



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

임의의 점으로부터 3차원 곡선까지 최단거리 계산 알고리즘 향상

An algorithm for the calculation of the shortest distance from
an arbitrary point to a 3-dimensional curve



지도교수 남종호

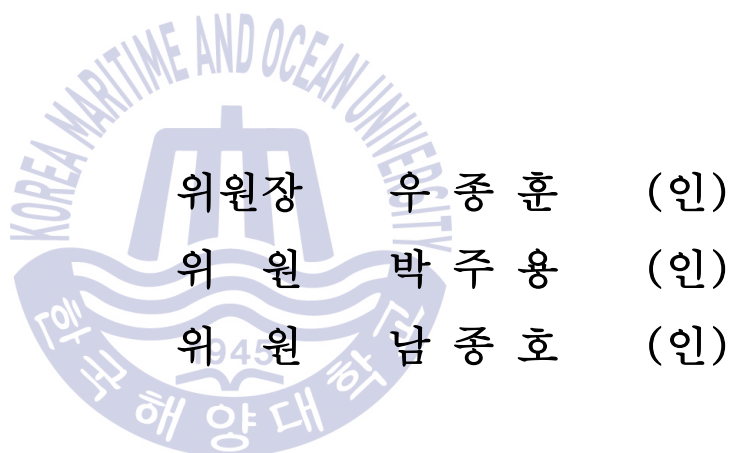
2015년 8월

한국해양대학교 대학원

조선해양시스템공학과

황 윤 석

本 論 文 을 黃 允 碩 의 工 學 碩 士 學 位 論 文 으 로 認 准 함 .



2015년 8월

한국해양대학교 대학원

목 차

List of Tables	v
List of Figures	vii
Abstract	viii

1. 서 론

1.1 연구배경 및 필요성	1
1.2 연구내용	6

2. 주요 기술에 대한 기본 이론

2.1 개요	7
2.2 OpenGL 그래픽 라이브러리	7
2.3 GUI	8
2.4 NURBS	9
2.4.1 NURBS 설명	9
2.4.2 Control Point	10
2.4.3 Knots	11
2.4.4 Degree	12
2.4.5 균일 곡선과 비균일 곡선 설명 및 비교	12
2.5 최단거리 기법	13
2.5.1 이분법(Bisection method)	14
2.5.2 활선법(Secant method)	14

2.5.3 Muller 법(Muller' s method)	15
2.5.4 고정점 반복법(Fixed-point interaction)	16
2.6 Newton' s 기법	17
2.6.1 Newton' s(Newton-Raphson) 기법	17
2.6.2 최단거리 기법 비교 및 Newton-Raphson사용 배경	20

3. Newton-Raphson 법을 활용한 최단거리 검색

3.1 선형 모델링의 NURBS 곡선	21
3.1.1 라이노(Rhino) 선형 모델링	21
3.1.2 선도(Lines)	22
3.2 곡선과 임의의 점과의 관계 분석	23
3.3 NURBS곡선에 Newton-Raphson 기법 적용	24
3.3.1 균일 곡선과 비균일 곡선에 Newton-Raphson 적용	24
3.3.2 구간별 NURBS곡선에 Newton-Raphson 적용	27
3.4 Line Searches and Backtracking 알고리즘	28

4. 최단거리 알고리즘 향상

4.1 프로그램 순서도	30
4.2 원인파악 및 분석	31
4.2.1 Newton-Raphson기법 $f'(t_n)$ 분석	31
4.2.2 초기값에 대한 원인분석	31
4.2.3 균일 곡선과 비균일 곡선 초기값 비교	32
4.3 원인분석에 따른 해결방안	33
4.3.1 곡선의 길이에 따른 등분	33
4.3.2 최대, 최소 범위 설정 및 값 탐색	37
4.3.3 Newton-Raphson에 adaptive value K 설정	38

4.4 향상된 알고리즘 적용	38
5. 결 론	
5.1 결 과	40
5.3 향후과제	41
참고문헌	42



List of Tables

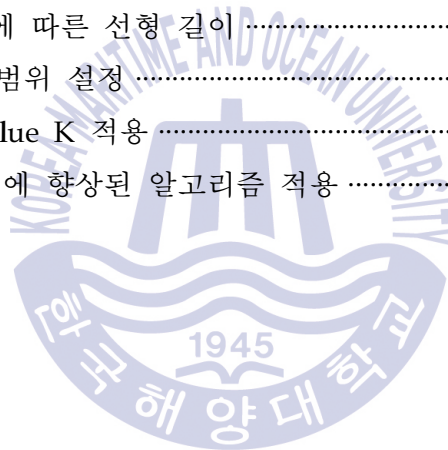
Table.1 Intersection, Offset, Projection	5
Table.2 Muller' method, Secant method, Newton's method, Bisection method, Fixed-point interaction	20
Table.3 균일 곡선에 대한 Newton' s 결과표	25
Table.4 불균일 곡선에 대한 Newton' s 결과표	26
Table.5 불균일 곡선에 향상된 알고리즘 적용 결과표	39



List of Figures

Fig. 1 OSNAP	3
Fig. 2 Connecting Surface and Surface	3
Fig. 3 Refit 기능 수행 전 곡선	4
Fig. 4 Refit 기능 수행 후 곡선	4
Fig. 5 NURBS Equation	9
Fig. 6 NURBS	10
Fig. 7 Knots, Control Point	10
Fig. 8 Degree	12
Fig. 9 Regular Curve	12
Fig. 10 Irregular Curve	13
Fig. 11 Bisection method	14
Fig. 12 Secant Equation	15
Fig. 13 Secant method	15
Fig. 14 Secant method and Muller's method	16
Fig. 15 Fixed-point interaction	16
Fig. 16 Newton-Raphson	18
Fig. 17 라이노(Rhino)를 이용한 선형모델링	21
Fig. 18 Sheer Plan, Half-breadth Plan, Body Plan	22
Fig. 19 곡선과 임의의 점의 관계 및 함수	23
Fig. 20 Curve profile of the stern	24
Fig. 21 Regular Curve for Newton-Raphson	25
Fig. 22 Irregular Curve for Newton's	26
Fig. 23 최단 거리 계산 가능 구간	27
Fig. 24 최단 거리 계산 불가능 구간	27

Fig. 25 최단 거리 계산 가능 구간	28
Fig. 26 Backtracking Line search	29
Fig. 27 Program Flowchart	30
Fig. 28 Regular Curve explanation	32
Fig. 29 Irregular Curve explanation	32
Fig. 30 [a,b] 범위의 곡선 길이	33
Fig. 31 선형을 이용한 곡선 길이	34
Fig. 32 2구간 등분	35
Fig. 33 4구간 등분	35
Fig. 34 8구간 등분	36
Fig. 35 피타고라스에 따른 선형 길이	36
Fig. 36 최대, 최소 범위 설정	37
Fig. 37 adaptive value K 적용	38
Fig. 38 불균일 곡선에 향상된 알고리즘 적용	39



임의의 점으로부터 3차원 곡선까지 최단거리 계산 알고리즘 향상

An algorithm for the calculation of the shortest distance from an arbitrary point to a 3-dimensional Curve

Hwang, Yoon Suk

Department of Naval Architecture and Ocean Systems Engineering
Graduate School of Korea Maritime and Ocean University

Abstract

현재 선박 및 해양플랜트 구조물에 대한 3차원 전산모델은 NURBS 표현 방법으로 구성되어 있다. 전산모델들을 활용한 후속작업에 빈번하게 사용되는 기능 중 하나가 모델간 거리 계산인데, 이는 주로 공간에 있는 임의의 3차원 점으로부터 한 모델까지 최단거리를 구하는 작업으로 귀결된다. 복잡한 3차원 모델 다룰 때는 투영, 오프셋, 교차와 같은 비선형 작업 과정을 거치게 되는데 그 결과 높은 차수를 가진 비정형적 곡면 및 곡선이 생성된다. 높은 차수의 비정형 모델까지 최단거리를 구하는 것은 단순 모델에서와는 달리 최단거리 계산에 오류가 수반된다. 이는 예측하기 힘든 비선형 특성으로 인한 것으로 일반적인 탐색기법이 불안정적인 과정을 거치기 때문이다. 본 연구에서는 비정형 곡선까지 최단거리를 계산하는 새로운 알고리즘을 개발한다. 개선된 알고리즘을 활용하여 복잡한 비정형 모델을 활용한 연산 작업에서 항상 최단거리를 확보할 수 있는 기술적 기반을 제공한다.

KEY WORDS: NURBS, 모델간 거리 계산, 최단거리 계산, 비정형 곡면 및 곡선

제 1 장 서 론

1.1 연구 배경 및 필요성

현재 선박 및 해양플랜트와 같은 복잡한 형상 구조물은 NURBS(Non-uniform rational B-spline) 기반으로 표현되어 있는데, 형상 구조물 모델링 작업을 수행할 경우 3차원 CAD 시스템을 사용한다.

CAD 시스템을 활용하여 복잡한 형상을 모델링 할 경우에 Intersection, Offset, Projection 과 같은 기능을 수행하게 된다. 이러한 기능을 수행할 경우, 불균일한 곡선(Knots가 규칙적으로 되어있지 않은 곡선)이 생성된다. 불균일한 곡선이 생성되면 CAD 시스템에서 형상 모델링의 후속작업을 수행 할 때 사용되는 최단거리 계산 오류가 발생한다. CAD 시스템에서 최단거리 계산이 필요한 이유는 Fig.1에서 보는 것과 같이 마우스 위치로부터 곡선의 끝점, 중간지점, 수직점, Tangent점 등 사용자가 원하는 위치로 가면 자동으로 인식하는 Object Snap인 OSNAP 기능을 수행하기 위해서 컴퓨터는 끊임없이 최단거리 계산을 한다. 또한 3차원 CAD 프로그램 사용하는데 있어서 Fig.2에서 보는 것과 같이 기하학 모델들 간의 Surface와 Surface를 연결하는 작업을 수행 할 경우에 최단거리 계산이 필요하다. 하지만 앞에서 언급한 것처럼 불균일 곡선이 생성되면 최단거리 계산에 오류가 수반된다. 이러한 문제를 발생하지 않기 위해서 기존 상용 CAD 프로그램에서는 불균일 곡선이 발생하면 균일한 곡선(Knot가 균일하게 분포되어 있는 곡선)으로 정제해주는 Refit기능이 있다. 이러한 과정을 통해 최단거리를 계산할 경우, 균일한 곡선을 이용하여 최단거리를 계산한다. 그러나 Refit 기능을 수행하면 정제된 곡선인 균일한 곡선이 생성되므로 사용자가 얻고자하는 정확한 곡선을 생성하지 못하며 불균일한 곡선이 생성될 경우, 임의의 점으로부터 최단거리 계산하는데 한계가 있다. 이러한 문제를 개선하기 위해 기존의 균일한 곡선에 적용한 최단거리 계산 알고

리즘을 불균일한 곡선까지 확장하여 최단거리를 구할 수 있도록 향상시켜 적용한다. 구현 된 알고리즘은 사용자가 쉽게 가시화 할 수 있도록 사용자인터페이스(MFC)와 범용 그래픽라이브러리(OpenGL)를 이용하여 GUI 프로그램을 구축 하도록 한다.



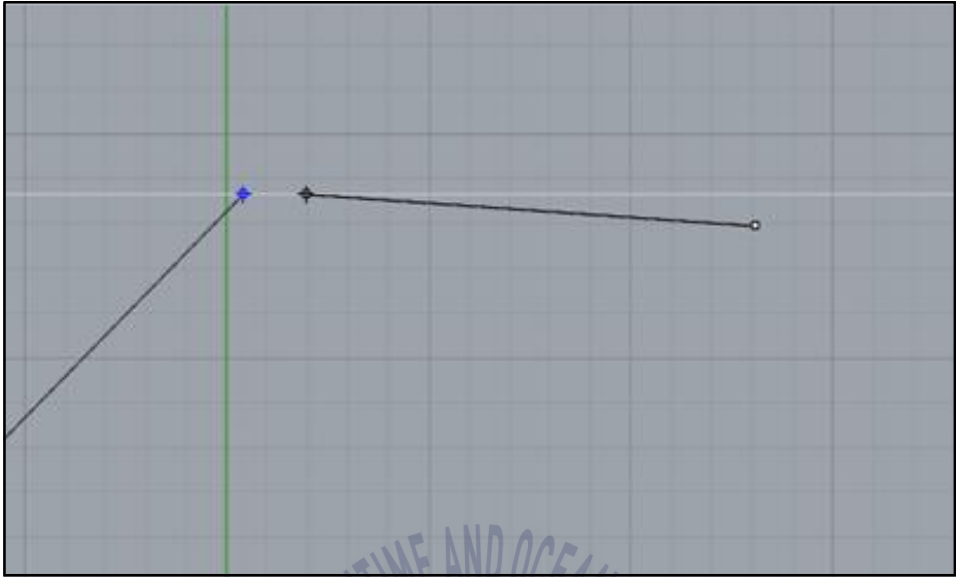


Fig.1 OSNAP 기능

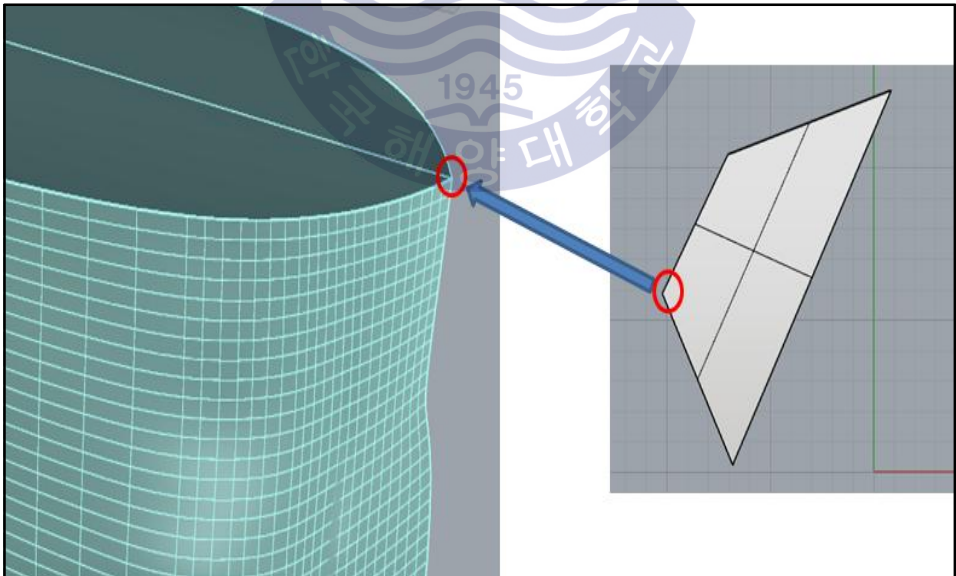


Fig.2 Connecting Surface and Surface

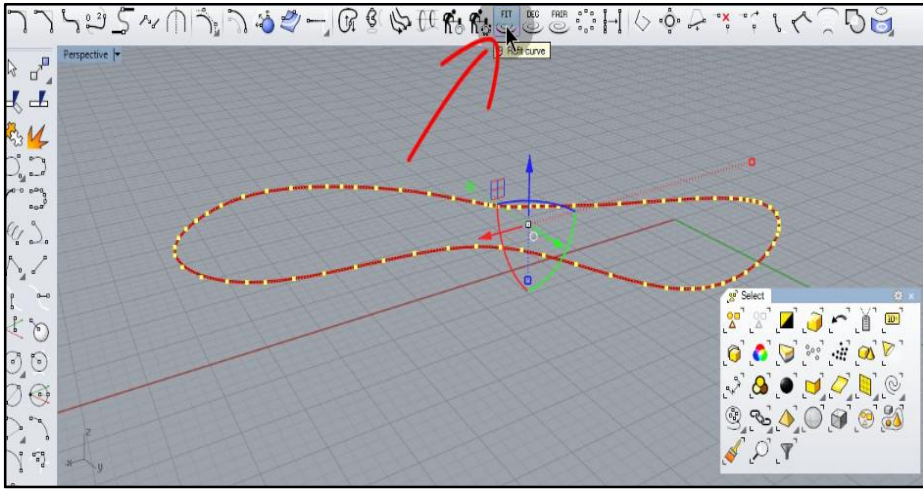


Fig.3 Refit 기능 수행 전 곡선

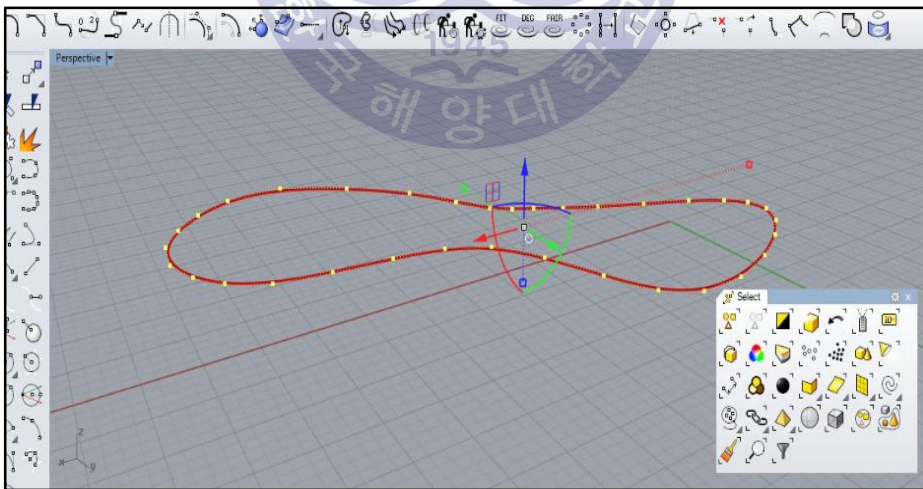
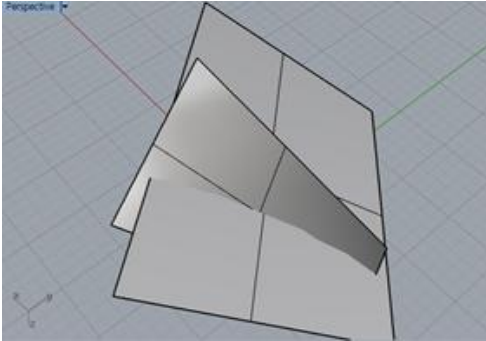
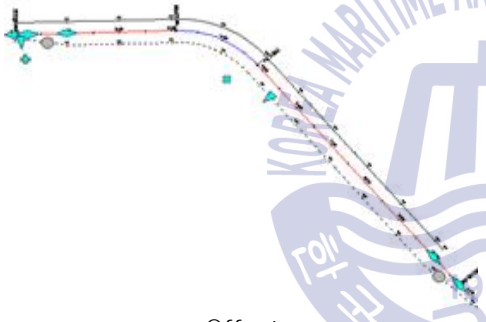


Fig.4 Refit 기능 수행 후 곡선

Table1. Intersection, Offset, Projection

	설 명
 <p data-bbox="333 730 546 759">< Intersection ></p>	<p data-bbox="732 537 1126 614">복잡한 형상을 모델링 할 때 Surface 교차 경우 사용한다.</p>
 <p data-bbox="371 1110 509 1139">< Offset ></p>	<p data-bbox="714 929 1144 1006">보강재 두께 모델링 및 외판두께 모델링 할 때를 사용한다.</p>
 <p data-bbox="344 1508 533 1537">< Projection ></p>	<p data-bbox="732 1319 1126 1396">3차원 모델을 2차원의 모델로 표현할 때 사용한다.</p>

1.2 연구 내용

불균일한 곡선을 생성하기 위해서 선체 모델링 중 복잡한 형상을 가진 곡선 중 일부를 추출하여 활용하였으며, 최단거리 계산에 이용되는 Newton-Raphson법에 적용하여 균일한 곡선에서의 최단거리 계산과 불균일 곡선에서의 최단거리 계산을 비교하였으며, 불균일 곡선에서 Newton-Raphson 기법이 적용되지 않는 원인에 대해서 파악하고, 해결방안을 제시하여 불균일 곡선에 기존 Newton-Raphson 기법 알고리즘이 적용되지 않는 부분을 수정하고, 새로운 알고리즘을 추가하여 기존의 Newton-Raphson 기법을 향상시킴으로서 불균일한 곡선에서도 최단거리 계산이 가능하도록 하였다.



제 2 장 주요 기술에 대한 기본 이론

2.1 개요

본 연구에서 사용된 주요 이론에 대한 정리이다. 곡선 및 곡면 표현기술인 NURBS와 기본적인 곡선, 최단 거리 계산 기법 중 하나인 Newton_Raphson 기법, 데이터 구조, 범용그래픽라이브러리(OpenGL)가 연구에 활용되었으며 이에 대한 기본 이론과 설명을 기술 한다. 2장에서 주요 기술들을 정리함으로 이후 자세한 설명은 하지 않도록 한다. NURBS와 Newton-Raphson은 주요 연구에 모두 활용 되었다.

2.2 OpenGL 그래픽스 라이브러리

OpenGL(Open Graphics Library)의 약자로, 시초는 1982년 실리콘 그래픽스(SGI)가 유닉스용으로 개발한 그래픽 라이브러리인 IRIS GL이다. 최초 2D용으로 개발되었다가 자사의 그래픽 워크스테이션을 위해 3D로 업그레이드되었다. 특정 기계를 위해 만든 라이브러리다보니 이식성이 부족했고, 특허 문제로 인해 범용적으로 사용하기 어려웠다. 그래서 여러 회사들과 OpenGL ARB(Architecture Review Board)를 결성하여 표준화 작업을 진행하였다. IRIS GL의 범용성과 이식성을 개선하여 1992년 개방형 라이브러리를 다시 만든 것이 OpenGL이다. 특징은 개방되어 있고 무료이며

이식성이 뛰어나다. 앞에서 언급했던 것처럼 개방된 표준이므로 플랫폼에 독립적이다. 운영체제와 연관된 부분이 없고 디바이스 드라이버를 통해 하드웨어를 직접 제어하므로 드라이버만 존재하면 모든 하드웨어 환경에서 실행할 수 있다. 대부분의 데스크탑 환경, 모바일 환경은 물론이고 PlayStation, Nintendo 등의 게임기 등에도 광범위하게 이식되어 있다. 그리고 간단한 함수들로 구성되어 있으므로 개발 언어에도 독립적이다. C/C++, C#, JAVA, Python 등 대부분의 언어에서 사용 가능하며, 문서와 예제가 풍부하고 재사용한 코드를 쉽게 구할 수 있다는 것이 큰 이점이다. 또 그래픽 프로그램이 요구하는 속도를 충분히 만족한다. 그리고 하드웨어의 가속 기능을 최대한 활용하여 효율적으로 동작한다.

2.3 GUI

GUI는 Graphic User Interface의 약자로 텍스트 위주의 화면이 아닌 그래픽을 통해 사용자와 컴퓨터간의 인터페이스를 구현하는 것을 뜻한다. 처음 컴퓨터와의 사용자 인터페이스가 그래픽이 아닌 텍스트와 키보드 중심이며, 많은 명령어들을 외워야하는 불편함 때문에 생겨났다. 검정색 바탕의 화면에 텍스트만 나왔던 DOS의 명령어 인터페이스가 GUI가 나타나기 이전의 대표적인 사용자 인터페이스의 예이다. 키보드로 명령어를 입력하는 대신 마우스를 이용하여 동작이 가능한 메뉴 기반의 인터페이스는 DOS와 같은 명령 중심의 인터페이스와 GUI사이의 중간단계의 인터페이스라고 할 수 있다.

오늘날 대부분의 운영체제는 그래픽 사용자 인터페이스를 지원한다. 응용프로그램들도 대부분 운영체제와 함께 따라오거나 또는 자신들만의 GUI요소와 아이디어를 추가하여 사용한다. 우리가 흔히 아이콘이라고 부르는 것이 GUI의 예이다.

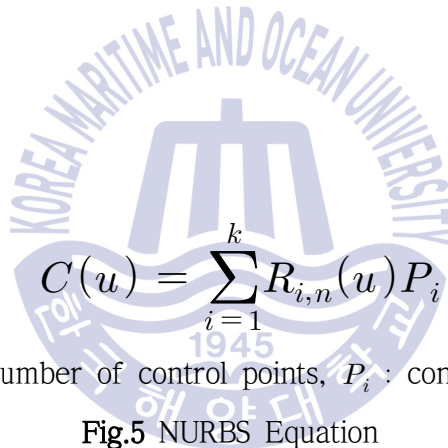
GUI는 실제 생활에서 잘 알려진 하나 또는 그 이상의 객체들을 비

유적으로 사용하는데, 그 예로는 바탕화면이나 윈도우 창을 통해 보는 것 등이 있다. GUI의 요소들은 윈도우, 풀다운 메뉴, 단추들, 스크롤바, 아이콘 이미지, 위저드, 마우스 등을 포함 한다.

GUI의 일부로서 멀티미디어의 사용이 증가함에 따라, 많은 응용프로그램에서 소리, 음성, 동영상 및 가상실 인터페이스 등도 GUI의 일부가 될 것으로 보인다. 어떠한 시스템의 그래픽 사용자 인터페이스는 그것의 입력장치들과 함께 그것의 look-and-feel' 이라고 불린다.

2.4 NURBS

2.4.1 NURBS



$$C(u) = \sum_{i=1}^k R_{i,n}(u) P_i$$

(k : the number of control points, P_i : control points)

Fig.5 NURBS Equation

NURBS는 Non-Uniform Rational B-spline의 약자로서 정형화되지 않은 함수의 곡선이라는 뜻(Piegl L, Tiller W(1997))으로, 3차원 기하체를 수학적으로 재현하는 방식 중 하나로서 2차원의 간단한 선분 원, 호, 곡선부터 매우 복잡한 3차원의 곡선이나 곡면까지 정확하게 표현할 수 있으며 편집이 쉽다는 특징이 있다. 이러한 특징을 가지고 있기 때문에 상용 CAD 프로그램(3D MAX, Auto cad, Rhino, NX 등)에서 사용된다.

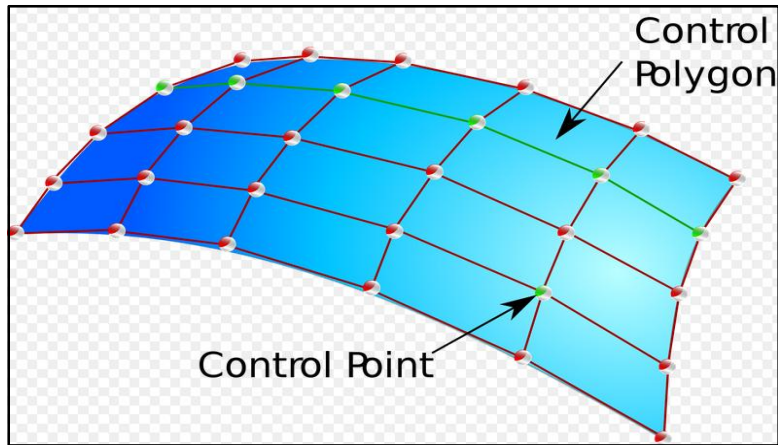


Fig.6 NURBS

2.4.2 Control Point

NURBS 곡선은 곡선의 차수에 1을 더한 수 이상의(Degree+1) Control Point를 가진다. NURBS 곡선의 형태를 바꿀 수 있는 가장 쉬운 방법이 바로 C.P(Control Point)를 움직이는 것이다. C.P는 무게(Weight)와 관련이 있는데 무게는 대부분 양수를 나타낸다. 한 곡선의 모든 C.P들이 같은 무게를 가지고 있을 때 일반적으로 1인 곡선을 비유리(Non-rational)이라고 한다. 실생활에서 대부분의 경우 NURBS 곡선은 유리하지 않다.

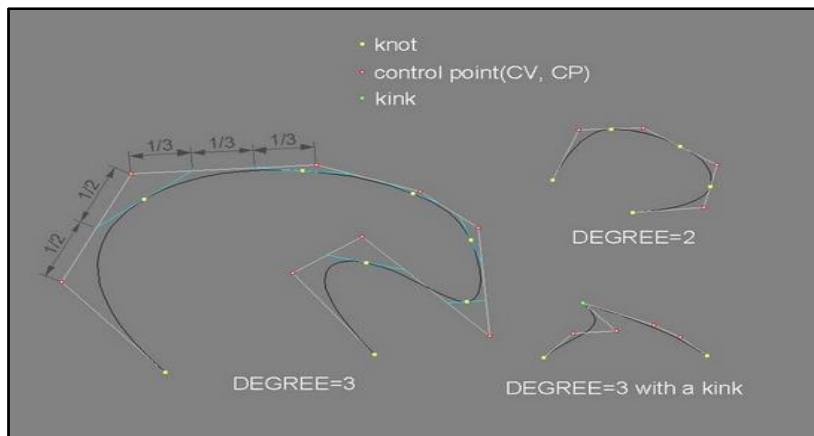


Fig.7 Knots, Control Point

2.4.3 Knots

Knot(매듭)이란 곡선의 차수에 C.P의 개수를 더한 뒤 다시 1을 뺀 수들의 목록을 의미한다. 이러한 수의 집합을 Knot vector(매듭 벡터)라고 하며, 여기서의 벡터는 3d에서 사용되는 방향을 의미하지 않는다. 이러한 수의 목록이 ‘매듭’이 되기 위해서는 여러 가지 조건을 만족시켜야 한다. 먼저 목록 내의 수가 뒤에 있을수록 수의 값이 그대로 이거나 증가해야 한다. 또한 같은 값이 반복될 경우 그 횟수는 곡선의 차수보다 클 수 없다. 예를 들어 3차에 11개의 컨트롤 포인트를 가진 NURBS 곡선의 경우, 0,0,0,1,2,2,2,3,7,7,9,9,9을 그 매듭으로 가질 수 있다. 하지만 0,0,0,1,2,2,2,2,7,7,9,9,9는 성립하지 않는데 그 이유는 2가 네 개이기 때문이다. 매듭 내에서 특정수가 반복될 경우 그 횟수를 ‘매듭 반복횟수(knot's multiplicity)’라고 한다.

위의 예를 다시 보면 매듭 값 0은 반복횟수가 3이며, 1은 1번, 2는 3번, 7은 2, 9는 3이다. 반복횟수가 차수와 같을 경우 이 매듭을 ‘최대반복매듭(full-multiplicity knot)’이라고 한다. 즉 이 예시에서 최대반복매듭은 0, 2, 9이다. 매듭 값이 한 번만 나타나는 경우에는 이를 단순 매듭(simple knot)이라고 한다. 위 예시의 경우 1과 3이 바로 ‘단순 매듭’이다. 매듭 내에서 특정수가 반복될 경우 그 횟수를 ‘매듭 반복횟수(knot's multiplicity)’라고 한다. 이러한 매듭 값이 시작과 끝이 아닌 중간에서 반복 된다면, 이것은 해당 NURBS 곡선이 덜 부드럽다는 것을 의미한다. 극단적으로 어떤 NURBS곡선에서 ‘최대반복매듭’이 매듭 목록의 중간에 있다면 이 곡선은 이 부분에서 더욱 급하게(kink) 구부러지는 것을 의미한다. 이러한 특성을 이해한다면 컨트롤 포인트를 움직이는 것뿐만 아니라 매듭을 더하거나 제거하면서 곡면을 디자인 하는 것이 가능하다. 매듭을 NURBS 곡선에 더해도 그 형상은 바뀌지 않는다. 하지만 Knot을 제

거하게 되면 일반적으로 곡선의 형상은 변하게 된다.

2.4.4 Degree

Degree는 NURBS 다항식의 차수이다. 수학에서 사용되는 2차 함수, 3차 함수 등을 얘기할 때 쓰는 방정식의 차수를 뜻한다. 즉, Degree = 1이라는 것은 1차 함수의 곡선을 말하는 것이므로 곡선은 직선이 된다. Degree = 3인 경우, Control Point를 하나 움직였을 때 곡선이 변화하는 Control Point가 3번째까지라는 것이다.

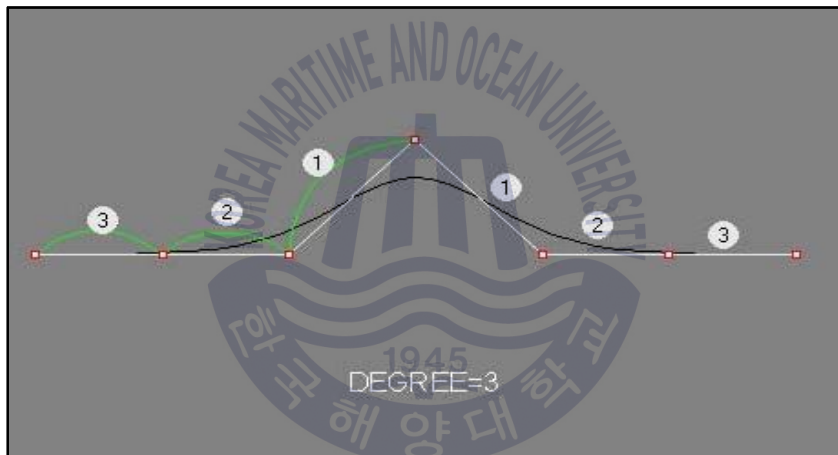


Fig.8 Degree

2.4.5 균일 곡선과 불균일 곡선 비교

- 균일 곡선

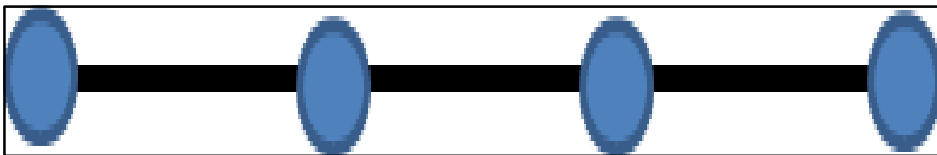


Fig.9 Regular Curve

균일 곡선이란, Knots가 일정하게 분포가 되어있는 곡선을 말한다. Knots Vector가 균등하기 때문에 parameter가 균등하게 되어있다. 이러한 곡선을 균일 곡선이라고 한다. 곡선을 속도에 비유하면, 곡선의 속도가 일정한 속도를 가지는 곡선을 말하며, Fig.6에서와 같이 구간별로 속도가 같은 것을 알 수 있다. 일반적으로 사용 CAD 프로그램에 사용되는 곡선이다.

- 비균일 곡선



Fig.10 Irregular Curve

비균일 곡선이란, Knots가 일정하지 않게 분포가 되어있는 곡선을 말한다. 즉, Knot Vector가 비균등하기 때문에 parameter가 균등하지 않게 되어있다. 그래서 곡선의 속도가 일정하지 않으며, Fig.7에서처럼 어떤 구간에서는 곡선의 속도가 빠른 속도로 가다가 어떤 구간에서는 곡선의 속도가 천천히 가는 곡선을 말한다. 그래서 곡선을 등분하면 일정하게 등분되지 않는다.

2.5 최단거리 기법 비교

최단거리 계산에 필요한 방정식의 근을 구하는 기법에는 수치해석을 이용한 이분법, Muller법, Secant법, Newton's 법 등의 기법이 있다.

2.5.1 이분법(Bisection method)

이분법은 방정식의 근을 구하는 가장 기초적인 방법으로서 구간 $[a,b]$ 의 양 끝점 사이에 근이 존재한다고 가정을 하는 구간법의 종류이다. $f(a)$ 와 $f(b)$ 의 부호는 반대이고, 두 수 사이에 근을 반드시 포함하는 두 개의 초기값이 필요하며, 수렴성이 좋다는 장점이 있다. 하지만 수렴속도가 느리고, 중근을 갖는 경우 부호변화가 나타나지 않을 수 있는 단점이 있다.

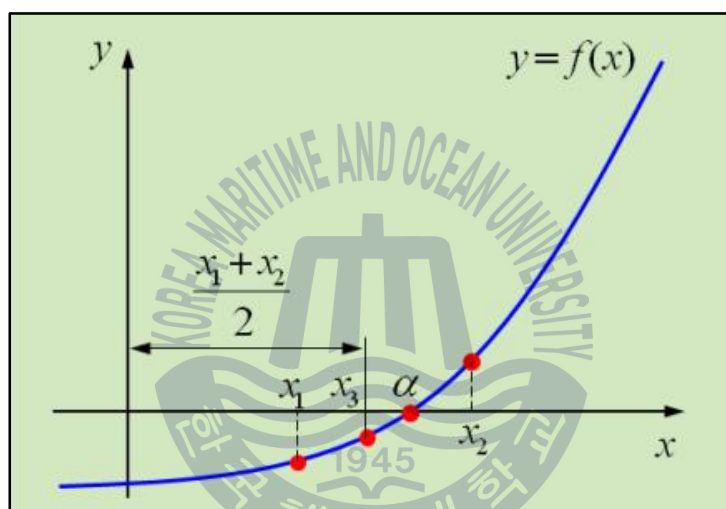


Fig.11 Bisection method

2.5.2 할선법(Secant method)

할선법은 $f'(x)$ 의 근사값을 이용하여 x 절편을 구하는 방법이다. 할선법으로 방정식 $f(x) = 0$ 의 해를 구하기 위해서, 해 근방의 두 점 x_0, x_1 (초기값)을 잡는다. 두 점 $(x_0, f(x_0)), (x_1, f(x_1))$ 을 지나는 직선을 구하여 이 직선이 x 축과 만나는 점의 좌표를 x_2 로 잡는다. 다음에 x_0 를 버리고 $(x_1, f(x_1)), (x_2, f(x_2))$ 를 잇는 할선의 x 절편을 찾는 것을 반복한다.

일반적으로, 두 점 $(x_n, f(x_n)), (x_{n-1}, f(x_{n-1}))$ 을 지나는 직선의 방정식은

$$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Fig.12 Secant equation

이므로 x축과 만나는 점은 y=0으로 놓고 x에 대하여 풀면 된다.

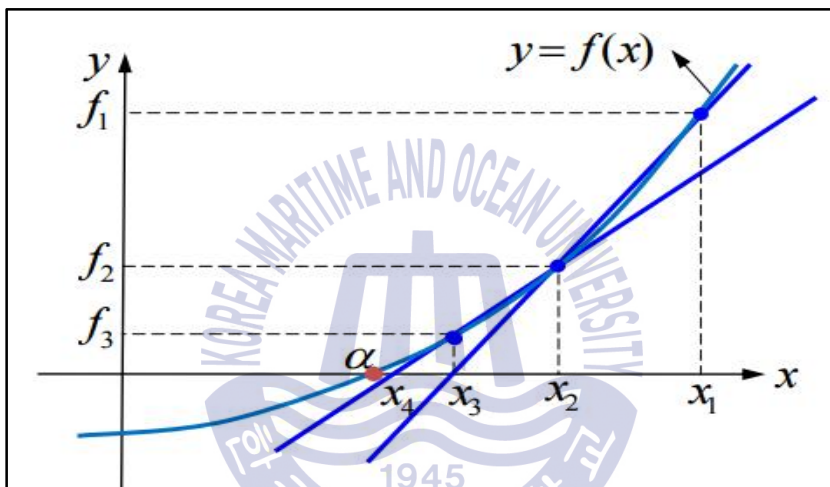


Fig.13 Secant method

2.5.3 Muller법(Muller' s method)

Muller법은 할선법의 연장으로서 인접해 있는 세 개의 점을 지나는 포물선을 구하고, 이 포물선이 x축과 만나는 점을 다음 점으로 하여 계속 반복하여 방정식의 근을 찾는 방법이다.

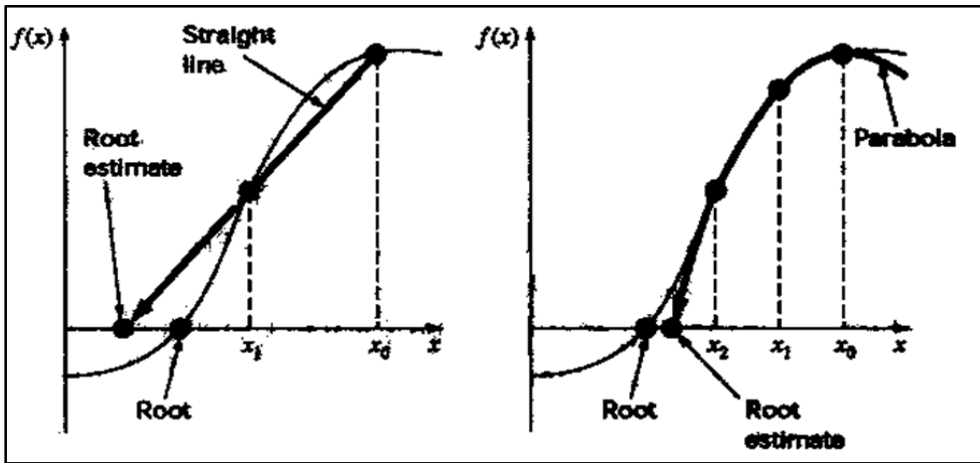


Fig.14 Secant method and Muller' s method

2.5.4 고정점 반복법(Fixed-point iteration)

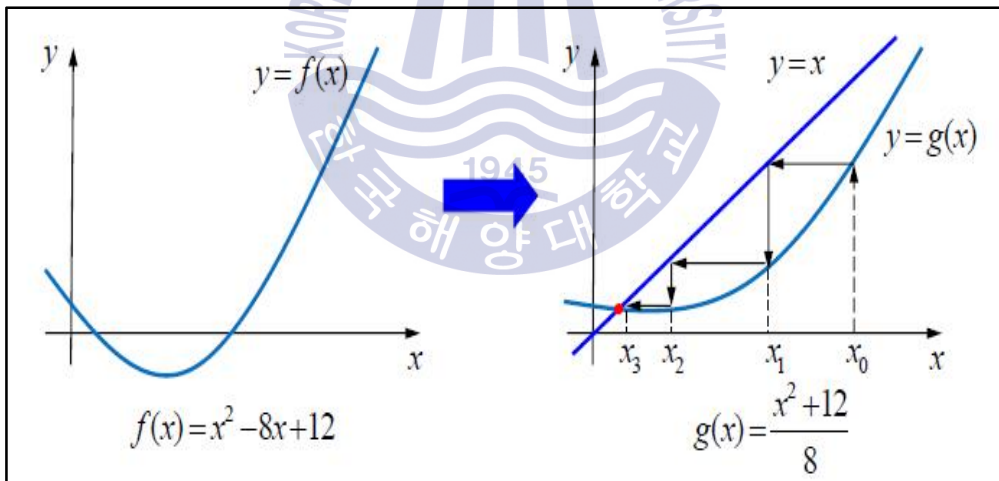


Fig.15 Fixed-point interaction

고정점 반복법은 $x = g(x)$ 의 형태로 나타내는 방정식의 해를 구하는 방법이다. 만약 α 가 방정식 $g(\alpha) = \alpha$ 의 해가 되면, 여기서 α 를 고정점이라고 한다. 즉, $\alpha = g(\alpha) = g(g(\alpha)) = g(g(g(\alpha))) = \dots$ 임을 의미한다. 예를 들

어, 임의의 상수 c 에 대하여 $g(x) = x - cf(x)$ 라 하면, $f(x) = 0$ 의 근이 고정점이 된다. 즉 $f(\beta) = 0$ 이면 $\beta = \beta - cf(\beta)$ 이다. 정리하면, 함수 $g(x)$ 가 구간 $[a, b]$ 에서 연속이고 모든 $x \in [a, b]$ 에 대하여 $a \leq g(x) \leq b$ 이면 $g(x)$ 는 적어도 하나의 고정점을 갖고, $|g'(x)| \leq L < 1$ 이면 유일한 고정점을 갖는다. 또한, 점화관계 $x_{n+1} = g(x_n)$ 를 만족하는 수열 $(x_n)_n^\infty$ 은 g 의 고정점 α 에 수렴한다.

2.6 Newton' s 기법

2.6.1 Newton' s 기법

방정식 $f(x) = 0$ 의 해를 근사적으로 찾을 때 유용하게 사용되는 방법이다.

- (i) 근의 적당한 위치를 추정하여 이것을 초기값 x_0 라고 한다.
- (ii) $x = x_0$ 에서 $y = f(x)$ 에 대한 접선은 $y_1 = f(x_0) + f'(x_0)(x_1 - x_0)$ 이므로 이 접선의 x 절편 x_1 은 방정식 $0 = f(x_0) + f'(x_0)(x_1 - x_0)$ 을 만족하고, 이 식을 x_1 에 대해서 풀면,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (1)$$

를 얻는다.

- (iii) x_1 을 다시 초기값으로 하여 이 과정을 반복하면,

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad (2)$$

을 얻을 수 있다.

(iv) 이 과정을 표준화시키면,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3)$$

을 얻을 수 있다. 이 과정을 뉴턴-랩슨 방법(Newton-Raphson method) 또는 뉴턴의 방법(Newton's method)이라 한다.

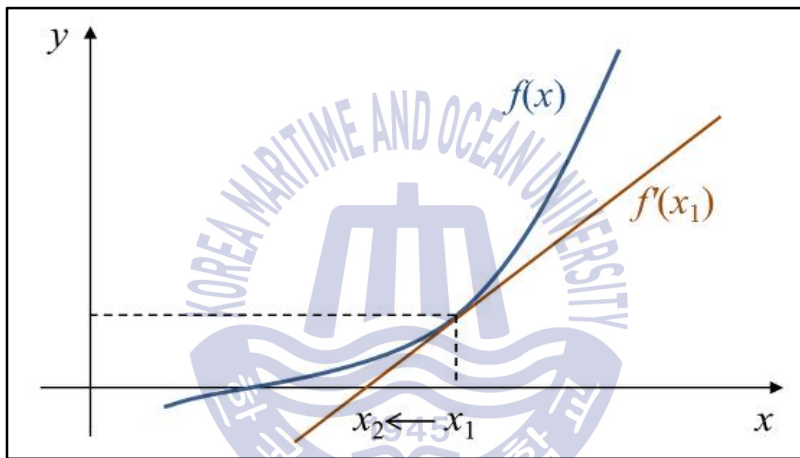


Fig.16 Newton-Raphson

Fig.12에서 만일 처음에 $x = x_1$ 에서 시작했다면 그 다음 x 값은 x_2 가 될 것이고, x_2 에서 다시 접선을 그려보면 점차 실제 해에 가까워지는 것을 확인할 수 있다. 이와 같은 과정을 계속 반복하다보면 해를 찾을 수 있다. 이 내용을 수식화하면 아무 값이나 초기값 x_1 에서 시작하여 다음 수식에 따라 수렴할 때까지 계속 x 를 이동시켜 나가는 것이다.

※ 뉴턴의 방법은 모든 함수에 적용할 수 있는 것은 아니다. 뉴턴의 방법으로부터 얻어지는 값들은 근에 점점 가까워져야 하며 원하는 정확도에 도달할 때까지 계산을 해야 한다. 즉, 구하는 근사 값에서 함수 값을 계산

해서 그 값이 0으로 접근하지 않는다면 그 값을 근사 근으로 택할 수 없다.

방정식을 예로 들면 $f(x) = x^5 - x + 1$ 함수의 근을 구할 경우,
 $f(-2) = -29$, $f(-1) = 1$ 이므로 $f(-2)f(-1) < 0$ 이다. 중간값 정리에 의하여 $[-2, -1]$ 에서 근을 갖는다.

$x_0 = -1$ 이라고 하고 뉴턴의 방법을 적용해보면,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -1 - \frac{(-1)^5 - (-1) + 1}{5(-1)^4 - (-1)} = -\frac{5}{4}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = \frac{5}{4} - \frac{\left(-\frac{5}{4}\right)^5 - \left(-\frac{5}{4}\right) + 1}{5\left(-\frac{5}{4}\right)^4 - \left(-\frac{5}{4}\right)} = -1.17846$$

$$x_3 \approx -1.167537$$

$$x_4 \approx -1.167304$$

$$x_5 \approx -1.167303978 \approx x_6$$

x_5 부터는 숫자가 변하지 않는 것을 알 수 있다.

따라서 $f(x) = x^5 - x + 1$ 의 근의 근사값은 -1.167303978 이다.

$f(-1.167303978) = 3 \times 10^{-12}$ 으로 거의 0임을 알 수 있다.

종료 조건은 x값의 변화가 거의 없을 때까지이다.

Newton-Raphson(Newton's)법의 특징은 상당히 단순함에도 불구하고 다항식의 해에 '매우 빠르게' 근접할 수 있는 특징이 있으며, 대부분의 경우 4~5회 정도 시행하게 된다면 10^{-4} 정도의 오차 내로 해를 구할 수 있게 된다. 본 연구에서는 이러한 특징을 가지고 있는 Newton-Raphson 기법을 이용한다.

2.6.2 최단거리 기법 비교 및 Newton-Raphson 사용 배경

Table.2 Muller, Secant, Newton, Bisection

	Muller 법	Secant 법	Newton 법	이분법	고정점 반복법
반복횟수	2	7	4	11	13
백분율 상대오차	0.014796%	0.0016%	0.0098%	0.0480%	0.04075%
CPU TIME	0.0006333	0.0003654	0.0005118	0.0004467	0.0005135

Table.2에 정리한 것처럼 Muller법, Secant법, Newton법, 이분법을 비교해 보았을 때 반복횟수, 백분율 상대오차, CPU TIME 값을 전체적으로 비교해 보았을 때 Newton법이 가장 효과적인 기법이라는 것을 확인 할 수 있으며, 그 결과 본 연구에서는 Newton-Raphson 기법을 응용한다.

제 3 장 Newton-Raphson 법을 활용한 최단거리 검색

3.1 선형 모델링의 NURBS 곡선

3.1.1 라이노(Rhino) 선형 모델링

선박에서 Lines 작업을 할 때 구간을 생성하여 Intersection과 같은 작업을 통해서 Curve profile을 생성한다. 이러한 작업을 통하면 불균일한 곡선이 생성되는데, 본 연구에서는 라이노(Rhino)를 이용하여 선형 모델링된 것을 활용하여 비교적 복잡한 형상으로 생성되어 있는 선미부분의 곡선 profile을 균일한 곡선과 불균일한 곡선을 이용하여 Newton_Raphson기법에 적용하여 확인하였다.

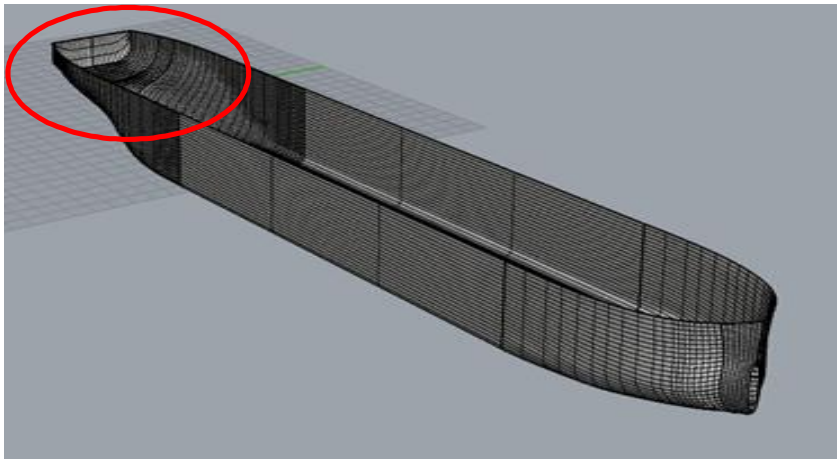


Fig.17 라이노(Rhino)를 이용한 선형모델링

3.1.2 선도(Lines)

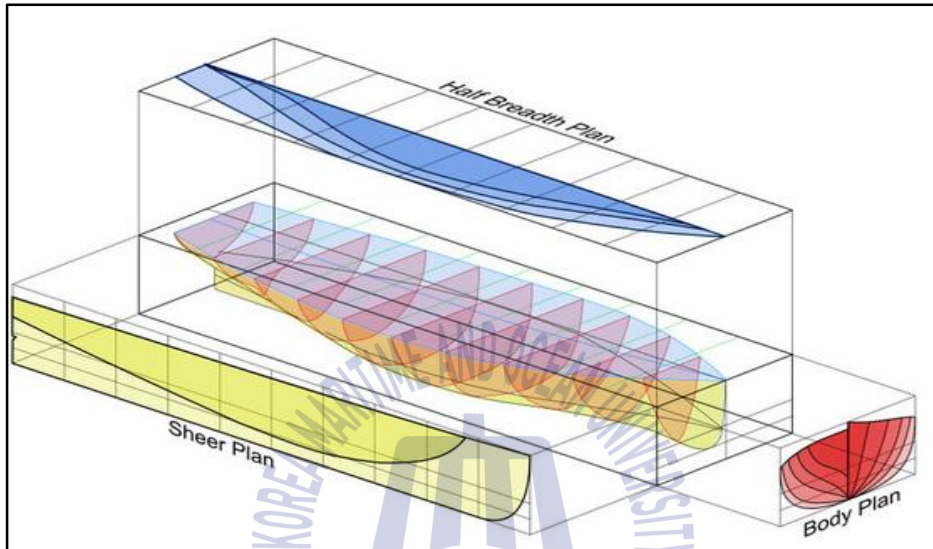


Fig.18 Sheer Plan, Half-breadth plan, Body plan

선박은 ‘선도’라 불리는 축소된 도면위에 도학의 방식에 따라 표시된다. 선도는 실질적으로 3개의 투상면(Plans or views), 즉 측면도(Sheer Plan), 반폭도(Half-breadth Plan) 및 정면도(Body plan)로 이루어져 있다. 측면도는 선박의 연직 종단면들의 모양을 나타내고, 반폭도는 선박의 수평 단면들의 모양을 나타낸다. 그리고 정면도는 연직 횡단면들을 나타낸다. 모든 선박은 세로 중심면(Longitudinal Center Plane)에 관하여 대칭이므로, 선박의 한쪽 모양만을 나타내는 것이 보통이다.

3.2 곡선과 임의의 점과의 관계 분석

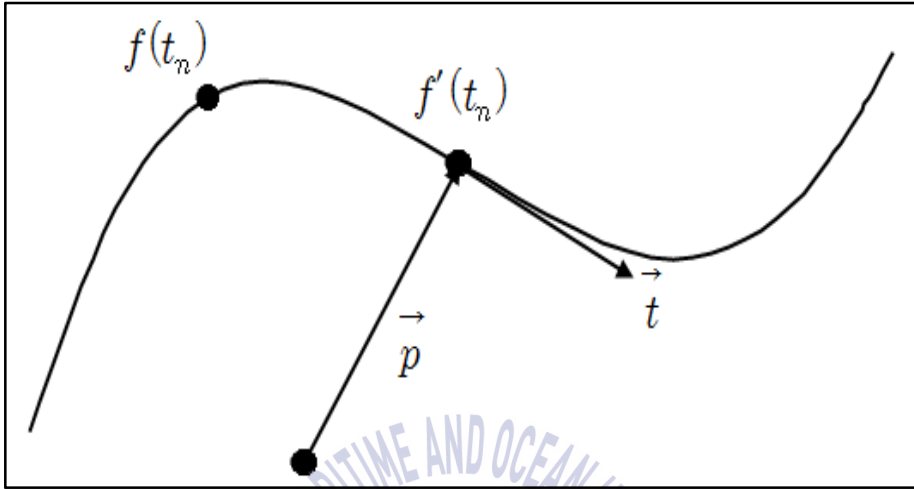


Fig.19 곡선과 임의의 점과의 관계 및 함수

$$t_{n+1} = t_n - \frac{f(t_n)}{f'(t_n)} \quad (1)$$

$$\cdot f(t_n)$$

Newton's 법을 활용하여 최단거리 탐색할 때 $\vec{c}(t)$ 함수를 곡선위의 함수라고 하고 곡선 밖의 임의의 점을 p 라고 한다. 그러면 곡선위의 점과 임의의 점사이의 \vec{p} 식이 나온다. 이 관계를 통해서 $f(t) = \vec{p} \cdot \vec{t}$ 관계식을 알 수 있다. $\vec{p} = (\vec{p}_0 - \vec{c}(t_q)) \cdot \vec{c}'(t_q)$ 다음과 같이 나타낼 수 있으며, 최종 $f(t_n)$ 는 $f(t) = (\vec{p}_0 - \vec{c}(t_q)) \cdot \vec{c}'(t_q)$ 이다. $f(t_n)$ 함수가 중요한 이유는 두 개의 함수의 내적이 0이 되는 지점이 곡선과 임의의

점과의 관계가 수직거리 및 최단거리 이다.

- $f'(t_n)$

$f'(t_n)$ 는 곡선위의 점에서 tangent인 함수이다. 즉, 곡선위의 점 함수인 $f(t_n)$ 에서 수직인 함수이며 아래와 같이 나타낼 수 있다.

$$f'(t) = (-\vec{c}'(t_q)) \cdot \vec{c}'(t_q) + (\vec{p}_0 - \vec{c}(t_q)) \cdot \vec{c}''(t_q)$$

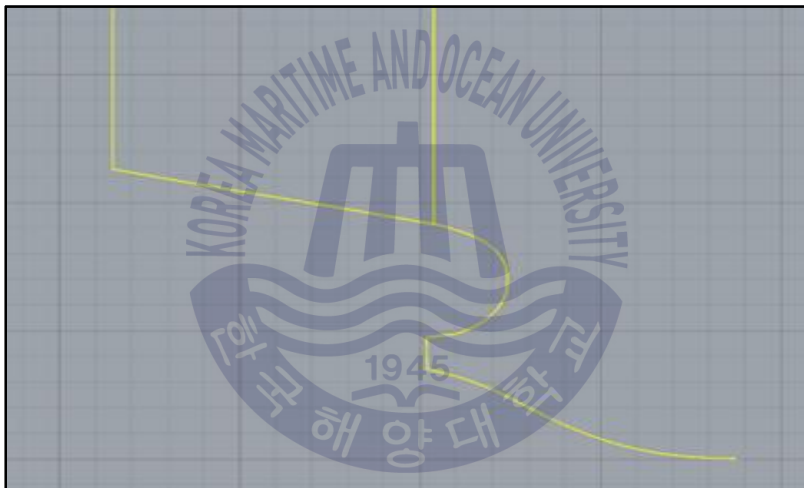


Fig.20 Curve profile of The stern

Fig.16은 선미 쪽의 곡선 profile을 나타낸 것이다.

3.3 NURBS곡선에 Newton-Raphson 기법 적용

3.3.1 균일 곡선과 비균일 곡선에 Newton-Raphson 적용

라이노를 이용하여 얻은 곡선 profile을 균일한 곡선과 비균일한 곡선에

각각 Newton-Raphson기법을 적용하였다.

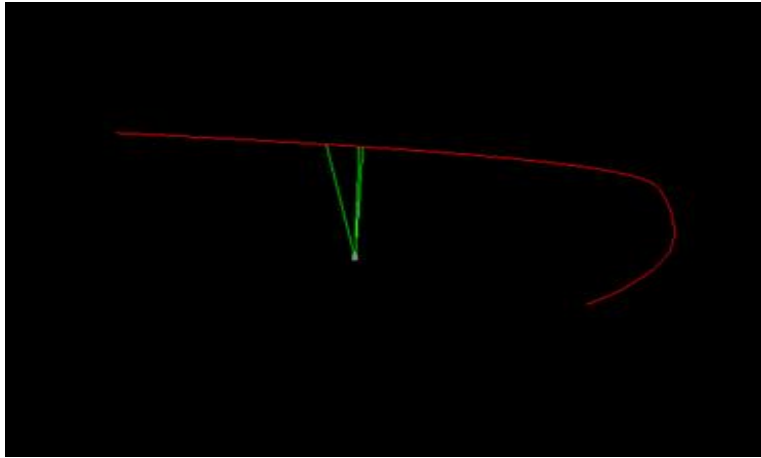


Fig.21 Regular Curve for Newton' s

Table.3 균일 곡선에 대한 Newton' s 결과표

X값	수렴값
X1	0.047
X2	0.049
X3	0.054
X4	0.057

Fig.17은 균일한 곡선에 Newton-Raphson 기법을 적용한 결과이다. 왼쪽의 결과표에서 보는 것처럼 X값이 증가함에 따라, 즉 시간이 지남에 따라 한 곳에 수렴하는 것을 확인할 수 있다. 이것으로 균일한 곡선에 Newton-Raphson 기법이 적용되는 것을 알 수 있으며 그림을 통하여 최

단거리를 찾는 것을 확인 할 수 있다.



Fig.22 Irregular Curve for Newton' s

Table.4 불균일 곡선에 대한 Newton' s 결과표

X값	수렴값
X1	0.047
X2	0.596
X3	23.77

Fig.18은 비균일 곡선에 Newton-Raphson기법을 적용한 결과이다. 오른쪽 표의 결과표에서처럼 X값이 증가할수록, 즉 시간이 지남에 따라 값이 발산하는 것을 확인할 수 있으며, 왼쪽 그림은 값이 최단거리에 수렴하는 것이 아닌 다른 곳에 수렴하는 것을 확인할 수 있다.

3.3.2 구간별 NURBS곡선에 Newton-Rapshon 적용

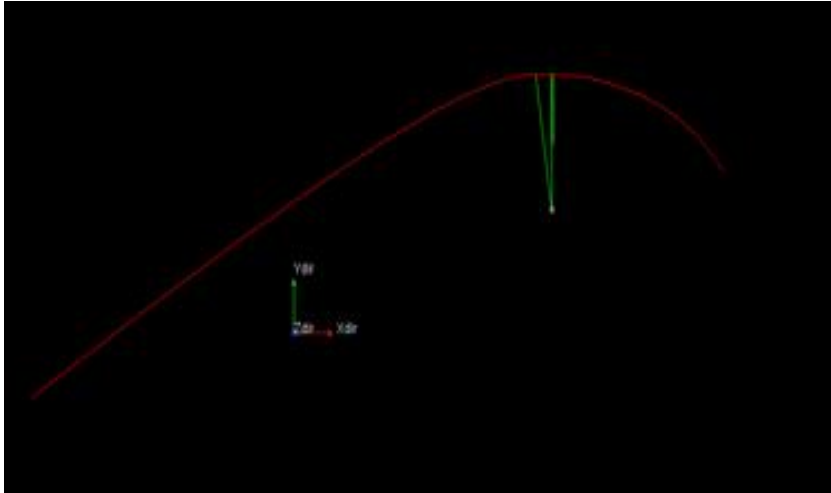


Fig.23 최단거리 계산 가능 구간

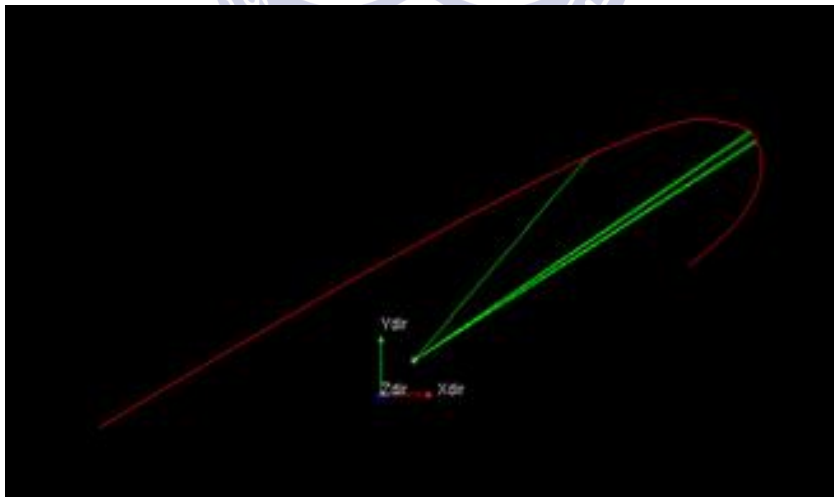


Fig.24 최단거리 계산 불가능 구간

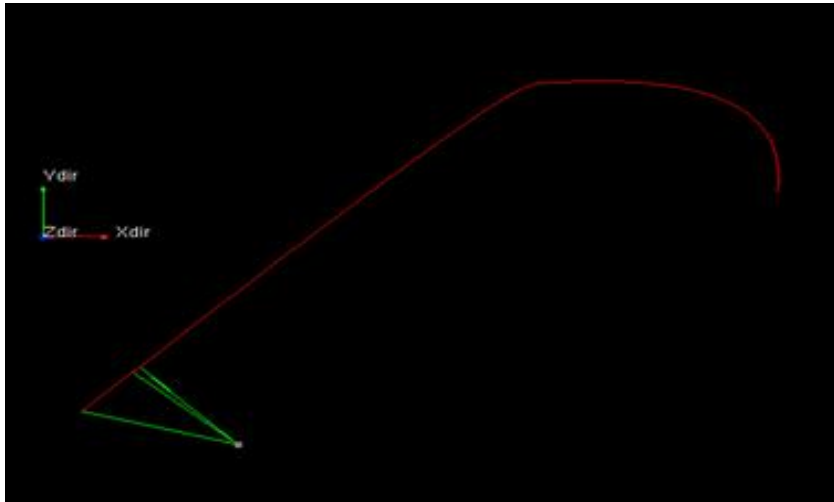


Fig.25 최단거리 계산 가능 구간

Fig.에서 보는 것과 같이 곡선의 오른쪽 구간과 왼쪽 구간에서는 초기점을 시작으로 한 점에 수렴하여 최단거리 계산이 가능하다는 것을 알 수 있다. 하지만 Fig.에서 보면 곡선의 중앙 지점에서 최단 거리 계산에 오류가 수반되는 것을 확인할 수 있다. 즉, 불균일한 곡선에서 전 구간에 Newton-Raphson을 적용하였을 때, 최단 거리 계산이 불가능하다는 것을 확인할 수 있다.

3.4 Line Searches and Backtracking 알고리즘

최적화 기법에서 스텝(Step)의 크기를 결정하는 방법(NUMERICAL RECIPES in C)에 관한 것이다. 스텝의 크기가 너무 크면 발산의 위험이 있고, 또한 극점 근처에서는 수렴속도가 느려지는 문제점이 있습니다. 그리고 이차미분을 이용한 경우에도 변곡점 근처에서는 탐색이 불안정해지는 문제점이 있다. 그런데 이러한 문제점의 원인은 스텝의 크기를 현재 지점에서의 국소적인 변화만을 보고 정하는데 있다. Line search 방법은 이동하고자 하는 방향을 따라서 실제 함수 값의 변화를 미리 살펴본 후에

이동할 양을 결정하는 방식이다.

Line search 방법에는 여러 가지 방법이 있는데 그 중에서 Backtracking 방법은 이동하고자 하는 방향을 따라서 최대한 멀리 간 다음에, 해당 지점의 함수 값이 현재의 함수 값에 비해서 충분히 작아졌는지 검사한다. 만일 충분히 작지 않다면 점차적으로 거리를 줄여가면서 다시 함수 값을 비교한다. 그래서 충분히 작아졌다면 해당 지점으로 이동한다. 이동한 지점에 다시 이동할 방향을 잡고 Line search 알고리즘을 적용하는 방식이다. 그러나 Backtracking 알고리즘을 불균일한 곡선에 적용해본 결과, 원하는 값을 찾지 못하는 것을 확인할 수 있었다. 즉, 이 방법은 적절하지 못하다는 것을 알 수 있었다.

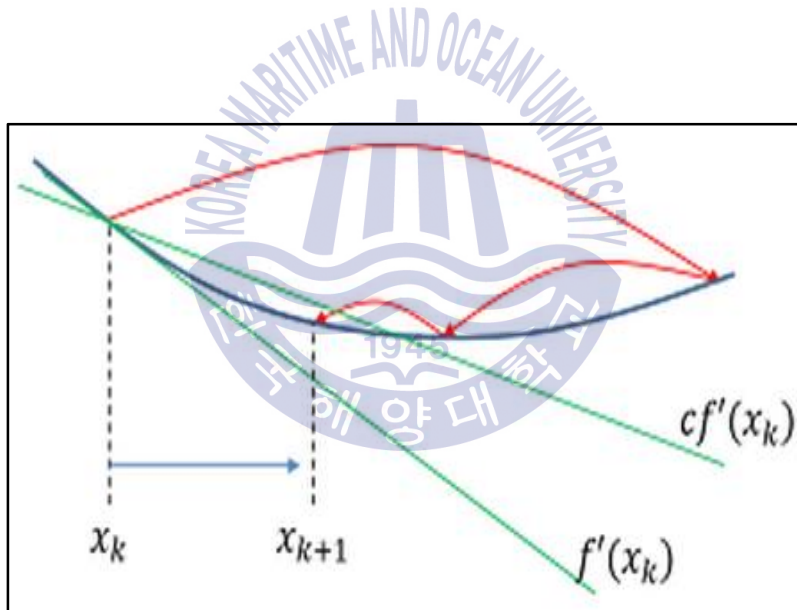


Fig.26 Backtracking line search

제 4 장 최단거리 알고리즘 향상

4.1 프로그램 순서도

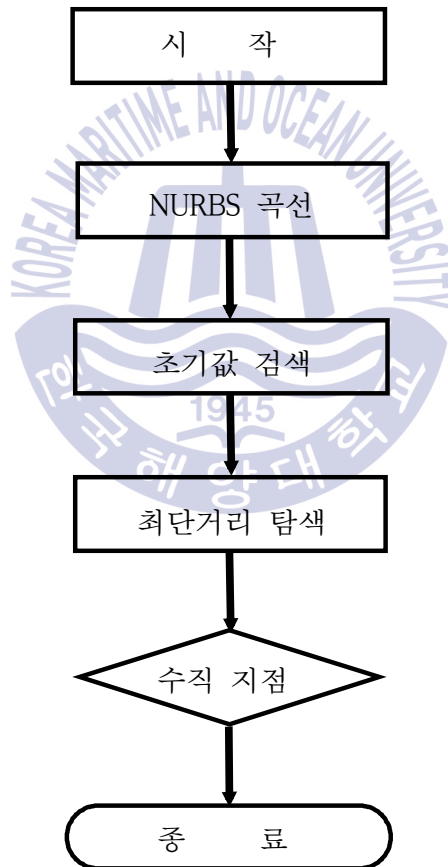


Fig.27 Program Flowchart

4.2 원인 파악 및 분석

4.2.1 Newton-Raphson기법 내의 $f'(t_n)$ 분석

본 연구에 활용된 Newton-Raphson 기법 식을 보면 $f(t_n)/f'(t_n)$ 가 있는데 그 중 분모($f(t_n)$)의 경우에 시간이 지남에 따라(Newton-Raphson이 진행됨에 따라) 큰 변화가 없다. 반면, 분모인 $f'(t_n)$ 와 같은 경우 시간이 지남에 따라 급격히 변화하는 것을 확인할 수 있었다. 특히, 분모의 값이 작은 값을 가질 때 발산하는 현상이 발생하여 한 곳에 수렴하지 못하여 불균일한 곡선에서 원하는 최단거리를 얻지 못하는 것을 알 수 있었다.

4.2.2 초기값에 대한 원인

균일 곡선에서 일정하게 등분한 점과 곡선 밖의 임의의 점과의 거리를 계산하여 각각의 거리 계산 중 최소거리를 초기값으로 시작하여 Newton-Raphson 기법을 적용하여 최단거리를 계산하였다. 이러한 방식은 불균일 곡선에 parameter가 일정하게 등분이 되지 않는 오류가 수반되어 적용되지 않는 것을 확인하였다. 그러므로, 균일 곡선에서 초기값 구하는 방식을 불균일한 곡선에 동일한 방식을 적용할 경우, 원하는 초기값을 얻지 못하여 최단거리를 찾지 못하는 것을 확인하였다.

4.2.3 균일 곡선과 비균일 곡선 초기값 비교

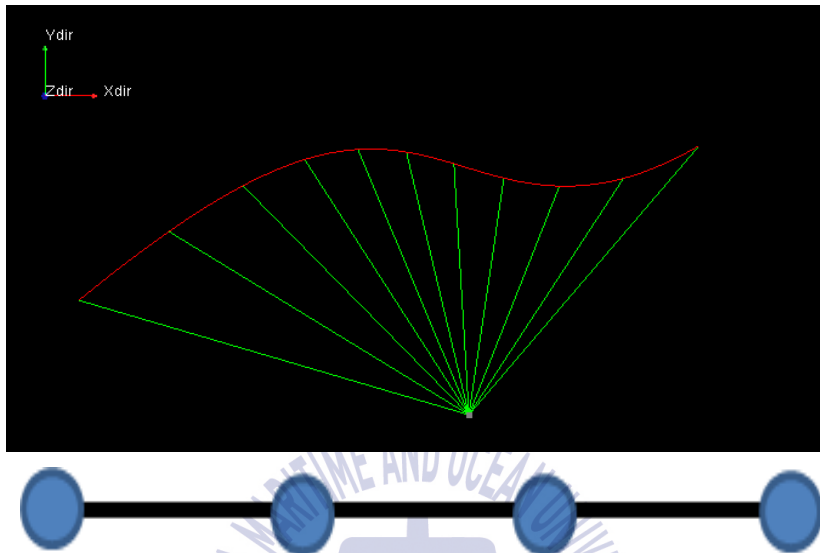


Fig.28 Regular Curve explanation

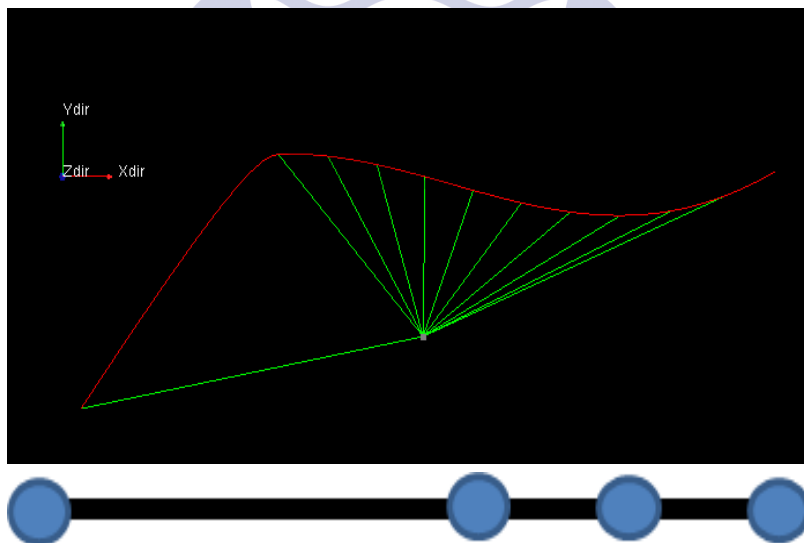


Fig.29 Irregular Curve explanation

Fig. 21에서 곡선과 점과의 관계를 보면 비교적 일정한 간격으로 점과 곡선과의 거리를 구할 수 있다. 등분한 직선거리 중 최소거리를 가지는 곡선 위의 점을 초기점으로 시작하여, Newton-Raphson 기법을 적용한다. 그 결과, 원하는 최단거리가 계산되는 것을 확인 할 수 있다. 즉, 적절한 초기값이 설정이 되었고, 그 초기값으로 Newton-Raphson 기법이 정상적으로 적용이 되는 것을 알 수 있다.

하지만 Fig.22 경우엔, 곡선과 점과의 관계를 보면 균일 곡선과 다르게 점과 곡선과의 거리가 균등한 간격으로 구할 수 없다. 임의의 점과 첫 시작점(파라미터 값이 0인 지점)에서 거리가 구해지고 다음 점은 Knotvector가 0.1인 지점(0.1은 임의로 정해진 값)부터 거리를 구할 수 있다. 0.1인 지점부터 거리를 구할 수 있는데, 일정한 간격으로 거리를 구한 값 중 최소거리를 초기값을 시작하는 것과 달리, 일정한 간격으로 등분이 안 되기 때문에 최소거리를 얻을 수 없다. 그러므로, Newton-Raphson 기법에 적용하여 최단거리 계산이 어렵다.

4.3 원인 분석에 따른 해결방안

4.3.1 곡선의 길이에 따른 등분

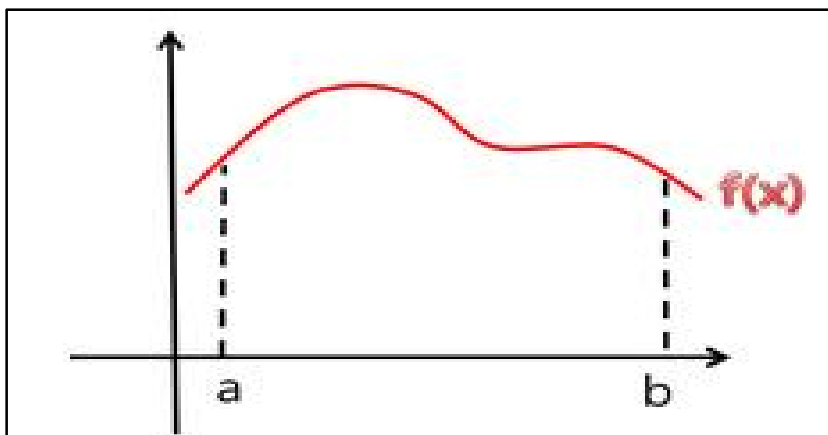


Fig.30 [a, b]에 범위의 곡선길이

함수 $f(x)$ 로 되어있는 곡선의 a에서 b까지의 길이를 구하는 것을 Arc Length라고 한다. Arc Length를 구하는 방법은 곡선을 구분구적법을 통해 미세하게 나눠가며 구간사이의 길이를 점차적으로 근사해 나가는 방식으로 구한다.

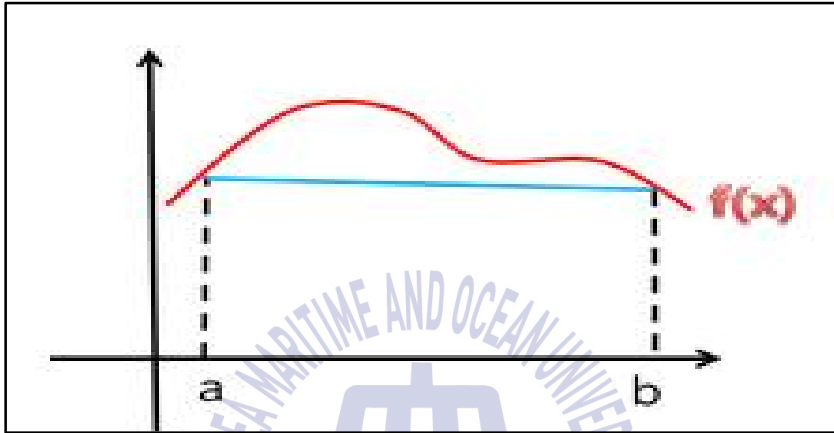


Fig.31 선형을 이용한 곡선 길이

Fig.24는 곡선의 길이를 구하는 가장 간단한 방법이다. 양끝 점을 연결하여 선형의 길이를 구하는 방식이다. 하지만 곡선의 길이가 정확하지 않는 것을 알 수 있다. 이런 방식으로 2구간, 4구간, 8구간 등 구간을 더 많이 나누며, 무한히 많은 구간으로 나눈 후 구간에서의 선형 길이를 구하여 직선의 길이를 합치면 곡선의 길이를 구할 수 있다.

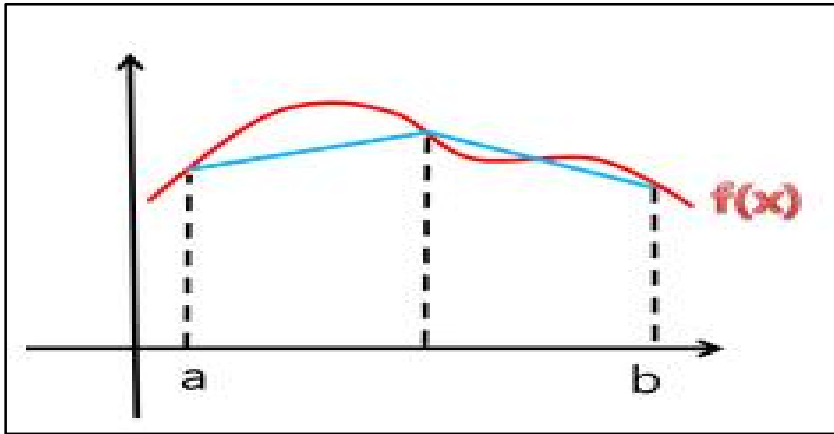


Fig.32 2구간 등분

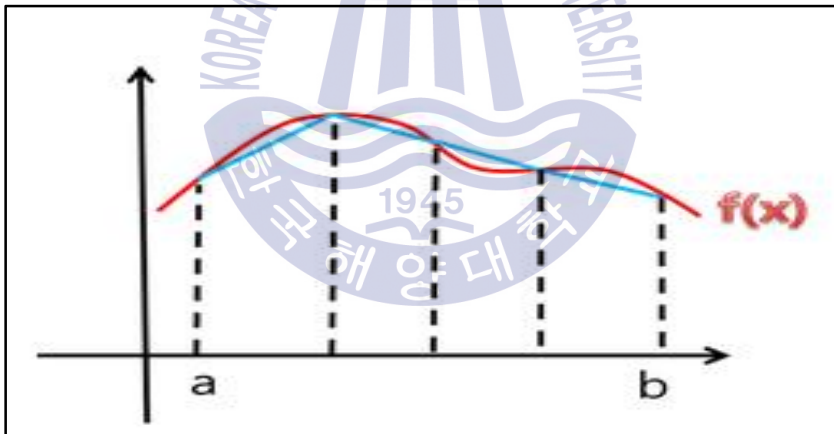


Fig.33 4구간 등분

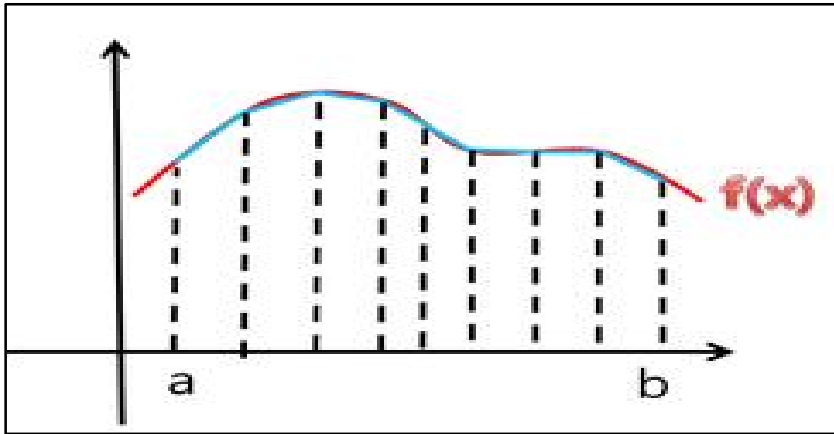


Fig.34 8구간 등분

Fig.25, Fig.26, Fig.27처럼 2구간, 4구간, 8구간으로 늘어날수록, $f(x)$ 곡선의 정확한 길이를 구할 수 있다.

Fig.28은 Arc Length 구하는 구간 중 일부 구간을 확대한 그림이다. 임의의 구간에 대해서 양끝점을 이은 직선은 Fig.28과 같이 표현되며, 그 길이는 피타고라스 정리에 의해

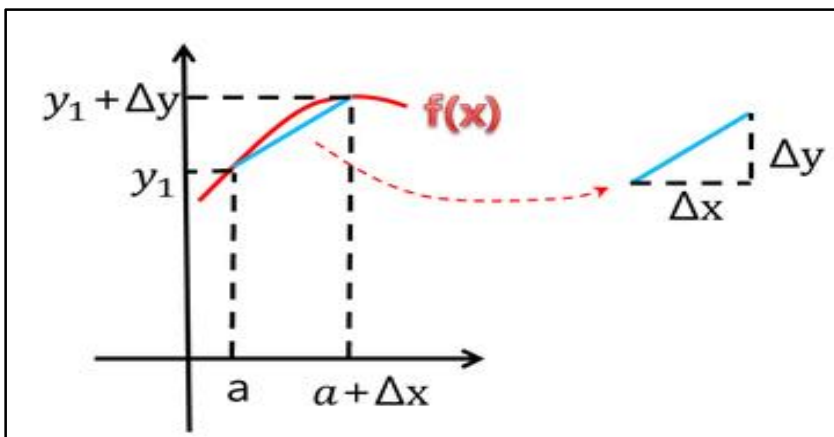


Fig.35 피타고라스에 따른 직선

$l = \sqrt{\Delta x^2 + \Delta y^2}$ 와 같이 나타낼 수 있다. 이러한 직선구간을 구하여 모두 합치면 곡선의 길이를 구할 수 있다.

4.3.2 최대, 최소 범위 설정 및 값 탐색

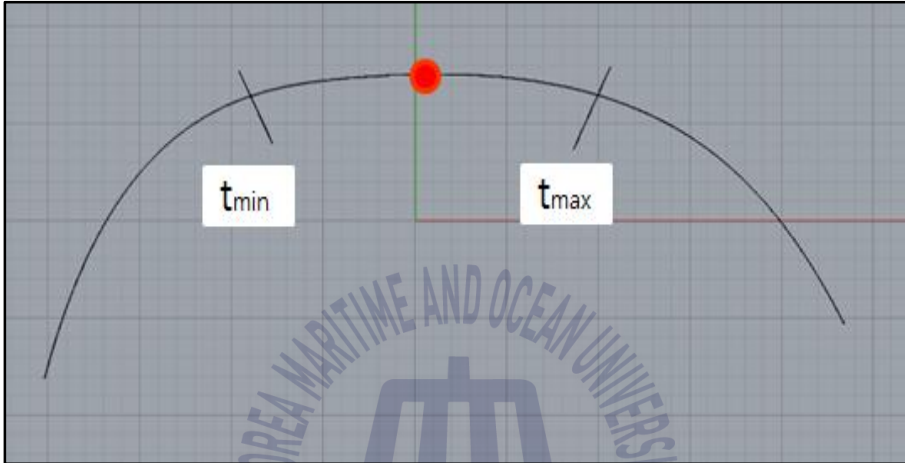


Fig.36 최대, 최소값 설정

앞의 설명과 같이, 파라미터로 곡선을 등분하는 것이 아니라 Arc Length 즉, 곡선의 길이로 균등하게 등분한다. 길이에 따라서 균등하게 등분을 할 경우, 불균일한 곡선의 경우에도 일정한 간격으로 등분이 된다. 위의 그림에서 보는 것과 같이 한 지점을 초기값으로 설정하였을 경우, 최대, 최소값을 초기값 이후의 값과 이전의 값으로 설정하여 초기값 다음의 값이 최대, 최소 범위를 벗어나지 않도록 하는 방법이다. 최대, 최소 범위를 설정하는 이유는 구하고자 하는 최단거리 값이 최대, 최소 범위 안에 존재하기 때문이고, 초기값 다음 값, 즉 두 번째 값이 너무 큰 범위를 벗어나서 발산하는 것을 방지하기 위한 것이다.

4.3.3 Newton-Raphson 식에 adaptive value K를 설정

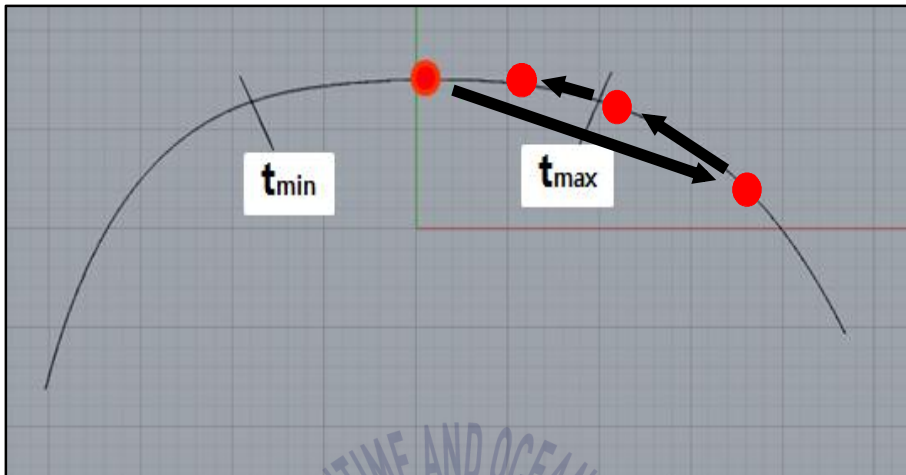


Fig.37 adaptive value K 적용

위의 그림에서 보는 것과 같이 초기 값을 설정해준 후, 다음 값이 최대 범위 밖으로 나가면, Newton-Raphson 식에 adaptive value인 K를 곱하여 최대, 최소범위 안에 들어올 수 있도록 한다. 이 범위 안에 들어오지 못했을 경우 위의 K보다 더 작은 값을 Newton-Raphson 식에 곱하여 준다. 이렇게 최대, 최소 범위 안에 들어올 때까지 이러한 작업을 반복하여 한 곳에 수렴하는 값을 찾는 방법이다.

4.4 향상된 알고리즘 적용

원인 분석 및 향상된 알고리즘을 불균일한 곡선에 적용한 결과이다. 앞에서 언급한 구간별로 Newton-Raphson을 적용해본 결과, 일부 수렴하지 않는 구간이 있었다. 하지만 향상된 알고리즘을 적용해본 결과 초기값에 대한 문제, 발산하는 문제, 원하는 값에 수렴하지 않는 문제 등을 해결함으로써 사용자가 원하는 값에 수렴가능하게되어 불균일한 곡선에서도 최단

거리 계산이 가능한 결과가 나오는 것을 Fig.을 통하여 알 수 있다.

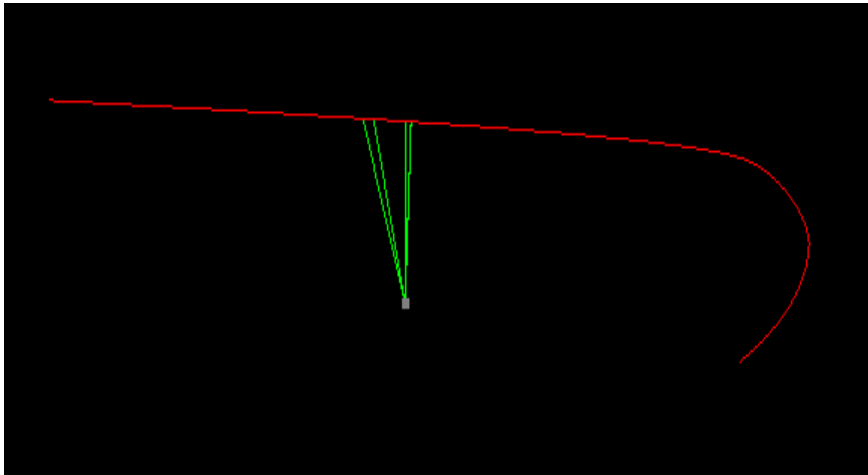


Fig.38 불균일 곡선에 향상된 알고리즘 적용

Table.5 불균일 곡선에 향상된 알고리즘 적용 결과표

x_n	수렴값
x_1	0.050
x_2	0.029
x_3	0.022
x_4	0.022

제 5 장 결 론

5.1 결 과

기존의 상용 3D CAD 프로그램에서 모델링 작업시 후속작업에 사용되는 최단거리 계산할 경우, 균일한 곡선에서는 오류가 수반되지 않으나 불균일한 곡선에서는 최단거리 계산시 오류가 수반되었다. 본 연구에서는 균일 곡선과 불균일 곡선에서의 최단거리 계산을 비교, 분석하여 오류가 수반되는 원인을 찾았으며 균일 곡선에 적용된 최단 거리 계산 방식인 Newton-Raphson법을 이용한 알고리즘에 adaptive value K를 곱하여 항상 시킨 알고리즘을 적용하여 테스트하였다. 그 결과, 불균일한 곡선에서 찾지 못했던 최단 거리를 찾는 것을 확인 할 수 있었으며, adaptive value K를 곱하는 방식은 일정 범위를 벗어나면 적용한 알고리즘을 통해서 범위 안에 들어올 때까지 adaptive value K를 계속 곱하여 준다. 일정 범위 안에 들어오면 이 함수는 종료되면서 최단 거리 계산이 가능하게 된다.

이러한 알고리즘을 통해 적용된 최단 거리 계산 사용자가 쉽게 가시화 할 수 있도록 그래픽라이브러리(OpenGL)과 MFC를 이용하여 GUI 프로그램을 설계하였다.

본 연구를 통해서, 기존 상용 3D CAD 프로그램에서 곡선 모델링 작업시 Refit기능을 통한 곡선이 아니라 사용자가 원하는 곡선을 정제 과정없이 좀 더 정확한 값의 곡선에서 최단 거리 계산이 가능하게 되었으며, 이 최단 거리 계산이 응용되는 OSNAP기능을 정확하게 수행할 수 있게 되었으며 CAD 프로그램이 향후 더 발전될 수 있는 가이드라인을 제시하였다.

5.3 향후 과제

현재 3차원 NURBS 곡선에 한해 향상된 Newton-Raphson 기법을 적용하여 최단거리를 계산하지만 추후, NURBS 곡선뿐만 아니라 NURBS 곡면에도 이와 같은 기법이 적용될 수 있도록 알고리즘을 구현할 필요가 있다. 또한, 기존 상용 3D CAD 프로그램에서의 더 많은 곡선 모델링 후 오류에 대한 검증과 확인이 필요할 것이다. 그리고 이를 통해서 향상된 알고리즘에 대한 더 많은 검증이 필요하며, 검증을 통하여 adaptive value K 값을 결정하는데 더 정확한 값을 계산하고 더 빠르게 값을 얻을 수 있도록 추가 연구 필요성이 있다.



참 고 문 헌

- Piegl L, Tiller W. <<The NURBS Book>>, Springer-Verlag(1997)
- William H.Press, Saul A.Teukolsky, William T.Vetterling, Brian P.Flannery <<NUMERICAL RECIPES in C: The Art of scientific Computing>>, CAMBRIDGE
- cemm#을 활용한 수치해석, 제 2 장 비선형방정식의 근 (www.msharpmath.com)
- Visual C++ 2010 MFC Programming, 정일홍 저, 생능출판사