삼각격자에 대한 위상학적 개선과정의 확장

맹주성† · 한석영* · 최형일**

(2000년 11월 2일 접수, 2001년 4월 12일 심사완료)

Extension of Topological Improvement Procedures for Triangular Meshes

Joo-Sung Maeng, Seog-Young Han and Hyung-Il Choi

Key Words: Triangular Mesh(비정렬 삼각격자), Clean Up, Smoothing(평활화), Rectangular Geometry(정사각 형상), Multi-Element Airfoil(다요소 익형)

Abstract

This paper describes the extended topological clean up procedures to improve the quality of unstructured triangular meshes. As a postprocessing step, topological improvement procedures are applied both for elements that are interior to the mesh and for elements connected to the boundary and then Laplacian-like smoothing is used by default. Previous clean up algorithms are limited to eliminate the nodes of degree 3,4,8,9,10 and pairs of nodes of degree 5. In this study, new clean up algorithms which minimize the triple connection structures combined with degree 5 and 7 (ie ; 5-7-5, 7-7-5, 7-5-7 etc) are added. The suggested algorithms are applied to two example meshes to demonstrate the effectiveness of the approach in improving element quality in a finite element mesh.

1. 서 론

비정렬격자의 삼각형 격자생성 알고리듬은 크게 Delaunay 삼각화,[1~3] 전방전진기법(Advancing Front Method),[4] Octree 알고리듬[5] 등이 있다. Delaunay 삼각화는 격자생성시간이 짧고 수학적으로 명료하여 자동격자생성에 유리하나 양질의 격자를 얻기 어렵다. 전방전진기법은 양질의 격자를 만들 수 있으나 격자생성시간이 길다는 장단점을 가지고 있다. Octree 알고리듬은 내부 영역에서는 양질의 격자를 얻을 수 있으나 경계 영역에서 양질의 격자를 얻기 어려우며, 특히 당겨진 격자(stretched mesh)를 생성하기가 어렵다. 따라서, Delaunay 삼각화와 전방전진기법의 장점을 취한 격자생성 알고리듬들이 많은 연구자들에 의해 개발되고 있으며,[6,7] 격자생성 후 질을 향상시키는 후처리 알고리듬에 관한 연구도 활발히 진행되고 있다. 후처리 알고리듬에는 크게 평활화(smoothing)[6,8,9]기법과 격자의 위상학적 연결성을 개선하는 clean up과정[10~12] 등이 있다.

평활화에는 간단한 라플라스 평활화[6]와 최적화기반(Optimization-based) 평활화[8,9] 등 여러 알고리듬이 개발되고 있으며, 이 기법은 격자 연결구조를 유지한 상태로 격자 질을 향상시키므로 초기격자의 연결구조가 좋지 않은 경우에는 최적화 기법의 평활화를 적용하더라도 격자 질 향상에는 한계가 있다. 따라서, 후처리 과정으로 평활화 기법과 함께 격자 연결구조를 개선하는 clean up이 필요하게 된다.

내부영역에서는 한 격자점 주위 360°에 6개의 격자를 분포시켜 평균적으로 60°의 각도분포를 가지는 것이 가장 이상적이다. 이와 같이 격자의 연결구조를 가능한한 이상적인 구조로 개선하여

† 회원, 한양대학교 기계공학부
E-mail : jsmaeng@email.hanyang.ac.kr
TEL : (02)2290-0439 FAX : (02)2281-8201
** 한양대학교 대학원 기계공학부

질을 향상시키는 방법을 위상학적 clean up이라 한다. Canann 등[10]과 Corral and Castaneda[11]는 내부영역에 대한 clean up에 대하여 연구하였는데, 그들은 각각 3, 4, 8, 10, 5-5 연결구조와 3, 4, 9, 5-5 연결구조에 제한시켜 적용하였다. 또한 Canann 등은 경계영역에서 특정의 제한된 경우에 대하여도 clean up을 적용시켰다. 이들의 연구결과는 한 격자점을 공유하는 격자의 개수를 5, 6, 7의 연결구조로 바꾸는 것인데, 5와 7의 조합인 5-7-5, 7-7-5 그리고 7-5-7의 바람직하지 않은 연결구조가 생길 수 있다. 따라서 본 연구는 위의 바람직하지 않은 연결구조를 최소화하기 위하여 최대한의 6 연결구조를 갖도록 하는 알고리듬을 보완하여 격자의 질을 향상시키고자 하는 것이다. 또한 보완된 알고리듬을 정사각 형상 및 다요소 익형(multi-element airfoil)을 대상으로 하여 평활화 및 교체(swapping) 그리고 기존의 clean up에 의해 형성된 격자각의 분포를 비교하여 상대적으로 양질의 격자가 생성됨을 확인하였다.

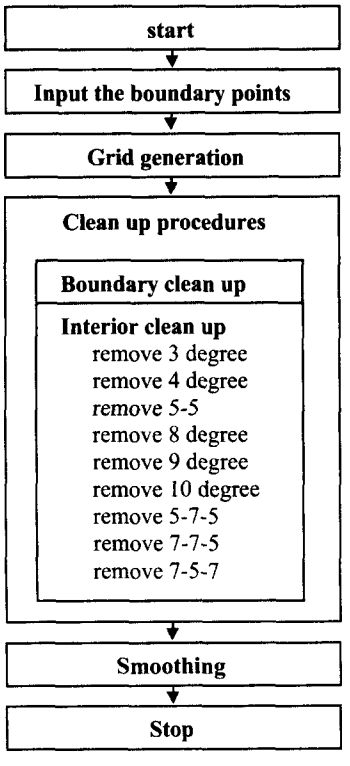


Fig. 1 Flow chart

2. 위상학적 clean up 과정

이상적인 삼각격자는 세 각이 60°인 정삼각형으로 clean up은 내부영역과 경계영역에 구분하여 적용된다. 내부영역에서 한 격자점 주위는 360°로 최적의 격자 개수는 6개로 정해지고 경계영역에서는 경계의 각도에 따른 최적의 개수가 정해진다.

clean up은 격자생성 후에 적용하는 후처리로 전체적인 흐름은 Fig. 1과 같다. 본 논문에서 i는 대상이 되는 격자점을 나타내며 j는 i주위의 격자점을 나타낸다. 그리고 N_i 혹은 N_j 는 격자점 i 또는 j를 공유하고 있는 격자의 개수(degree)를 나타낸다.

2.1 경계영역 clean up 과정

경계영역 clean up을 위해서는 경계영역의 각도(Fig. 2)에 따른 최적의 격자수(N_{op})와 격자점 i를 공유하고 있는 격자들의 개수(N_i)를 비교 검사하여야 한다. Table 1은 경계의 각도에 따른 최적의 격자개수를 나타낸 것이다.

Table 1 Optimal degree for the boundary angles

Angle	Optimal degree(N_{op})
$0^\circ < \theta < 90^\circ$	1
$90^\circ \leq \theta < 150^\circ$	2
$150^\circ \leq \theta < 210^\circ$	3
$210^\circ \leq \theta < 270^\circ$	4
$270^\circ \leq \theta < 330^\circ$	5
$330^\circ \leq \theta < 360^\circ$	6

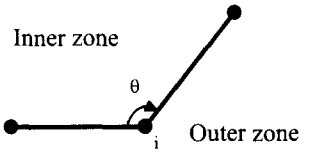


Fig. 2 Boundary angle

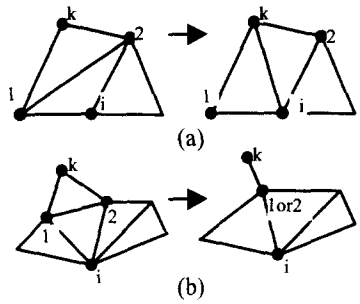


Fig. 3 Clean up at the boundary

i) $N_{op} > N_i$ 연결구조

Fig. 3(a)에서와 같이 격자점 i를 공유하고 있는 격자들 중에서 이웃한 격자의 k격자점 중 N_k가 가장 작은 격자점 k를 선택하여 연결구조를 변경한다. 따라서, 각 격자점을 공유하고 있는 격자의 개수는 $N_i = N_i+1$, $N_1 = N_1-1$, $N_2 = N_2-1$, $N_k = N_k+1$로 변경된다.

ii) $N_{op} < N_i$ 연결구조

Fig. 3(b)에서와 같이 격자점 i를 공유하고 있는 격자들 중 이웃한 격자의 격자점 k를 조사하여 N_k가 가장 큰 격자점 k를 선택하여 합병한다. 여기서 격자점 1과 격자점 2가 경계점이면 합병시키지 않으며, 둘 중 하나만 경계점이면 경계가 아닌 격자점을 삭제한다. 따라서 각 격자점을 공유하고 있는 격자의 개수는 $N_i = N_i-1$, $N_k = N_k-1$, $N_{1or2} = N_1+N_2-4$로 변경된다.

2.2 내부영역 clean up 과정

내부영역 clean up으로 3, 4, 8, 9, 10, 5-5, 5-7-5, 7-5-7, 7-7-5연결구조를 제거하는 알고리듬을 제시하였다.

i) $N_i = 3$ 연결구조

Fig. 4(a)와 같이 i를 격자점 1에 합병한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 $N_i = N_i-1$, $N_2 = N_2-1$, $N_3 = N_3-1$로 변경된다.

ii) $N_i = 4$ 연결구조

Fig. 4(b)에서와 같이 i를 격자점 1에 합병한다. 여기서 i의 주변점인 격자점 j 중에서 $N_j = 6$

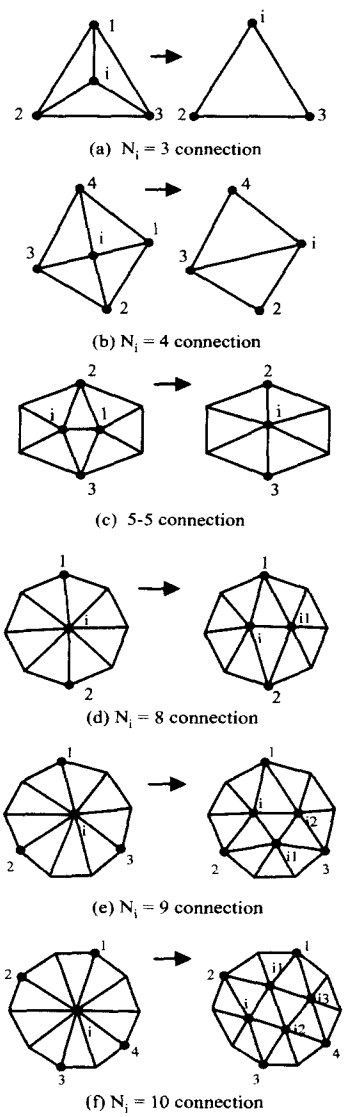


Fig. 4 Old clean up algorithms

인 것을 격자점 1로 선택하거나, $N_j = 7$인 j를 격자점 2로 선택한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 $N_i = N_1$, $N_2 = N_2-1$, $N_3 = N_3-1$, $N_4 = N_4-1$로 변경된다.

iii) 5-5 연결구조

Fig. 4(c)와 같이 5-5 연결구조를 선택하여 i를 격자점 1에 합병한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 $N_i = 6$, $N_2 = N_2-1$, $N_3 = N_3-1$로 변경된다.

iv) N_i =8 연결구조

Fig. 4(d)와 같이 새로운 점 $i1$을 삽입하여 연결구조를 변경한다. 여기서 i 주위의 격자점 j 중 N_j= 5인 j를 찾아 격자점 1로 선정한다. 그러면, 각 격자점을 공유하고 있는 격자의 개수는 N_i= 6, N_{i1}= 6, N_1= N_1+1, N_2= N_2+1로 변경된다.

v) N_i =9 연결구조

Fig. 4(e)와 같이 새로운 점 $i1$, $i2$을 삽입하여 연결구조를 변경한다. 여기서 i 주위의 격자점 j 중 1,2,3위치에 N_j= 5인 격자점 j를 가능한 한 많이 위치시킨다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_i= 6, N_{i1}= 6, N_{i2}= 6, N_1= N_1+1, N_2= N_2+1, N_3= N_3+1로 변경된다.

vi) N_i =10 연결구조

Fig. 4(f)와 같이 새로운 점 $i1$, $i2$, $i3$을 삽입하여 연결구조를 변경한다. 여기서 i 주위의 격자점 j 중 1,2,3,4 위치에 $N_i \leq 5$인 격자점 j를 가능한 한 많이 위치시킨다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_i= 6, N_{i1}= 6, N_{i2}= 6, N_{i3}= 6, N_1= N_1+1, N_2= N_2+1, N_3= N_3+1, N_4= N_4+1로 변경된다.

vii) 5-7-5 연결구조

Fig. 5(a)와 같이 5-7-5연결구조를 선택한다. 이러한 경우는 3가지로 나타나는데 Fig. 5-a-1은 5-5인 연결구조로 해결한다. Fig. 5-a-2는 대각선 교체로 연결구조를 바꾼다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_i= 6, N_1= 6, N_2= 6, N_3= N_3+1로 변경된다. Fig. 5-a-3은 새로운 점 i1을 삽입하여 연결구조를 변경한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_i= 6, N_{i1}= 5, N_1= 6, N_2= 6으로 변경된다.

viii) 7-7-5 연결구조

Fig. 5(b)와 같이 7-7-5 연결구조를 선택한다. 이러한 경우는 7-7 연결구조 주위의 격자점 j 중 N_j= 5인 격자점의 위치에 따른 경우로 3가지 경우가 나타난다. Fig. 5-b-1은 대각선 교체로 연결구조를 변경한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_1= 6, N_2= 6, N_3= 6, N_4= N_4+1로 변경된다. Fig. 5-b-2는 새로운 점 $i1$을 삽입하여 연결구조를 변경한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_{i1}= 6, N_1= 6, N_2= 6, N_3= 6, N_4= N_4+1로 변경된다. Fig. 5-b-3은 새로운 점 $i1$을 삽입하여 연결구조를 변경한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_1= 6, N_2= 8, N_3= 6, N_{i1}= 5로 변

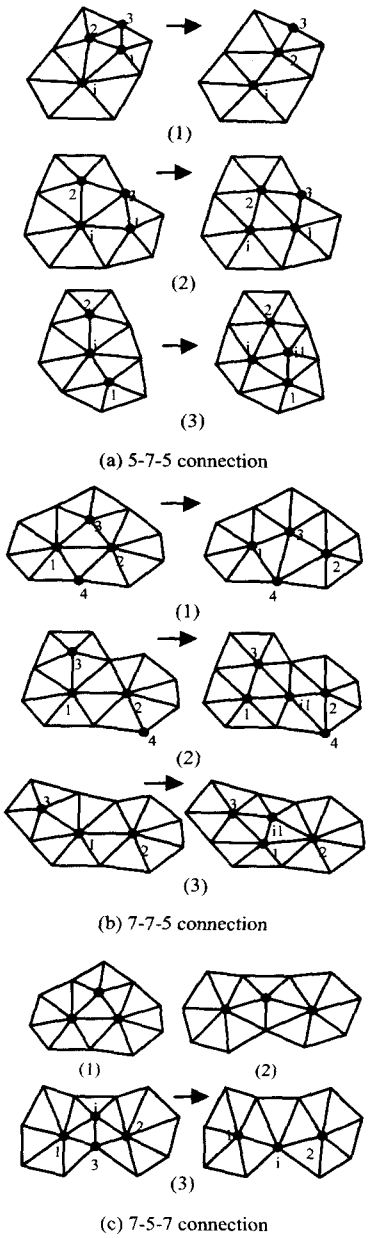


Fig. 5 Added clean up algorithms

경된다. 이 때 N_2= 8은 Fig. 4-d의 예로 연결구조를 변경한다.

ix) 7-5-7 연결구조

Fig. 5(c)와 같이 7-5-7 연결구조를 선택한다. 이러한 연결구조는 2가지 경우가 있다. Fig. 5-c-1의 경우는 7-7-5의 Fig. 5-b-1 경우로 해결한다. Fig. 5-c-2의 경우는 Fig. 5-c-3과 같이 i를 격자점 3에 합병한다. 그러면 각 격자점을 공유하고 있는 격자의 개수는 N_i= N_3+1, N_1= 6, N_2= 6으로 변경된다.

3. 결과 및 고찰

3.1 정사각 형상(Rectangular geometry)

첫번째 예는 정사각형 형상에 대해서 내부점을 난수로 삽입한 후에 Delaunay 삼각화를 이용하여 격자를 생성하였다. Fig. 6(a)는 일반적으로 널리 사용하는 후처리 알고리듬인 라플라스 평활화[6]와 대각선 교체로 격자의 질을 향상시킨 것이다. Fig. 6(b)는 기존의 clean up을 이용하여 격자 질을 향상시킨 것이며, Fig. 6(c)는 본 연구에서 제안한 clean up을 이용하여 격자 질을 향상시킨 것이다. Table 2에서는 위 세 가지 경우에 대해서 격자의 질을 정량적으로 비교하였다. 본 연구에서 제시한 clean up의 결과로 기존의 clean up보다 격자의 수는 증가하였으나 최대각이 18° 정도 감소하였고 최소각은 10° 정도 증대되었다. 격자의 각도별 분포에서는 라플라스 평활화와 대각선 교체, 그리고 기존의 clean up에 비하여 본 연구에서 제시된 clean up이 더 나은 결과를 보여주며, 또한 60° 에 대한 표준 편차 역시 가장 작아 전체적인 격자 질의 향상을 볼 수 있다. 이 예제를 통하여 알 수 있는 것은 첫째, 아주 좋지 않은 격자도 clean up 과정을 통하여 격자 질을 상당히 향상시켰다는 것과 둘째, 기존의 clean up을 보다 개선하였다는 것이다.

3.2 다요소 익형(Multi-element airfoil)

다음은 실제 유동해석 문제에서 자주 등장하는 다요소 익형에 대해 격자를 생성하였다. 격자 생성 알고리듬은 Marcum and Weatherill[6] 이 제안한 알고리듬을 사용하였으며 이 방법은 양질의 격자를 생성하는 알고리듬으로 알려져 있다. 특히 이 알고리듬은 격자 크기에 대한 함수를 선형 보간하여 격자의 크기를 제어하는 알고리듬이 포함되어 있어 양질의 격자를 생성하게 된다. 따라서 이러한 양질의 격자에서 기존의 clean up과 개선된 clean up을 비교하였다.

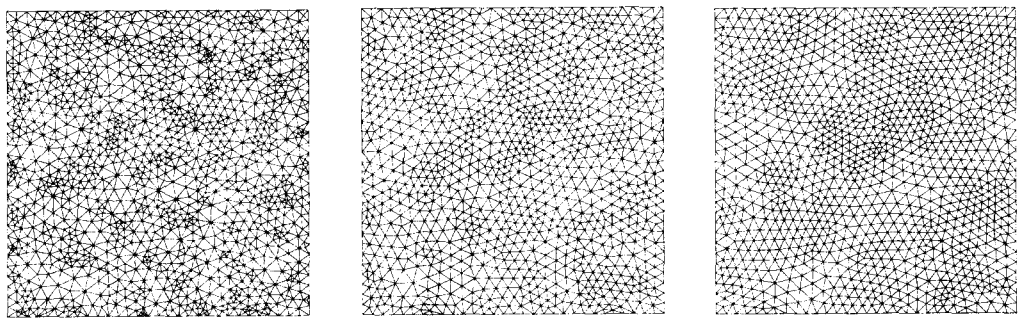
Table 2 Mesh improvement for the rectangular geometry

	Smoothed, swapped grids	Old clean up	New clean up
Node num.	1400	1083	1151
Cell num.	2678	2044	2180
Max. angle	147.29°	129.86°	111.61°
Min. angle	15.90°	16.09°	26.00°
Standard deviation	16.47	8.86	6.92
angle < 30°	1.71 %	0.24 %	0.05 %
angle < 40°	9.40 %	0.91 %	0.31 %
angle < 50°	28.14 %	10.09 %	6.22 %
angle > 70°	24.70 %	11.84 %	6.73 %
angle > 80°	11.24 %	1.79 %	0.83 %
angle > 90°	4.92 %	0.52 %	0.12 %

Table 3 Mesh improvement for the multi-element airfoil

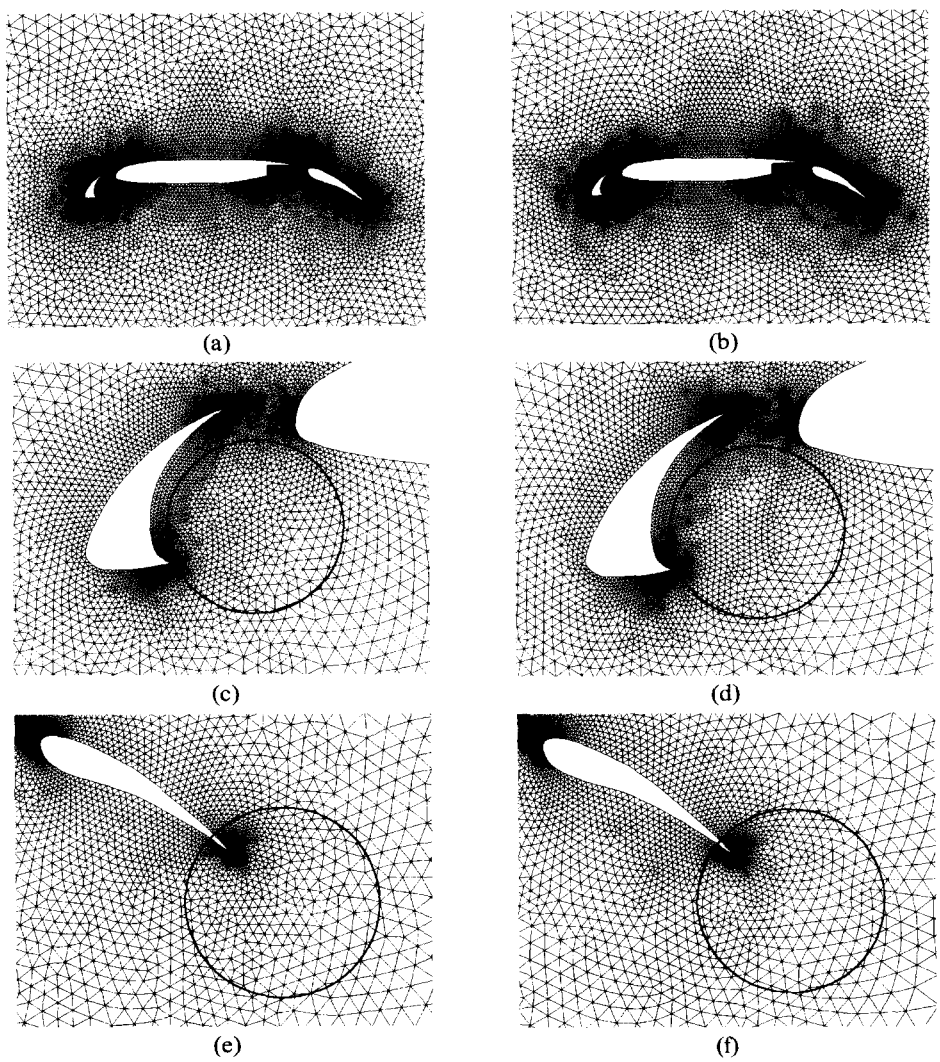
	Old clean up	New clean up
Node num.	18154	18967
Cell num.	35560	37186
Max. angle	91.67°	91.77°
Min. angle	32.90°	35.79°
Standard deviation	5.795	4.755
angle < 30°	0.065 %	0.020 %
angle < 40°	0.718 %	0.219 %
angle < 50°	4.643 %	1.992 %
angle > 70°	5.026 %	2.496 %
angle > 80°	1.667 %	0.555 %
angle > 90°	0.290 %	0.082 %

Fig. 7의 (c),(d),(e),(f)는 (a),(b)의 슬랫(slat)과 플랩(flap)을 확대한 것으로 원으로 표시한 부분을 살펴보면 Fig. 7(c)보다는 Fig. 7(d)의 격자가 규칙성을 가지고 배열하는 것을 볼 수 있다. 즉 격자선의 변화가 보다 더 완만해짐을 볼 수 있다. Fig. 7(e)와 Fig. 7(f)의 비교에서도 같은 결과를 얻을 수 있다. 정량적인 비교는 Table 3에서 볼 수 있는데, 격자의 최소각은 기존 clean up보다



(a) Smoothed, swapped grids (b) Grids after old clean up (c) Grids after new clean up

Fig. 6 Rectangular geometry



(a) (b) (c) (d) (e) (f)

Fig. 7 Old clean up(left) and New clean up(right)

제안된 clean up에서 약 3° 가량 증가하였으나, 최대각은 두 방법 모두 90° 정도이다. 이는 기존 clean up 후에 양질의 격자가 생성되므로 개선된 clean up에서 최대, 최소각의 변화는 거의 없게 된다. 그러나, 격자의 각도별 분포와 이에 따른 60° 에 대한 표준편차가 기존 clean up보다 감소하였으므로 전체적인 격자의 질은 좀더 개선되었음을 알 수 있다.

4. 결 론

본 논문에서는 기존의 clean up에 바람직하지 않은 5-7-5, 7-7-5, 7-5-7 연결구조를 제거하는 알고리듬을 추가한 clean up 알고리듬을 제안하였다. 이 알고리듬은 격자의 연결구조를 근본적으로 변화시켜 6 연결구조의 개수를 최대화함으로써, 기존의 라플라스 평활화 등이 가진 단점인 초기 격자의 질에 따른 격자 질 향상의 한계를 극복하였다. 또한, 초기 격자가 양질의 격자인 경우와 그렇지 않은 경우 공히 기존의 clean up보다 본 연구에서 제안한 알고리듬이 격자의 질을 더 향상시킴을 두 가지 예에서 격자각의 분포와 표준편차의 변화를 통하여 입증하였다.

후 기

본 연구는 2000년도 한양대학교 연구비 지원으로 이루어졌으며 이에 감사드립니다.

참고문헌

(1) Holmes, D. G. and Snyder, D. D., 1988, "The Generation of Unstructured Meshes Using Delaunay Triangulation," *Proceedings of the 2nd Int. Conference on Numerical Grid Generation in CFD*, pp. 643～652.

(2) Rebay, S., 1993, "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm," *J. Comput. Phys.*, Vol. 106, No. 1, pp. 125～138.

(3) 하태성, 김종태, 맹주성, 1994, "정렬배후면 격자계를 이용한 Delaunay 삼각화," 한국항공우주학회지 제 22권 4호, pp. 43～52.

(4) Lo, S. H., 1985, "A New Mesh Generation Scheme for Arbitrary Planar Domains," *Int. J. Numer. Methods Eng.*, Vol. 21, pp. 1403～1426.

(5) Yerry, M. A. and Shephard, M. S., 1984, "Three-dimensional Mesh Generation by Modified Octree Technique," *Int. J. Numer. Methods Eng.*, Vol. 20, pp. 1965～1990.

(6) Marcum, D. L. and Weatherill, N. P., 1995, "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA J.* Vol. 33, No. 9, pp. 1619～1625.

(7) Mavriplis, D. J., 1993, "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," *J. Comput. Phys.*, Vol. 117, pp. 90～101.

(8) Canann, S. A., Tristano, J. R. and Staten, M. L., 1998, "An Approach to Combined Laplacian and Optimization Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes," *Proceedings. 7th Int. Meshing Roundtable.*

(9) Freitag, L. A. and Carl Ollivier-Gooch, 1997, "Tetrahedral Mesh Improvement Using Swapping and Smoothing," *Int. J. Numer. Methods Eng.*, Vol. 40, pp. 3979～4002.

(10) Canann, S. A., Muthukrishnan, S. N. and Phillips, R. K., 1996, "Topological Refinement Procedures for Triangular Finite Element Meshes," *Engineering with Computers*, Vol. 12, pp. 243～255

(11) Corral, R. and Castaneda, J. F., 1998, "Surface Mesh Generation by Means of Steiner Triangulations," *AIAA-98-3013.*

(12) Buscaglia, G. C. and Dari, E. A., 1997, "Anisotropic Mesh Optimization and Its Application in Adaptivity," *Int. J. Numer. Methods Eng.*, Vol. 40, pp. 4119～4136.