCY FOR THREE DIMENSIONAL EULER EQUATION

D.U. Jo,<sup>1</sup> K.H. Park,<sup>2</sup> H.M. Kang,<sup>3</sup> and D.H. Lee\*<sup>1</sup>

<sup>1</sup>Dept. of Mechanical Engineering, Hanyang Univ.

<sup>2</sup>PMD Motor & Control Group, Samsung Electro-Mechanics

<sup>3</sup>Aerodynamics Team, Korea Aerospace Research Institute

*The adaptive wavelet method is studied for the enhancement of computational efficiency of three-dimensional flows. For implementation of the method for three-dimensional Euler equation, wavelet decomposition process is introduced based on the previous two-dimensional adaptive wavelet method. The order of numerical accuracy of an original solver is preserved by applying modified thresholding value. In order to assess the efficiency of the proposed algorithm, the method is applied to the computation of flow field around ONERA-M6 wing in transonic regime with 4th and 6th order interpolating polynomial respectively. Through the application, it is confirmed that the three-dimensional adaptive wavelet method can reduce the computational time while conserving the numerical accuracy of an original solver.*

**Key Words :** 전산유체역학(CFD), 적응 웨이블릿(Adaptive Wavelet), 3차원 오일러 방정식(Three Dimensional Euler Equation)

## 1. 서 론

전산유체역학(Computational Fluid Dynamics, 이하 CFD)에서 높은 신뢰도(high fidelity)의 유동 해석에 대한 요구는 점점 증가하고 있으며 이를 위한 방대한 양의 격자와 함께 정확하고 빠른 계산을 강조한다. 그러나 현재 CFD의 기술 수준은 다수의 격자점과 고차 정확도의 알고리즘이 계산에 상당한 부담을 초래하고 어떤 경우에는 계산 효율성에 대한 고려가 역으로 수치 정확도의 향상을 제한하기도 하는 것처럼 효율성과

정확성 측면에서 모두를 만족시키기에 부족한 부분이 많다.

이러한 문제점을 해결하기 위해 많은 수치적 방법들이 다양한 CFD 분야에 적용되어 왔으며 이들 중, adaptive wavelet 기법은 solution algorithm, feature extraction, data compression 등에 매우 유용한 도구로써 활발하게 연구 및 개발되어 왔다. Adaptive wavelet의 기본적인 접근 방식은 유동장의 wavelet 변환 과정을 통해 wavelet coefficient를 구하는 것이며 이 값들은 유동 값의 변화가 완만한 영역에서 매우 작고 shock, vortex, boundary layer 등과 같이 유동이 급격하게 변화하는 영역에서 큰 값을 갖는다. 대부분의 계산 영역이 유동의 변화가 크지 않고 부드러운 구간인 CFD data의 특성을 바탕으로 유동 해의 중요한 특징들은 wavelet 변환에 의해 자동으로 추출되고 유동의 국소적인 특징을 따라 adaptive dataset이 구성된다. 그 후, dataset에 포함된 유동의 중요 영역들에서만 기존

Received: April 2, 2014, Revised: June 13, 2014,

Accepted: June 13, 2014.

\* Corresponding author, E-mail: dohyung@hanyang.ac.kr

DOI <http://dx.doi.org/10.6112/kscfe.2014.19.2.058>

© KSCFE 2014

CFD 해석 기법을 수행하고 그 외 영역에서는 계산 비용이 낮은 interpolation을 적용함으로써 계산 효율을 높이며 모든 유동장의 계산을 수행하게 된다. 이렇게 유동장의 정보에 따른 해석 방법의 변환을 통해 adaptive wavelet은 계산 효율성에 중요한 장점을 제공할 수 있다.

이와 같은 상당한 이점으로 adaptive wavelet 기법에 대한 많은 연구가 진행되어 왔다. Harten은 쌍곡선형 편미분방정식에서의 불연속적인 해를 구하기 위해 wavelet을 이용한 multi-resolution 기법을 제시했고[1] Holmström은 dataset을 구성하기 위한 interpolation wavelet 변환 알고리즘을 제안했다[2]. 또한, Sjögreen은 압축성 오일러 방정식의 해석을 위한 interpolation wavelet 변환 기반의 multi-resolution 기법을 사용했고[3] Kang 등은 adaptive wavelet에 수정된 thresholding method를 개발, 추가하여 2차원 유동 해석에 적용시켰고 해의 수치 정확도를 보존하며 유동장 계산 효율을 향상시켰다[4,5]. 그리고 D. Wirasat 등은 adaptive wavelet을 이용하여 온도차를 갖는 3차원 cavity 내부의 유동을 시뮬레이션 했다[6].

여기서 최적설계 또는 민감도분석 등을 위한 3차원 해석과 같은 보다 실제적인 응용에 있어, adaptive wavelet 기법은 기존 CFD solver에의 이식이 용이해야 하고 계산 효율을 향상시키며 solver가 가진 수치 정확도를 보존해야 한다. 본 연구에서는 이와 같은 목적을 위해 Kang 등에 의해 제안된 2차원 adaptive wavelet 기법을 3차원 분석으로 확장 개발하여 실제 응용문제에 적용했다. 해당 기법은 4차 또는 6차 interpolation coefficient를 wavelet 기저로 사용하기 때문에 기존 solver에 연결시키는 것이 매우 간단하며 또한, solver의 수치 정확도는 수정된 threshold value에 의해 보존될 수 있다. 개발된 3차원 adaptive wavelet 기법의 성능과 효과를 평가하기 위해서 천음속 영역의 ONERA-M6 날개 주위 유동장 해석에 적용시켜 기존 CFD solver와 본 기법으로 계산한 결과들을 비교 분석하였다.

## 2. Procedure of Adaptive Wavelet Method

### 2.1 Wavelet Transformation

#### 2.1.1 Adaptive Wavelet based on the Interpolating Polynomial

Adaptive wavelet의 주요 아이디어는 유동 패턴 변화의 정도를 주변 cell을 이용한 근사 값과 원래 값의 차이의 크기에 의해 판별하는 것이다. 즉, 변화가 완만한 영역에서는 주변 cell의 값들로 원래 값들을 정확하게 예측할 수 있지만 반대로 급격한 변화 영역에서는 많은 오차를 수반하며 정확한 예측이 어려워진다. Interpolation을 기반으로 한 adaptive wavelet 기법에서 CFD data를 wavelet coefficient로 변환하기 위해 우선, 식 (1)에 나타난 것처럼 dyadic gridset을 가정하고 이것은

식 (2)와 같이 분해될 수 있다. 여기서  $P_{i,j}$ 는 interpolating polynomial의 coefficient를 의미한다.

$$V_l = \{x_{l,k} \in R : x_{l,k} = 2^{-l}k, k \in Z\}, l \in Z \quad (1)$$

$$\begin{cases} \bar{f}_{l+1,2k} = f_{l,k} \\ \bar{f}_{l+1,2k+1} = P_{l+1,2k+1}(f_{l,k-\frac{p}{2}+1}, \dots, f_{l,k}, f_{l,k+1}, \dots, f_{l,k+\frac{p}{2}}) \end{cases} \quad (2)$$

식 (1), (2)와 같은 방법으로 정의된 gridset에서 Interpolating polynomial이 홀수 점에서의 실제 값을 예측하기 위해 사용된다. 식 (3)은 4차 interpolating polynomial을 나타내고 이 때, wavelet coefficient 즉, 실제 값과 예측 값의 차이는 식 (4)와 같이 정의된다.

$$\begin{aligned} \tilde{f}_{l+1,2k+1} = & -\frac{1}{16}f_{l,k-1} + \frac{9}{16}f_{l,k} \\ & + \frac{9}{16}f_{l,k+1} - \frac{1}{16}f_{l,k+2} \end{aligned} \quad (3)$$

$$d_{l,k} = f_{l+1,2k+1} - \tilde{f}_{l+1,2k+1}, \forall k \in Z \quad (4)$$

값의 변화가 완만한 영역에서는 wavelet coefficient의 크기가 매우 작기 때문에 해당 점들은 부분적으로 dataset에서 제외된다. 다시 말해, 변화가 급격한 영역의 점들만이 dataset에 남게 되고 이렇게 구성된 adaptive dataset은 국소적인 유동의 특징을 따라 얻어지며 최대 오차는 식 (5)와 같이 threshold value의 차수 보다 작은 범위에 있게 된다.

$$|d_{l,k}|_{\infty} = |f_{l,k} - \tilde{f}_{l,k}|_{\infty} < O(\varepsilon) \quad (5)$$

#### 2.1.2 Three Dimensional Wavelet Decomposition

본 연구에서 유동 문제를 풀기 위해 지배방정식으로 사용된 3차원 오일러 방정식을 식 (6)에 나타내었다.

$$\begin{aligned} \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = 0 \\ Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \end{bmatrix}, E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho e_t + p)u \end{bmatrix}, F = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (\rho e_t + p)v \end{bmatrix}, \end{aligned} \quad (6)$$

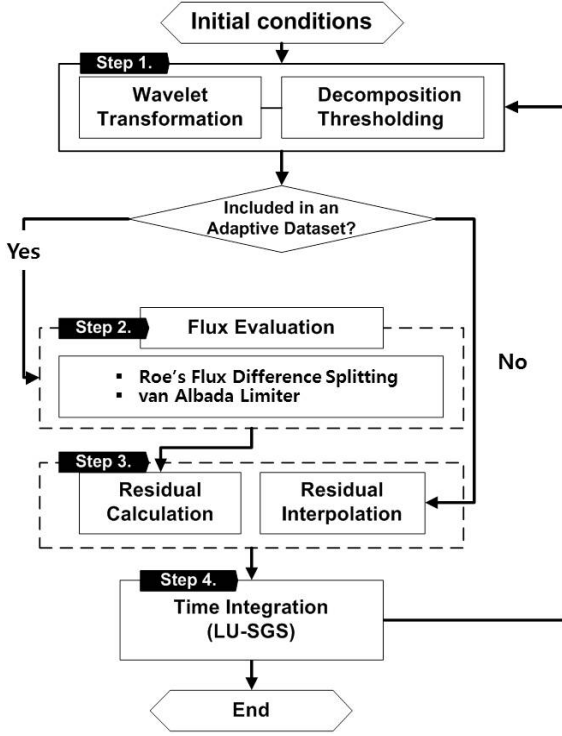


Fig. 1 Overall procedure of wavelet analysis algorithm

$$G = \begin{bmatrix} \rho w \\ \rho w u \\ \rho w v \\ p w^2 + p \\ (\rho e_t + p) w \end{bmatrix}, \quad e_t = \frac{a^2}{\gamma(\gamma-1)} + \frac{1}{2}(u^2 + v^2 + w^2)$$

여기서 모든 변수들 및 방정식은 무차원화 되고 좌표 변환된 지배방정식은 식 (7)과 같다.

$$\begin{aligned} \frac{\partial \bar{Q}}{\partial \tau} &= -\left[\frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} + \frac{\partial \bar{G}}{\partial \zeta}\right] = -R_{i,j,k}^n \\ \bar{Q} &= \frac{Q}{J}, \\ \bar{E} &= \frac{1}{J}[\xi_t Q + \xi_x E + \xi_y F + \xi_z G], \\ \bar{F} &= \frac{1}{J}[\eta_t Q + \eta_x E + \eta_y F + \eta_z G], \\ \bar{G} &= \frac{1}{J}[\zeta_t Q + \zeta_x E + \zeta_y G + \zeta_z G] \end{aligned} \quad (7)$$

Fig. 1은 이와 같은 3차원 오일러 방정식에 adaptive wavelet 기법이 삽입되어 실행되는 전반적인 절차를 나타낸 그림이다. 그 중, 3차원 wavelet decomposition을 위한 dyadic gridset은

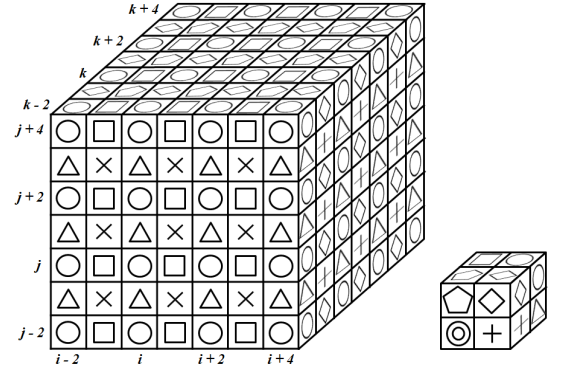


Fig. 2 Three-dimensional dyadic gridset

Fig. 2와 같이 가정한다. 여기서 다해상도 wavelet 변환 시 작 수 점 역할을 하는  $\circ$  cell에서는 상위 level로 올라갈수록 더 성긴(coarse) 격자 수준으로 실제 값이 해당 격자점에 저장되고 그 외의 다른 cell들( $\square$ ,  $\triangle$ ,  $\diamond$ ,  $\times$ ,  $+$ ,  $\hexagon$ ,  $\odot$ )에서는 고차의 interpolation이 사용되어 실제 값을 예측한다. 이 때, 사용되는 4차와 6차 interpolating polynomial에 해당하는 식들을 각각 식 (8), (9)에 나타내었다.

$$\begin{aligned} \square: \tilde{Q}_{i+1,j,k}^n &= \frac{1}{16}(-Q_{i-2,j,k}^n + 9Q_{i,j,k}^n + 9Q_{i+2,j,k}^n - Q_{i+4,j,k}^n) \\ \triangle: \tilde{Q}_{i,j+1,k}^n &= \frac{1}{16}(-Q_{i,j-2,k}^n + 9Q_{i,j,k}^n + 9Q_{i,j+2,k}^n - Q_{i,j+4,k}^n) \\ \diamond: \tilde{Q}_{i,j,k+1}^n &= \frac{1}{16}(-Q_{i,j,k-2}^n + 9Q_{i,j,k}^n + 9Q_{i,j,k+2}^n - Q_{i,j,k+4}^n) \\ \times: \tilde{Q}_{i+1,j+1,k}^n &= \frac{1}{32}(-Q_{i-2,j-2,k}^n + 9Q_{i,j,k}^n + 9Q_{i+2,j+2,k}^n \\ &\quad - Q_{i+4,j+4,k}^n - Q_{i-2,j+4,k}^n + 9Q_{i,j+2,k}^n \\ &\quad + 9Q_{i+2,j,k}^n - Q_{i+4,j-2,k}^n) \\ +: \tilde{Q}_{i,j+1,k+1}^n &= \frac{1}{32}(-Q_{i,j-2,k-2}^n + 9Q_{i,j,k}^n + 9Q_{i,j+2,k+2}^n \\ &\quad - Q_{i,j+4,k+4}^n - Q_{i,j-2,k+4}^n + 9Q_{i,j,k+2}^n \\ &\quad + 9Q_{i,j+2,k}^n - Q_{i,j+4,k-2}^n) \\ \hexagon: \tilde{Q}_{i+1,j,k+1}^n &= \frac{1}{32}(-Q_{i-2,j,k-2}^n + 9Q_{i,j,k}^n + 9Q_{i+2,j,k+2}^n \\ &\quad - Q_{i+4,j,k+4}^n - Q_{i-2,j,k+4}^n + 9Q_{i,j,k+2}^n \\ &\quad + 9Q_{i+2,j,k}^n - Q_{i+4,j,k-2}^n) \\ \odot: \tilde{Q}_{i+1,j+1,k+1}^n &= \frac{1}{64}(-Q_{i-2,j-2,k-2}^n + 9Q_{i,j,k}^n \\ &\quad + 9Q_{i+2,j+2,k+2}^n - Q_{i+4,j+4,k+4}^n \\ &\quad - Q_{i-2,j-2,k+4}^n + 9Q_{i,j,k+2}^n \\ &\quad + 9Q_{i+2,j+2,k}^n - Q_{i+4,j+4,k-2}^n \\ &\quad - Q_{i-2,j+4,k-2}^n + 9Q_{i,j+2,k}^n \\ &\quad + 9Q_{i+2,j,k+2}^n - Q_{i+4,j-2,k+4}^n \\ &\quad - Q_{i+4,j-2,k-2}^n + 9Q_{i+2,j,k}^n \\ &\quad + 9Q_{i+2,j,k+2}^n - Q_{i-2,j+4,k+4}^n) \end{aligned} \quad (8)$$

$$\square: \tilde{Q}_{i+1,j,k}^n = \frac{1}{256} (3Q_{i-4,j,k}^n - 25Q_{i-2,j,k}^n + 150Q_{i,j,k}^n + 150Q_{i+2,j,k}^n - 25Q_{i+4,j,k}^n + 3Q_{i+6,j,k}^n)$$

$$\triangle: \tilde{Q}_{i,j+1,k}^n = \frac{1}{256} (3Q_{i,j-4,k}^n - 25Q_{i,j-2,k}^n + 150Q_{i,j,k}^n + 150Q_{i,j+2,k}^n - 25Q_{i,j+4,k}^n + 3Q_{i,j+6,k}^n)$$

$$\diamond: \tilde{Q}_{i,j,k+1}^n = \frac{1}{256} (3Q_{i,j,k-4}^n - 25Q_{i,j,k-2}^n + 150Q_{i,j,k}^n + 150Q_{i,j,k+2}^n - 25Q_{i,j,k+4}^n + 3Q_{i,j,k+6}^n)$$

$$\times: \tilde{Q}_{i+1,j+1,k}^n = \frac{1}{512} (3Q_{i-4,j-4,k}^n - 25Q_{i-2,j-2,k}^n + 150Q_{i,j,k}^n + 150Q_{i+2,j+2,k}^n - 25Q_{i+4,j+4,k}^n + 3Q_{i+6,j+6,k}^n + 3Q_{i-4,j+6,k}^n - 25Q_{i-2,j+4,k}^n + 150Q_{i,j+2,k}^n + 150Q_{i+2,j,k}^n - 25Q_{i+4,j-2,k}^n + 3Q_{i+6,j-4,k}^n)$$

$$+: \tilde{Q}_{i,j+1,k+1}^n = \frac{1}{512} (3Q_{i,j-4,k-4}^n - 25Q_{i,j-2,k-2}^n + 150Q_{i,j,k}^n + 150Q_{i,j+2,k+2}^n - 25Q_{i,j+4,k+4}^n + 3Q_{i,j+6,k+6}^n + 3Q_{i,j-4,k+6}^n - 25Q_{i,j-2,k+4}^n + 150Q_{i,j,k+2}^n + 150Q_{i,j+2,k}^n - 25Q_{i,j+4,k-2}^n + 3Q_{i,j+6,k-4}^n)$$

$$\circlearrowleft: \tilde{Q}_{i+1,j,k+1}^n = \frac{1}{512} (3Q_{i-4,j,k-4}^n - 25Q_{i-2,j,k-2}^n + 150Q_{i,j,k}^n + 150Q_{i+2,j,k+2}^n - 25Q_{i+4,j,k+4}^n + 3Q_{i+6,j,k+6}^n + 3Q_{i-4,j,k+6}^n - 25Q_{i-2,j,k+4}^n + 150Q_{i,j,k+2}^n + 150Q_{i+2,j,k}^n - 25Q_{i+4,j,k-2}^n + 3Q_{i+6,j,k-4}^n)$$

$$\begin{aligned} \odot: \tilde{Q}_{i+1,j+1,k+1}^n &= \frac{1}{1024} (3Q_{i-4,j-4,k-4}^n - 25Q_{i-2,j-2,k-2}^n + 150Q_{i,j,k}^n \\ &+ 150Q_{i+2,j+2,k+2}^n - 25Q_{i+4,j+4,k+4}^n + 3Q_{i+6,j+6,k+6}^n \\ &+ 3Q_{i-4,j-4,k+6}^n - 25Q_{i-2,j-2,k+4}^n + 150Q_{i,j,k+2}^n \\ &+ 150Q_{i+2,j+2,k}^n - 25Q_{i+4,j+4,k-2}^n + 3Q_{i+6,j+6,k-4}^n \\ &+ 3Q_{i-4,j+6,k-4}^n - 25Q_{i-2,j+4,k-2}^n + 150Q_{i,j+2,k}^n \\ &+ 150Q_{i+2,j,k+2}^n - 25Q_{i+4,j-2,k+4}^n + 3Q_{i+6,j-4,k+6}^n \\ &+ 3Q_{i-4,j-4,k-4}^n - 25Q_{i-2,j-2,k-2}^n + 150Q_{i,j,k}^n \\ &+ 150Q_{i+2,j+2,k+2}^n - 25Q_{i+4,j+4,k+4}^n + 3Q_{i-4,j+6,k+6}^n) \quad (9) \end{aligned}$$

각 cell에서의 값들을 interpolation하게 된 후, 해당 위치에서의 wavelet coefficient는 식 (10)을 이용하여 계산된다.

$$\square: d_{i+1,j,k}^n = Q_{i+1,j,k}^n - \tilde{Q}_{i+1,j,k}^n$$

$$\triangle: d_{i,j+1,k}^n = Q_{i,j+1,k}^n - \tilde{Q}_{i,j+1,k}^n$$

$$\diamond: d_{i,j,k+1}^n = Q_{i,j,k+1}^n - \tilde{Q}_{i,j,k+1}^n$$

$$\times: d_{i+1,j+1,k}^n = Q_{i+1,j+1,k}^n - \tilde{Q}_{i+1,j+1,k}^n$$

$$+: d_{i,j+1,k+1}^n = Q_{i,j+1,k+1}^n - \tilde{Q}_{i,j+1,k+1}^n$$

$$\circlearrowleft: d_{i+1,j,k+1}^n = Q_{i+1,j,k+1}^n - \tilde{Q}_{i+1,j,k+1}^n$$

$$\odot: d_{i+1,j+1,k+1}^n = Q_{i+1,j+1,k+1}^n - \tilde{Q}_{i+1,j+1,k+1}^n \quad (10)$$

### 2.1.3 Modified Thresholding Value

Decomposition을 마친 후, wavelet coefficient는 threshold value와 비교되어 threshold value보다 크다면 해당 cell의 위치는 계산된 CFD 해의 국소적인 특징에 따라 adaptive dataset에 남아있게 되며 반대로, wavelet coefficient가 threshold value보다 작은 경우 해당 cell은 adaptive dataset에 포함되지 않는다. 여기에 decomposition에서의 예측 과정과 thresholding에서의 data 손실로 인해  $O(\varepsilon)$ 의 오차가 추가되며 기존 solver의 수치 정확도를 저해한다. 이  $O(\varepsilon)$  오차의 추가를 방지하기 위해 본 연구에서 thresholding 기법이 절단 오차의 차수를 고려하여 수정된다.

본 연구에서는 2차의 MUSCL 기법이 flux 계산에 적용되었으므로  $(i, j, k)$  cell에서의 flux는 식 (11)과 같다.

$$\begin{aligned} E_{\frac{1}{2}}^n &= E_{\frac{1}{2},real}^n + O(\Delta x^2), \\ F_{\frac{1}{2}}^n &= F_{\frac{1}{2},real}^n + O(\Delta y^2), \\ G_{\frac{1}{2}}^n &= G_{\frac{1}{2},real}^n + O(\Delta z^2) \end{aligned} \quad (11)$$

이 때 식 (7)과 식 (11)의 정의에 따라  $R_{i,j,k}^n$ 는 식 (12)와 같이 유도된다.

$$R_{i,j,k}^n \approx R_{i,j,k,real}^n + O(\Delta x^4) \quad (12)$$

또한 thresholding으로 인한  $O(\varepsilon)$  오차가 식 (13)처럼 해에 추가되며 결론적으로  $R_{i,j,k}^n$ 는 식 (14)와 같이 쓰일 수 있다.

$$\begin{aligned} \tilde{E}_{\frac{1}{2}}^n &= E_{\frac{1}{2}}^n + O(\varepsilon), \\ \tilde{F}_{\frac{1}{2}}^n &= F_{\frac{1}{2}}^n + O(\varepsilon), \\ \tilde{G}_{\frac{1}{2}}^n &= G_{\frac{1}{2}}^n + O(\varepsilon) \end{aligned} \quad (13)$$

$$\tilde{R}_{i,j,k}^n \approx R_{i,j,k}^n + O(\Delta x^2 \varepsilon) \quad (14)$$

그리고 adaptive dataset에서 제외된 점들에서의 residual interpolation 과정 중에 또 다른 유형의 오차가 발생된다. Residual 값들의 계산에 4차의 interpolating polynomial이 사용

되는 경우, 오차의 차수는 약  $O(\Delta x^4)$ 가 된다. 그러므로 dataset에서 제외된 점들로 인한 오차의 order는 식 (15)와 같이 유도된다.

$$\tilde{R}_{i,j,k}^n \approx R_{i,j,k,real}^n + O(\Delta x^4) + O(\Delta x^2 \epsilon) \quad (15)$$

결국, 식 (15)로부터  $O(\epsilon) < O(\Delta x^2)$  조건이 2차 공간 정확도를 보존하기 위해 만족되어야 한다. 따라서 격자 간격을 고려한 기존 threshold value의 전환을 통해  $(i, j, k)$  cell에서의 새로운 threshold value는 식 (16)과 같이 변경되어야 한다.

$$\epsilon' = \min(\epsilon, \Delta x^2) \quad (16)$$

## 2.2 Flux Evaluation & Residual Interpolation

Decomposition부터 thresholding까지의 wavelet 변환 과정으로 모든 계산 영역에 대해 adaptive dataset이 구해지고 flux 값들은 오직 dataset에 포함된 cell에서만 계산된다. 본 연구에서 사용된 오일러 solver는 유한체적법(Finite Volume Method) 기반으로 수치적인 flux 계산에 Roe's FDS(Flux Difference Splitting) method를 이용하여 공간 차분을 수행하고 flux 값들은 모든 방향으로 2차의 정확도를 가지며 계산된다. Adaptive dataset에 포함되지 못한 cell에서는 flux 계산을 수행하지 않고 바로 residual을 예측하게 되는데 이 때, residual 값들의 계산에 식 (17)과 같은 4차 interpolating polynomial을 사용하며 6차의 경우는 식 (9)와 동일한 방식으로 residual을 계산한다.

$$\square: \tilde{R}_{i+1,j,k}^n = \frac{1}{16}(-R_{i-2,j,k}^n + 9R_{i,j,k}^n + 9R_{i+2,j,k}^n - R_{i+4,j,k}^n)$$

$$\triangle: \tilde{R}_{i,j+1,k}^n = \frac{1}{16}(-R_{i,j-2,k}^n + 9R_{i,j,k}^n + 9R_{i,j+2,k}^n - R_{i,j+4,k}^n)$$

$$\diamond: \tilde{R}_{i,j,k+1}^n = \frac{1}{16}(-R_{i,j,k-2}^n + 9R_{i,j,k}^n + 9R_{i,j,k+2}^n - R_{i,j,k+4}^n)$$

$$\times: \tilde{R}_{i+1,j+1,k}^n = \frac{1}{32}(-R_{i-2,j-2,k}^n + 9R_{i,j,k}^n + 9R_{i+2,j+2,k}^n - R_{i+4,j+4,k}^n - R_{i-2,j+4,k}^n + 9R_{i,j+2,k}^n + 9R_{i+2,j,k}^n - R_{i+4,j-2,k}^n)$$

$$+: \tilde{R}_{i,j+1,k+1}^n = \frac{1}{32}(-R_{i,j-2,k-2}^n + 9R_{i,j,k}^n + 9R_{i,j+2,k+2}^n - R_{i,j+4,k+4}^n - R_{i,j-2,k+4}^n + 9R_{i,j,k+2}^n + 9R_{i,j+2,k}^n - R_{i,j+4,k-2}^n)$$

$$\circ: \tilde{R}_{i+1,j,k+1}^n = \frac{1}{32}(-R_{i-2,j,k-2}^n + 9R_{i,j,k}^n + 9R_{i+2,j,k+2}^n - R_{i+4,j,k+4}^n - R_{i-2,j,k+4}^n + 9R_{i,j,k+2}^n + 9R_{i+2,j,k}^n - R_{i+4,j,k-2}^n)$$

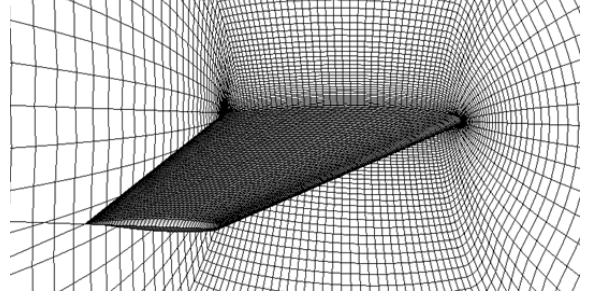


Fig. 3 Grid of ONERA-M6 wing

$$\odot: \tilde{R}_{i+1,j+1,k+1}^n = \frac{1}{64}(-R_{i-2,j-2,k-2}^n + 9R_{i,j,k}^n + 9R_{i+2,j+2,k+2}^n - R_{i+4,j+4,k+4}^n - R_{i-2,j-2,k+4}^n + 9R_{i,j,k+2}^n + 9R_{i+2,j+2,k}^n - R_{i+4,j+4,k-2}^n - R_{i-2,j+4,k-2}^n + 9R_{i,j+2,k}^n + 9R_{i+2,j,k+2}^n - R_{i+4,j-2,k+4}^n - R_{i+4,j-2,k-2}^n + 9R_{i+2,j,k}^n + 9R_{i,j+2,k+2}^n - R_{i-2,j+4,k+4}^n) \quad (17)$$

Residual interpolation을 마친 후, 모든 residual이 구해지면 LU-SGS 기법을 사용하여 전체 계산 영역에 대해 시간 적분을 수행한다.

## 3. 수치해석 및 결과

개발된 3차원 adaptive wavelet 기법의 정확도와 계산 효율을 검증하기 위해 ONERA-M6 날개 주위의 유동장 해석에 본 기법을 적용했다. 사용된 격자의 크기는  $121 \times 61 \times 41$ 로 Fig. 3와 같은 O-H type의 격자를 사용하였으며 유동 조건으로 마하수는 0.8395, 받음각은  $3.06^\circ$ 로 설정하였다[7].

Wavelet 해석 조건에 따른 계산 효율 및 정확도, 압축률을 비교해보기 위해 Table 1, 2와 같이 interpolation order, threshold value, resolution level을 달리하여 계산을 수행했다. 여기서 adaptive dataset이 얼마나 집약적으로 구성되었는지를 볼 수 있는 압축률(compression ratio)을 식 (18)과 같이 정의하여 전체 격자수에 대한 dataset에 포함된 격자수의 비로 나타내었다. 따라서 압축률은 1을 넘지 않으며 값이 작을수록 더 적은 격자로 효과적인 계산을 수행한 것이다. 또한, 계산 효율을 나타내는 time ratio는 wavelet이 적용되지 않은 기존 solver(이하 original solver)로 해석한 시간을 wavelet이 적용된 solver의 해석 시간으로 나누어 표현하였고, 식 (19)와 같이  $L_2$  error를 사용하여 original solver와 wavelet solver간 해석 결과의 차이를 비교해 본 기법의 정확도를 나타내었다.

$$R_{comp} = \frac{N}{S} \tag{18}$$

where,  $R_{comp}$  = compression ratio  
 $N$  = number of remaining grid points  
 $S$  = number of total grid points

$$L_2 error = \frac{\sqrt{\sum_S (O_{i,j,k} - W_{i,j,k})^2}}{S} \tag{19}$$

where,  $O_{i,j,k}$  = result of original solver at  $(i, j, k)$  cell  
 $W_{i,j,k}$  = result of wavelet solver at  $(i, j, k)$  cell  
 $S$  = number of total grid points

모든 case에 걸쳐 3배에서 5.5배까지 빨라진 계산 시간과 평균 86 % 정도의 압축률을 가지고 계산이 수행되었다. 전체적으로 interpolation order와 threshold value의 변화에 따른 결과는 어느 정도 경향성을 나타내지만 resolution level에 따른 결과는 일정하지 않은 패턴을 보인다. 이것은 격자수가 유동 현상을 제대로 표현하기에 충분하지 않은 경우 단순히 level을 올린다고 계산 효율이 향상되지 않는 것으로써 유동의 복잡성 및 격자수의 제한 등으로 인해 각 문제마다 최적의 효율을 얻을 수 있는 level이 존재하는 것으로 보인다.

본 연구에서는 4차 interpolation, threshold  $10^{-5}$ 의 2 level에서 가장 높은 계산 효율을 보이고 6차, threshold  $10^{-6}$ , 1 level의 해석이 가장 정확한 결과를 나타낸다는 것을 알 수 있다. 6차 interpolation의 경우 4차에 비해 대부분의 case에서 다소 낮은 계산 효율을 가지고 있지만 정확도는 10배 가까이 개선 되었

Table 1 Results of efficiency improvement(4<sup>th</sup> interpolation)

Threshold	Level	CPU time (sec)	Time ratio	Comp. ratio	L <sub>2</sub> error
Original	-	7920.60	-	-	-
10 <sup>-5</sup>	1	1430.35	5.538	0.826	1.17E-6
	2	1427.24	5.550	0.822	1.43E-6
	3	1604.11	4.938	0.841	3.58E-6
10 <sup>-6</sup>	1	2289.79	3.459	0.877	8.70E-8
	2	1947.94	4.066	0.876	1.59E-7
	3	2017.90	3.925	0.892	5.81E-7

Table 2 Results of efficiency improvement(6<sup>th</sup> interpolation)

Threshold	Level	CPU time (sec)	Time ratio	Comp. ratio	L <sub>2</sub> error
Original	-	7920.60	-	-	-
10 <sup>-5</sup>	1	1853.34	4.274	0.856	3.70E-7
	2	1746.98	4.534	0.855	3.55E-7
	3	1469.50	5.390	0.858	3.51E-7
10 <sup>-6</sup>	1	2302.60	3.440	0.890	4.12E-8
	2	2497.37	3.172	0.888	1.19E-7
	3	2525.56	3.136	0.892	8.94E-8

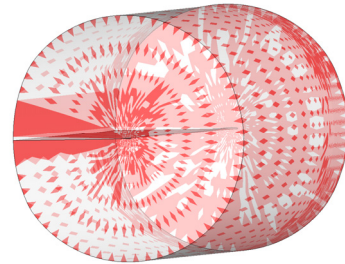
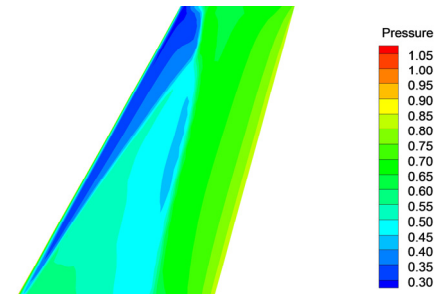
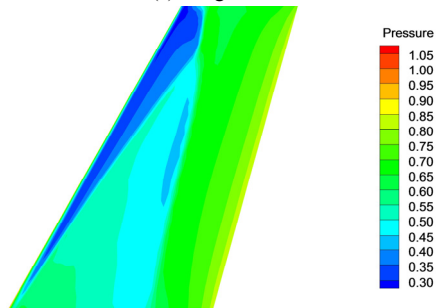


Fig. 4 Adaptive dataset of ONERA-M6 wing flow (2 level)



(a) Original solver



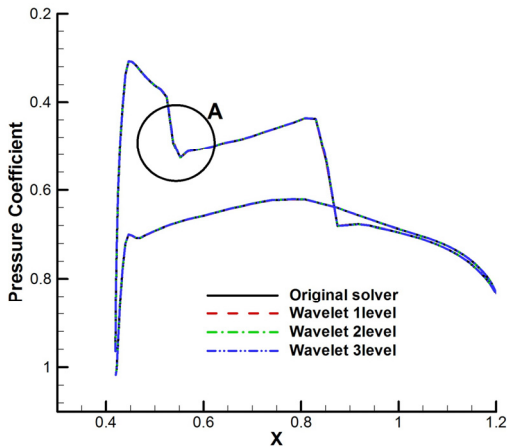
(b) Adaptive wavelet (2 level)

Fig. 5 Pressure contour on the upper surface of wing

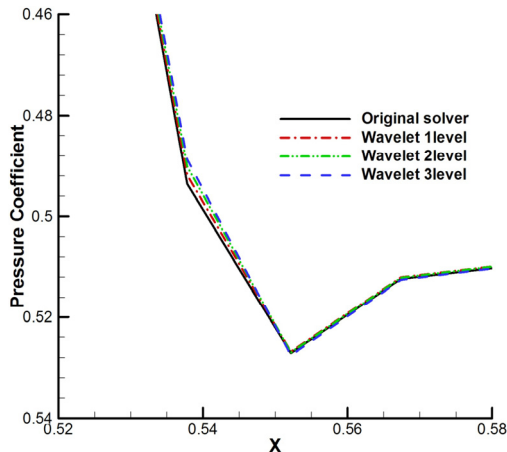
으며 특히, threshold  $10^{-5}$ , 3 level의 case는 4차의 동일한 case 보다 낮은 압축률에도 더 높은 계산 효율과 정확도를 보이며 본 연구 목표에 가장 적합한 case임을 알 수 있다.

Fig. 4는 4차 interpolation,  $10^{-5}$ 의 threshold 조건으로 계산한 결과의 adaptive dataset 분포를 나타낸다. 그림의 우측에서 좌측으로 유동이 흐르며 붉은색으로 표시된 부분이 dataset에 포함된 cell들이고 나머지는 dataset에서 제외된 부분을 의미한다. 그림에서 볼 수 있듯이 많은 cell들이 날개 주변과 하류 쪽에 주로 분포되어 있으며 날개에서 멀리 떨어져 유동의 변화가 작은 영역의 cell들은 dataset에 포함되지 않아 dataset이 유동 해의 국소적인 특징들을 잘 따라서 구성되었다는 것을 알 수 있다.

Fig. 5에 ONERA-M6 날개의 주요한 특징인 lambda shock를 관찰하여 해석 결과의 정확도를 비교하기 위해 날개 윗면에

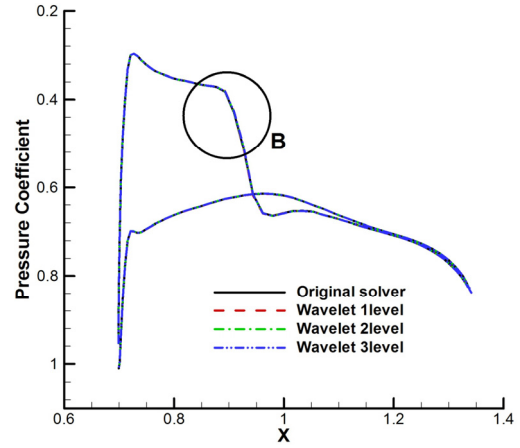


(a) Pressure distribution

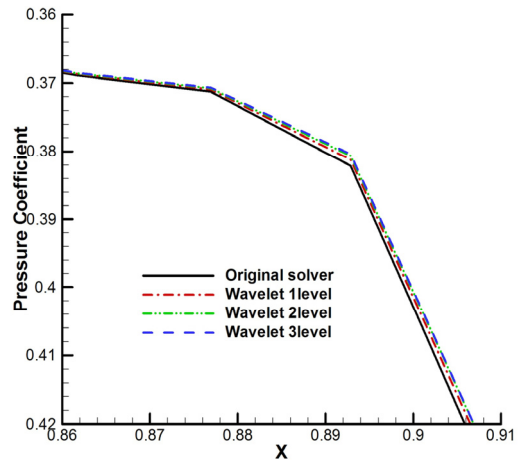


(b) Zoom-in pressure distribution at A region

Fig. 6 Pressure distribution on the surface of wing (at 0.5 of span)



(a) Pressure distribution



(b) Zoom-in pressure distribution at B region

Fig. 7 Pressure distribution on the surface of wing (at 0.8 of span)

서의 압력 contour를 나타내었다. Fig. 5(a)는 기존 solver로 해석한 결과이고 Fig. 5(b)는 4차 interpolation, threshold  $10^{-5}$ 에서 2 level로 wavelet 해석을 수행한 결과이다. Fig. 5(b)에서 날개의 좌측 상단 부분과 중앙의 contour가 조금 일그러져 있는 것을 확인할 수 있지만, Fig. 5(a)와 비교해 전체적으로 동일한 shock의 형태와 contour를 보이며 기존 solver로 해석한 유동 해를 매우 정확하게 표현한다는 것을 알 수 있다.

더 세부적인 비교를 위해 날개 span 길이의 50%, 80%인 지점의 압력 분포를 각각 Fig. 6와 Fig. 7에 그래프로 정리하였다. 각 위치에서 충격파의 위치, 강도 및 압력 분포 등이 기존 결과와 각 level의 wavelet을 적용시킨 해석 결과에서 거의 일치함을 알 수 있고 이를 통해 3차원 adaptive wavelet 기법이 solver의 수치 정확도를 잘 보존하고 있다는 것을 확인할 수 있다.

### 4. 결론

본 연구에서는 3차원 오일러 방정식의 계산 효율성을 증대시키기 위해 3차원으로 확장시킨 interpolation 기반의 adaptive wavelet 기법을 개발하였다. Decomposition, thresholding 과정을 통해 adaptive dataset을 구성하는 wavelet 변환 과정 및 고차의 residual interpolation, 그리고  $\epsilon$ ,  $\Delta x^2$ 의 선택적인 사용으로 solver의 공간 정확도를 보존하는 수정된 thresholding 기법을 도입하여 효율과 정확성을 확보하였다.

개발된 기법을 천음속 영역에서의 ONERA-M6 날개 해석에 적용시켰고 그 결과, wavelet 해석이 기존 해석보다 최대 5.5배 정도 빠른 계산 속도를 제공하는 동시에 lambda shock과 같은 유동의 고유한 특성들을 놓치지 않고 잘 잡아내며

기존 solver로 구한 해와 거의 일치하는 결과를 제공함을 확인하였다. 이로써 wavelet 해석을 통한 계산 시간 단축과 solver의 정확도 유지를 입증했으며 3차원 adaptive wavelet 기법이 유동장의 계산 효율성 향상에 유용한 도구가 될 수 있음을 확인하였다. 추후 더 높은 정확도 보존과 계산속도 향상을 위한 부가적인 기법의 추가와 함께, 격자의 형상 및 간격의 변화에 따른 낮은 압축률 등의 문제점을 보완하기 위한 후속 연구를 진행하여 본 기법의 성능을 더욱 극대화 시킬 수 있을 것이다.

## 후 기

본 연구는 미래창조과학부 첨단 사이언스 · 교육 허브 개발 사업(NRF-2011-0020562) 지원 및 국방과학연구소(과제번호 ADD-12-01-08-13)의 기초 연구비 지원에 의한 연구 결과입니다.

## References

- [1] 1994, Harten A., "Adaptive multiresolution schemes for shock computation," *Journal of Computational Physics*, Vol.115, pp.319-338.
- [2] 1999, Holmström M., "Solving hyperbolic PDEs using interpolation wavelets," *SIAM journal on Scientific Computing*, Vol.21, pp.405-420.
- [3] 1995, Sjögreen B., "Numerical experiments with the multi-resolution scheme for the compressible Euler equations," *Journal of Computational Physics*, Vol.117, pp.251-261.
- [4] 2008, Kang, H., Kim, K., Lee, D. and Lee, D., "Improvement in computational efficiency of Euler equations via a modified Sparse Point Representation method," *Comput. and Fluids*, Vol.37, pp.265-280.
- [5] 2008, Kang, H., Kim, K., Lee, D. and Lee, D., "Improved computational efficiency of unsteady flow problems via the modified wavelet method," *AIAA Journal*, Vol.46, pp.1191-1203.
- [6] 2006, Wirasaet, D. and Paolucci, S., "Three-dimensional Flow in the Differentially Heated Cavity Using an Adaptive Wavelet Method," *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, San Francisco, California, AIAA-2006-3924.
- [7] 1979, Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," *Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04*, AGARD AR 138, May.