한국군사과학기술학회지 제14권 제2호, pp. 313~320, 2011년 4월

학술논문 에너지 부문

SPH에서의 Convex Hull 알고리즘 적용연구

A Syudy on Applications of Convex Hull Algorithm in the SPH

이 진 성* 이 영 신**
Jin-Sung Lee Young-Shin Lee

Abstract

SPH(Smoothed Particle Hydrodynamics) is a gridless Lagrangian technique that is useful as an alternative numerical analysis method used to analyze high deformation problems as well as astrophysical and cosmological problems.

In SPH, all points within the support of the kernel are taken as neighbours. The accuracy of the SHP is highly influenced by the method for choosing neighbours from all particle points considered.

Typically a linked-list method or tree search method has been used as an effective tool because of its conceptual simplicity, but these methods have some liability in anisotropy situations. In this study, convex hull algorithm is presented as an improved method to eliminate this artifact. A convex hull is the smallest convex set that contains a certain set of points or a polygon.

The selected candidate neighbours set are mapped into the new space by an inverse square mapping, and extract a convex hull. The neighbours are selected from the shell of the convex hull. These algorithms are proved by Fortran programs. The programs are expected to use as a searching algorithm in the future SPH program.

Keywords : SPH(Smoothed Particle Hydrodynamics), Stress Particle(응력입자), Velocity Particle(속도입자), Neighbour Particle(이웃입자), Convex Hull

1. 서 론

고폭탄두나 화약이 무기체계에 적용되기 위해서는 요구성능을 만족함은 물론 여러 외부자극에 대하여 안전하면서도 생존성이 보장되어야 한다.

탄두 및 화약에 가해지는 외부자극 중 하나로 hydrodynamic shock 이나 고속의 기계적 충격 등을 들 수 있으며, 이런 고속 충격상태에 대한 재료의 변형거동을 해석하는 수치적 방법으로 현재 많이 사용되고 있는 것은 유한차분법 및 유한요소법에 기초한 유체동력학(Hydrodynamics)이라 할 수 있다. 그러나 요소(Element)와 격자(Grid)를 기초로 계산이 이루어지는 유한요소법과 유한차분법들은 고속 충돌시 발생되는 대

† 2011년 2월 1일 접수~2011년 3월 25일 게재승인
* 국방과학연구소(ADD)
** 충남대학교(Chungnam National University)
책임저자 : 이진성(ljs4168@paran.com)

변형 해석에는 많은 문제점을 가지고 있어 이를 해결하기 위하여 연구되고 있는 것이 무요소 수치해석 방법(Meshless Method)이며, 그 중 최근에 고속 충돌 문제에 많이 적용되고 있는 방법 중의 하나가 바로 Smoothed Particle Hydrodynamics(SPH)이다.

SPH에서 커널함수에 의한 완화된 질량으로 수치해석 하기 위해서는 계산하고자 하는 particle을 중심으로 일정한 영역 주변의 neighbour particle(이웃입자)들을 효율적으로 선택하여 계산에 활용해야 한다. 주변의 neighbour particle들을 선택하는 방법에는 여러 가지가 있으며 현재 대표적으로 사용되고 있는 방법에는 완화길이(Smoothed Length)를 설정하고 계산하고자 하는 입자를 기준으로 완화길이 내에 있는 particle들을 모두 neighbour particle로 선택하여 계산에 활용하는 tree 방법과, linked list에 의하여 neighbour particle 들을 선택하는 방법 등이 있다. 그러나 이들 방법들은 재료의 충돌에 의하여 대변형이 발생되면 입자들 간의 거리가 일정해지지 않으며, neighbour particle들 사이로 다른 particle들이 침투하는 현상이 발생하여 효과적으로 neighbour particle들을 선택할 수 없어 계산의 정확도가 떨어지게 된다. 이러한 단점을 보완하고자 소개된 방법이 convex hull 알고리즘을 이용한 neighbour particle들의 선택방법이다. Convex hull 알고리즘 자체는 기하수학과 관련된 이론으로 다양한 분야에 사용되고 있으나, SPH에서 이웃입자를 선택하는 방법에 적용한 것은 2000년 P. W. Randles와 L. D. Libersky에 의하여 처음 시도되었다[3]. 그들이 Convex hull 알고리즘을 neighbour particle 선택에 적용한 내용을 간단히 살펴보면 한 particle를 중심으로 일정 개수의 주변 particle들을 mapping 과정을 통하여 형성된 공간에 재배치시킨 후 이 공간에서 최외곽으로 구성될 수 있는 볼록 다각형, 즉 convex hull을 이루는 구성점을 선택하면 이 구성점들이 바로 특정 particle 중심에서 가장 근접되어 있는 neighbour particle이 되며, 이 이론을 적용하면 대변형이 심하게 발생하여도 neighbour particle 들을 효과적으로 선택할 수 있다.

이에 본 연구에서는 효율적인 neighbour particle들을 선택하기 위하여 P. W. Randles와 L. D. Libersky가 제안한 convex hull 알고리즘에 의한 neighbour particle 선택방법을 연구하게 되었다[3]. 연구내용으로는 particle 들을 형성한 후 각각의 particle들에 대한 candidate neighbour set(이웃입자 후보군)을 설정하고, candidate neighbour set들이 mapping 공간상에 위치하면 이들 중 convex hull을 이루는 최외곽 구성점을 찾는 일련의 과정을 전산프로그램을 작성하여 수행해 보았다. 수행결과 P. W. Randles와 L. D. Libersky가 발표한 내용과 같이 특정 particle을 중심으로 convex hull을 구성하는 particle들은 가장 근접된 neighbour particle들이 되는 것을 확인할 수 있었으며, 덧붙여 예제 시행시 나타난 알고리즘의 문제점을 프로그램 개발을 통해 해결 할 수 있었다.

본 연구에서 개발된 convex hull 알고리즘에 의한 neighbour particle searching 프로그램은 SPH 프로그램에서 활용이 기대된다.

2. 본 론

가. Neighbour particle 선택

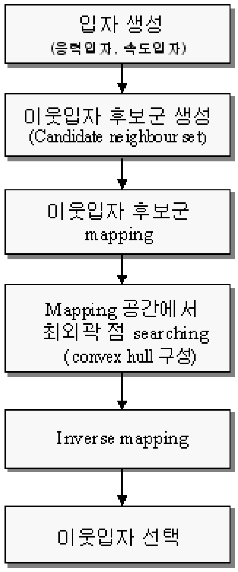

Fig. 1. Convex hull 알고리즘에 의한 neighbour particle 선택과정

주변의 neighbour particle들을 선택하기 위한 전체적인 알고리즘은 Fig. 1과 같은 과정으로 수행된다. 그림에서 convex hull을 적용하기 위해서 첫 번째로 문제영역에 맞는 particle을 선택하고, 각각의 particle들에 대한 주변의 candidate neighbour set를 지정해 준다.

Candidate neighbour set들은 mapping 과정을 통하여 mapping 공간으로 분포하게 되며, 여기에서 convex hull

을 이루는 최외곽 구성점을 찾는다. 최외곽 구성점들은 inverse mapping을 통하여 다시 초기상태의 particle 위치로 환원되어, neighbour particle로서 계산에 적용된다.

나. Convex hull 알고리즘

1) Polygon 형성

Convex hull이란 전체 particle을 포함하는 최소 convex 집합을 말하며, searching algorithm으로 convex를 이용하여 여기에서 convex hull을 이루는 particle들을 neighbour particle로 구하는 것이다.

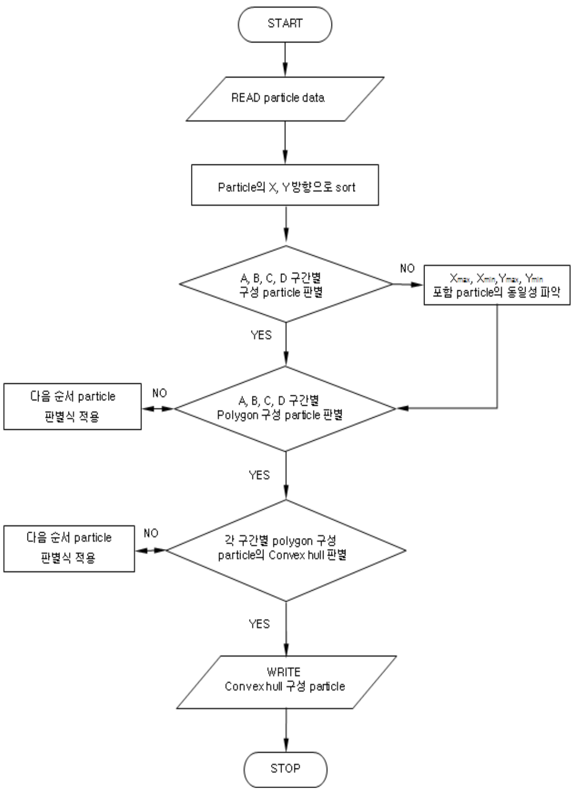

Fig. 2. Convex hull algorithm flow chart

Fig. 2의 알고리즘을 살펴보면, convex hull 알고리즘은 다음과 같은 두 단계로 나눌 수 있다.

첫 번째는 하나의 particle 주변의 neighbour 후보 particle들을 mapping하여 새로운 좌표계를 구성했을 때, 전체 particle들 가운데 최외곽의 polygon을 형성하는 점들을 선택하는 것이다.

두 번째로는 형성된 polygon에서 convex hull 알고리즘을 통해서 polygon에 포함된 concave를 구성하는 particle들을 제거하는 작업을 한다. 이 작업을 통하여 polygon은 convex hull로 재구성된다.

Polygon은 Fig. 3에 나타난바와 같이, 전체 particle들 중 외곽에 위치하는 particle들로 구성되어 있으며, polygon을 구성하는 particle들을 찾기 위한 작업으로, 이 과정에서는 전체 particle들을 X, Y 좌표축에 대하여 sort 작업을 한 후 최대, 최소값을 갖는 particle을 선택하여, 이를 기준으로 전체를 4 구간으로 나누어 최대, 최소 particle을 포함하는 polygon을 형성하는 작업을 수행한다. 전체 4 구간은 A, B, C, D 구간으로 나누어 각각의 polygon 형성 조건을 설명하였다

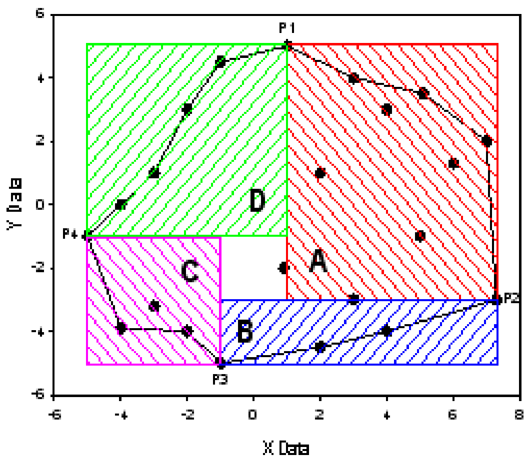

Fig. 3. 구간별 particle set

가) A 구간

A 구간의 구성 particle은 Fig. 3에서 Ymax를 가진 P1 particle의 X data 이상의 X 값을 가지고, Xmax를 갖는 P2 particle의 Y 값을 초과하는 Y 값을 가지는 particle 집합으로 이루어진다. 이 집합에서의 polygon 연결은 Y 축 최대값을 가진 P1 particle을 시작점으로 하며, 전체적으로 정렬된 결과로부터 P1 particle의 Y 값보다 다음순서의 작은 값을 취하는 particle과 P1 particle의 X 값 다음으로 큰 값을 가진 particle이 동시에 고려된다. 여기서 polygon 형성 particle 선택기준은 X, Y 한쪽방향만 다음 값을 가지는 것으로는 안 되고 X, Y의 sort 결과가 “AND”로 적용되어야 한다. A 구간에서의 이와 같은 particle 선택과정은 Xmax를 갖는 particle의 Y 값보다 바로 위 Y 값을 갖는 particle에서 종료되고 Xmax를 갖는 particle은 B 구간 polygon 검색의 시작점이 된다.

－ A 구간 범위
P : Particle
P1(X1, Ymax), P2(Xmax, Y2)
A 구간 X 범위 : X1 $\leq$ X $<$ Xmax
A 구간 Y 범위 : Ymax $\geq$ Y $>$ Y2
A 구간 범위 : 전체 particle 중 X, Y 값이 X 범위, Y 범위에 있는 particle
(X 범위 particle $\cap$ Y 범위 particle)

나) B 구간

B 구간의 구성 particle은 Xmax를 가진 P2 particle이 가지는 Y 값 이하의 Y 값을 가지고, Ymin을 갖는 particle의 X 값을 초과하는 X 값을 가지는 particle 집합으로 이루어진다. 이 집합에서의 polygon 형성은 P2 particle을 시작점으로 하며, 전체적으로 정렬된 결과로부터 P2 particle의 X, Y 값보다 다음순서의 작은 값을 취하는 particle이 선택된다. B 구간에서의 이와 같은 particle 선택과정은 Ymin을 갖는 particle 바로 위 Y 값을 갖는 particle에서 종료되고 Ymin을 갖는 P3 particle은 C 구간 polygon 검색의 시작점이 된다.

－ B 구간 범위
P : Particle
P2(Xmax, Y2), P3(X3, Ymin)
B 구간 X 범위 : Xmax $\geq$ X $>$ X3
B 구간 Y 범위 : Y2 $\geq$ Y $>$ Ymin
B 구간 범위 : 전체 particle 중 X, Y 값이 X 범위, Y 범위에 있는 particle
(X 범위 particle $\cap$ Y 범위 particle)

다) C 구간

C 구간의 구성 particle은 Ymin을 포함하는 P3 particle이 가지는 X 값 이하의 X 값을 가지고, Xmin을 갖는 P4 particle의 Y 값 미만의 Y 값을 가지는 집합으로 이루어진다. 이 집합에서의 polygon 연결은 Ymin을 포함하는 P3 particle을 시작점으로 하며, 전체적으로 정렬된 결과로부터 P3 particle의 Y 값보다 바로 다음순서의 큰 값을 취하는 particle과 P3 particle의 X 값 다음순서의 작은 값을 가진 particle이 동시에 고려된다. C 구간에서의 이와 같은 particle 선택과정은 Xmin을 갖는 P4 particle 바로 아래 particle에서 종료되고 X축 최소값을 가진 P4 particle은 D 구간 polygon 검색의 시작점이 된다.

－ C 구간 범위
P : Particle
P3(X3, Ymin), P4(Xmin, Y4)
C 구간 X 범위 : X3 $\geq$ X $>$ Xmin
C 구간 Y 범위 : Ymin $\leq$ Y $<$ Y4
C 구간 범위 : 전체 particle 중 X, Y 값이 X 범위, Y 범위에 있는 particle
(X 범위 particle $\cap$ Y 범위 particle)

라) D 구간

D 구간의 구성 particle은 Xmin을 가진 P4 particle이 가지는 Y 값 이상의 Y 값을 가지고, Y 축의 최대값을 갖는 particle의 X 값 미만의 X 값을 가지는 집합으로 이루어진다. 이 집합에서의 polygon 연결은 X 축 최소값 P4 particle을 시작점으로 하며, 전체적으로 정렬된 결과로부터 P4 particle의 Y 값보다 바로 다음순서의 큰 값을 취하는 particle과 P4 particle의 X 값 다음순서의 큰 값을 가진 particle이 동시에 고려된다. D 구간에서의 이와 같은 particle 선택과정은 Y 축 최대값을 갖는 P1 particle 바로 전 particle에서 종료된다.

－ D 구간 범위
P : Particle
P4(Xmin, Y4), P1(X1, Ymax),
D 구간 X 범위 : Xmin $\leq$ X $<$ X1
D 구간 Y 범위 : Y4 $\leq$ Y $<$ Ymax
D 구간 범위 : 전체 particle 중 X, Y 값이 X 범위, Y 범위에 있는 particle
(X 범위 particle $\cap$ Y 범위 particle)

마) 각 구간이 합쳐진 경우

지금까지의 A - D 구간처럼 전체 particle들을 sorting 하여 4 구간으로 나누어 polygon을 형성하는 particle 선택방법에 추가하여, 전체 4 구간 중 한 구간이 없어지는 경우를 가정하여 보면, 없어지는 구간은 지금까지의 particle 선택방법에 의하면 문제가 해결되지 않는다.

없어지는 구간이 A 구간으로 가정하자. A 구간은 먼저 P1(X1, Ymax)과 P2(Xmax, Y2)가 명확히 구분되어야 한다. 그러나 P1과 P2 particle의 해당조건이 동일한 particle에 해당되는 경우, A 구간의 polygon 형성 particle들은 없다. 또한 P1은 P2와 동일하므로 A 구간을 구성하는 범위조건에 위배가 되어 해를 구할 수

없다. 이에 P1과 P2가 같은 경우, 별도의 조건을 만들어 프로그램이 진행할 수 있도록 해야 한다. 이와 같은 경우는 A 구간뿐만 아니라 B, C, D 구간에도 해당이 되며 이에 추가조건을 삽입하여야 한다. 본 프로그램에서는 이와 같은 경우 P1을 A 구간의 유일한 polygon 형성 particle로 남겨서 구간 자체는 유지하게 하였고, B 구간의 시작은 다시 P1을 시작점으로 사용하게 하여, polygon 형성 particle들을 선택하는 과정에서의 P1은 중복처리 하였다. 이와 같이 중복 처리되어 2개로 표시된 P1 particle은 convex hull searching 과정에서 최종적으로 하나의 P1로 나타난다. 이와 같은 과정을 통해 기존의 알고리즘에 나타난 문제점을 해결할 수 있었다.

2) Convex hull searching

하나의 polygon에 대한 convex hull을 찾는 것은 Sklansky(1972), Akl and Toussaint(1978)에 의해 사용되어진 vector cross-product rule을 이용하였다. 이 방법은 polygon을 구성하는 particle 간의 벡터를 시계 방향 또는 반시계 방향으로 회전을 시켜서 concave를 이루는가를 검색하는 방법으로 세 particle 간의 각도를 판단 기준으로 삼는다. 이 과정에서는 전체 particle이 아닌 polygon을 구성하는 particle만을 이용하며, 이 과정을 통하여 convex hull을 이루는 polygon으로 다시 구성되어진다.

검색에 필요한 particle의 수는 polygon 구성 particle 개수보다 하나 더 많으며, 이는 검색의 마지막 부분에서 첫 번째 particle이 다시 이용되기 때문이다.

진행 순서는 다음과 같다.

1 step : P = (Xj+1 - Xj)(Yj+2 - Yj+1) - (Yj+1 - Yj)(Xj+2 - Xj+1)

2 step : P ＜ 0 인 경우, j+2 particle은 검색의 끝점

3 step : P ＜ 0 인 경우, 현재 j 보다 한 점 전진하여 j 위치 부여(한 점씩 이동)하고, 1 step으로 돌아간다.

4 step : P ＞ 0 인 경우, j 점은 고정된 상태에서 현재 j + 1을 삭제, 다음 점으로 전진하여 j + 1 위치 부여하고 1 step으로 돌아간다.

이와 같은 과정은 Fig. 4에서 flow chart로 잘 나타나 있으며, 여기서 얻은 convex hull 구성 particle들은 찾고자 하는 neighbour particle들이며, 이를 다시 inverse mapping을 통하여 초기 좌표계로 옮겨서 사용하거나 mapping 초기에 부여한 번호를 이용해 mapping 전의 좌표계 값으로 변환시킬 수 있다.

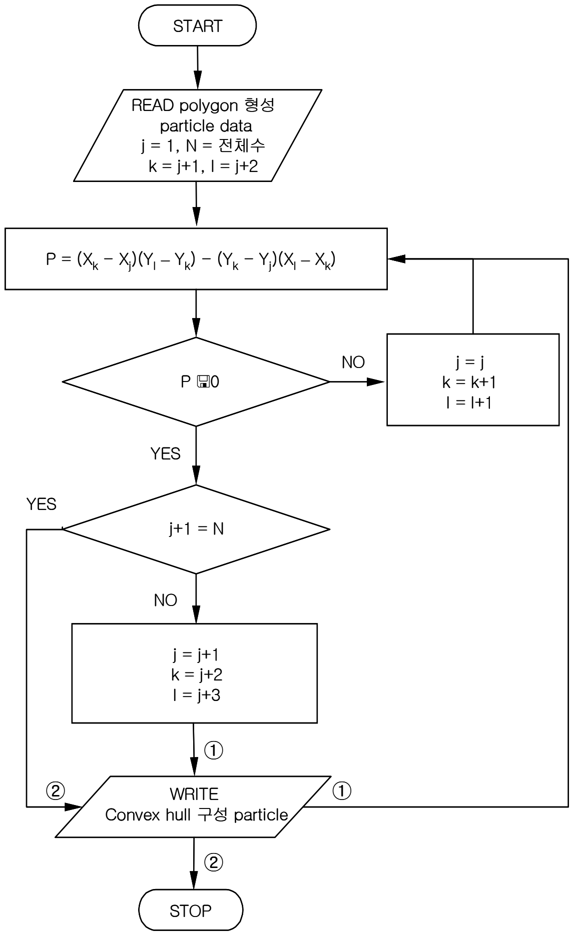

Fig. 4. Convex hull searching algorithm flow chart

3) 예 제

개발된 convex hull searching program을 가지고 다음과 같은 예제를 통하여 검증작업을 실시하였다. 예제는 particle들이 mapping coordinate에 분포되었다고 가정하였고, 예제 두가지를 통해 각각 다른 문제부분을 검증하였다.

가) 예제 1

예제 1은 기본적인 convex hull 알고리즘의 진행과정을 설명하기 위한 것으로 먼저 Fig. 5의 전체 particle 중 X, Y 방향의 최대, 최소 particle을 sorting 하여 구

하고, Ymax를 갖는 particle을 기점으로 하여 Y값이 작은 particle을 검색한다.

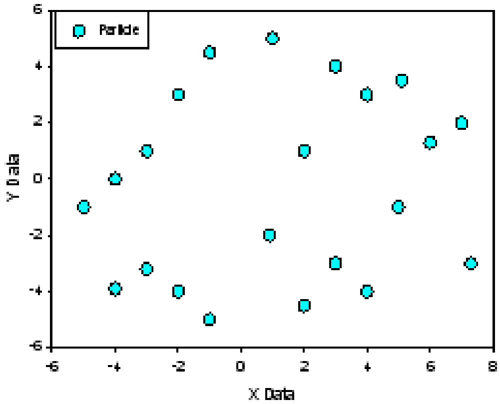

Fig. 5. 전체 particle set

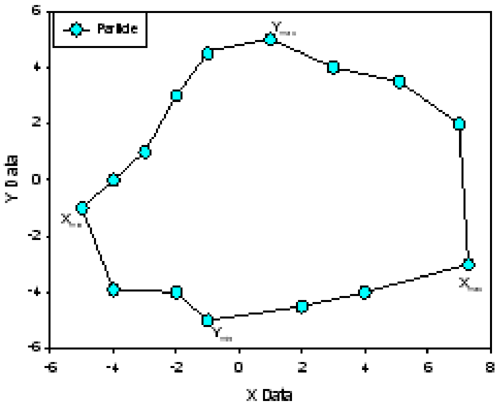

Fig. 6. Polygon 형성 particle

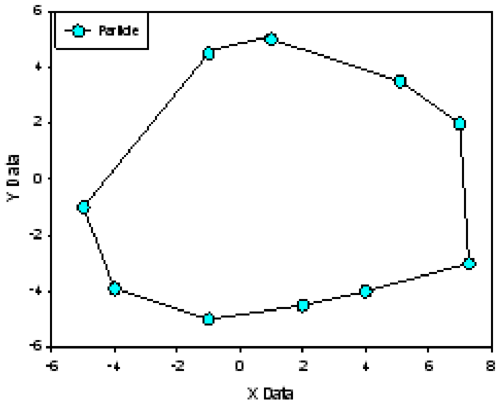

Fig. 7. Convex hull 형성 particle

검색의 동시 추가조건은 Ymax를 갖는 particle의 X값 이상의 값을 가지는 particle을 검색한다. Polygon을 형성하는 particle들의 집합은 Fig. 6에 나타나 있다. Convex hull 형성 particle의 선택결과는 Fig. 7에 나타나 있다. 이와 같은 과정을 통하여 convex hull을 갖는 particle을 찾을 수 있었다.

나) 예제 2

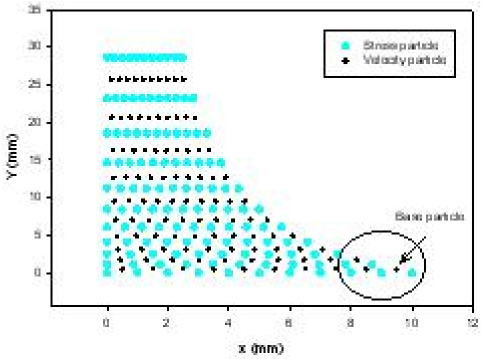

Fig. 8. 변형된 particle 모델

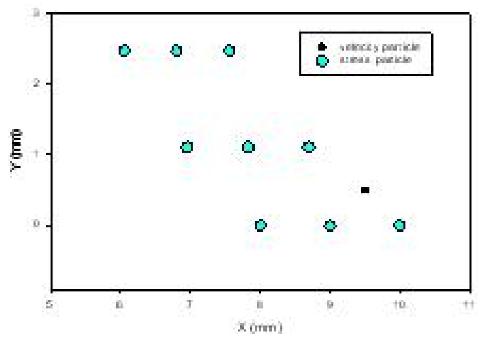

Fig. 9. 변형된 가장자리 particle 모델

예제 2는 임의의 응력에 의해 변형된 좌표계에서 한 velocity particle의 neighbour particle을 구하는 과정을 나타내고 있다. 지금까지 임의의 데이터를 사용하여 convex hull을 구성하는 particle들을 찾은 것과 달리 실제적인 접근으로 변형된 형태의 particle에 대하여 convex hull을 적용하여 개발된 convex hull 알고리즘을 평가해 보았다. Fig. 8은 봉재(rod)가 수직으로 충돌이 일어난 경우에 변형된 형태를 축대칭으로 나타내는 것으로 생각할 수 있으며, Fig. 9에서 표시한 것과 같이 변형이 심하게 발생되는 가장자리 부분의 velocity particle을 중심으로 stress particle neighbour set을 구해 보고자 한다. Fig. 8의 가장자리 부분을 확대해서 표현하면 Fig. 9와 같으며, 한개의 velocity particle을 중심으로 9개의 stress particle이 분포하는 형태이다.

Fig. 9에서 선택된 velocity particle을 중심으로 주변의 stress particle들에 대하여 mapping을 수행하면 Fig. 10과 같이 mapping 공간에서 분포하게 된다.

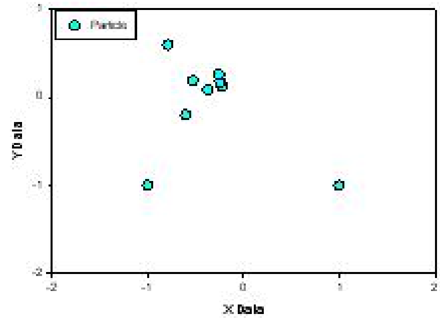

Fig. 10. 전체 particle set

Fig. 11은 mapping된 particle에 대하여 본 연구에서 개발된 convex hull 알고리즘을 적용하여 polygon을 형성하는 particle set을 구한 것을 나타내고 있으며, 이를 기준으로 convex hull을 이루고 있는 구성점을 계산하면 Fig. 12와 같다. Fig. 12에서 알 수 있듯이 4개 구성점으로 convex hull을 구성하고 있으며 이를 통하여 변형이 심한 가장자리 부분에서는 4개의 neighbour particle set을 가지고 있는 것으로 나타나고 있다.

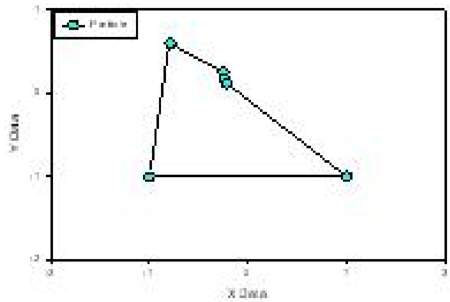

Fig. 11. Polygon 형성 particle

최종적으로 velocity particle 주변의 neighbour stress particle의 위치를 구하기 위하여 convex hull 이루고 있는 구성점으로부터 inverse mapping을 통하여 구한 값이 Fig. 13이다. 예제 2를 통하여 본 연구에서 개발된 convex hull 알고리즘을 실제적인 문제에 가까운 모델을 적용한 결과, 주변의 neighbour particle searching 원활하게 수행됨을 확인할 수 있었다.

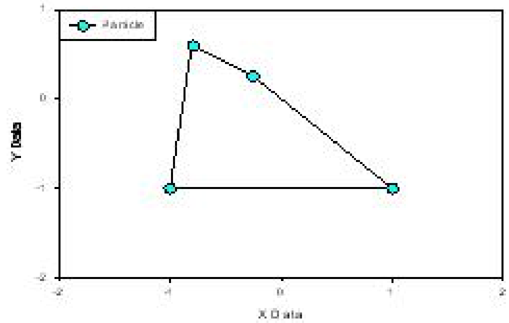

Fig. 12. Convex hull 형성 particle

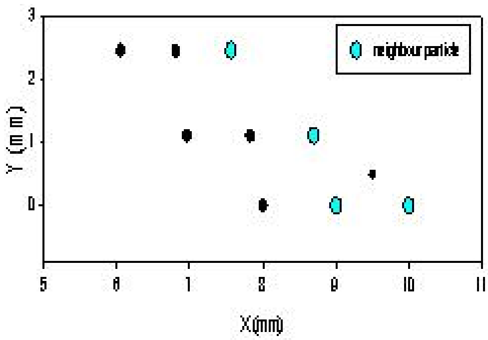

Fig. 13. Neighbour particle

3. 결 론

SPH(Smoothed Particle Hydrodynamics)에서 neighbour particle searching algorithm으로 사용하기 위해서 convex hull algorithm을 이용한 프로그램을 개발하였다. 이는 기존의 tree method나 linked list method보다 neighbour 선택에서 개선된 방법이다. 이와 같은 Convex hull algorithm의 개발을 통하여 최신의 searching algorithm을 확보하였으며, source code를 개발함으로서 이를 통하여 다양한 문제에 적용할 수 있으며, code를 이루는 sub module을 점진적으로 개선할 수 있게 되었다. 또한 개발된 프로그램의 검증을 통하여 프로그램의 유효성을 확인할 수 있었다.

알고리즘을 개발하는 과정에서 기존에 나타나지 않는 문제점을 발견하고 이를 보완하는 프로그램 작업

을 통하여, 기존보다 안정적인 프로그램을 작성할 수 있었다.

마지막으로 프로그램을 neighbour searching 예제에 적용하여 neighbour searching 진행에 있어서 전반적인 과정을 살펴볼 수 있었고, 응력에 의해 변형된 위치의 particle들에서도 정확한 neighbour searching이 이루어지고 있음을 확인할 수 있었다.

Reference

[1] Lucy, L. B., "A Numerical Approach to Testing of the Fission Hypothesis", The Astronomical Journal, 82, pp. 1013～1024, 1977.

[2] Gingold, R. A. and Monaghan, J. J., "Smoothed Particle Hydrodynamics : Theory and Application to Non-spherical Stars", Mon. Not. Roy. Asto. Soc., 181, pp. 375～389.

[3] P. W. Randles and L. D. Libersky., "Normalized SPH with Stress Points. International Journal for Numerical Methods in Engineering", 48, pp. 1445～1462, 2000.

[4] Qin-Zhong Ye., "A Fast Algorithm for Convex Hull Extraction in 2D Images", Pattern Recognition Letters 16, pp. 531～537, 1995.

[5] J. W. Swegle, S. W. Attaway, M. W. Heinstein, F. J. Mello, D. L. Hicks., "An Analysis of Smoothed Particle Hydrodynamics", SAND93-2513., 1994.