

최적설계 관점에서의 설계소프트웨어 성능 비교에 관한 연구

홍을표* · 박철민** · 박경진***

A Comparative Study of the Design Software Systems from the View Point of Optimization

U.P. Hong, C.M. Park, G.J. Park

Key Words : Design software(설계소프트웨어), Optimization(최적설계), Design environment(설계환경)

Abstract

Analysis technology is widely accepted and quite popular these days. Incorporation of the analysis result into design process is a key factor for the success of the analysis area. A few design software products have been commercialized. Generally, they are trying to make an interface between various design methods and analysis software. Optimization is a representative design method. The products are investigated and compared for the aspects of user convenience and algorithm performance. A few popular products are selected. Graphic user interface (GUI) is compared for the function and efficiency. The performances of the optimization algorithms are tested by mathematical and engineering examples. The results are discussed.

1. 서론

해석소프트웨어의 성능이 향상되고 안정됨에 따라, 제품 설계에서 해석소프트웨어의 사용 범위와 역할이 점점 확대되고 있다. 기존의 설계는 해석소프트웨어에서 얻어지는 해석 결과만을 이용하여 이루지는 경우가 많았다. 이런 설계과정은 설계자의 직관, 경험, 기술 등에 의해 좌우된다. 설계 전반에 걸친 설계자의 개입은 복잡하게 구성되어 있는 시스템을 설계하는 경우 부정확하고, 비효율적인 설계 결과를 가져 올 수 있다⁽¹⁾. 이런 단점들은 최적 설계와 같은 이미 잘 정립되어 있는 설계 방식을 사용함으로써 극복될 수 있다.

상용 해석소프트웨어 중에는 최적 설계와 같은 설계 기능을 가지고 있는 것들도 있다. 그러나 많은 경우, 해석소프트웨어에는 설계 기능이 포함되어 있지 않다. 또한, 다분야통합최적설계(MDO, Multidisciplinary Design Optimization)⁽²⁾와 같은 설계 방법을 사용하려면, 각 소프트웨어의 설계 기능만

으로는 설계를 수행할 수 없다. 이는 다분야통합 설계가 여러 분야를 동시에 고려하여 서로 연관된 요소들을 설계하는 방식으로, 여러 해석소프트웨어를 동시에 사용하기 때문이다. 이런 경우 설계자는 설계 기능을 가진 프로그램을 직접 개발해야 한다. 그러나 사용하는 소프트웨어의 코드가 없는 경우가 많다. 그리고, 코드가 존재한다고 하더라도, 매우 복잡하거나 설계자에게 익숙하지 않은 언어로 개발된 코드로 구성되어 있는 경우가 많다. 이런 소프트웨어를 이용하여 설계 기능을 가진 소프트웨어를 개발하는 것은 복잡하고 어려운 작업이다.

최근 해석소프트웨어를 제어하여 설계 기능을 제공해 주는 소프트웨어들이 많이 개발되고 있다. 본 논문에서는 이런 기능을 가진 소프트웨어를 설계소프트웨어라고 부르기로 한다. 설계소프트웨어는 설계자가 이미 잘 정립된 설계 기법들을 사용하여 설계를 수행할 수 있도록 하기 위해 개발되었다. 이 소프트웨어는 사용자 환경을 통하여 쉽게 해석소프트웨어에 설계 기능을 추가하여 설계를 수행할 수 있도록 도와준다. 이 기능을 이용하면 설계자는 직접 프로그램을 개발하는 어려움을 피할 수 있다.

* 한양대학교 대학원 기계설계학과

** 포항산업과학연구원

*** 회원, 한양대학교 기계정보산업공학부

Table 1 Software release

Software	Release	Developer
VisualDOC ⁽³⁾	1.2	VR&D
iSIGHT ⁽⁴⁾	5.0	Engineous
OPTIMUS ⁽⁵⁾	2.2	LMS International
ModelCenter ⁽⁶⁾	2.0	Phoenix Integration

설계소프트웨어는 해석소프트웨어의 입/출력 과정을 통하여 설계 데이터를 정의하고, 이를 통해 해석소프트웨어를 이용한 설계가 가능하도록 하는 방식을 사용하고 있다. 그리고 이렇게 얻어진 결과를 설계자에게 표시해 준다. 이러한 기능들을 가진 소프트웨어로는 VisualDOC⁽³⁾, iSIGHT⁽⁴⁾, OPTIMUS⁽⁵⁾, ModelCenter⁽⁶⁾ 등이 있다.

설계소프트웨어의 성능은 여러 항목을 통해 비교될 수 있다. 설계의 정확성과 효율성 역시 하나의 비교 항목이 될 수 있다. 그리고, 설계의 효율성과 정확성은 설계 모듈의 성능과 직접적인 관계가 있다. 이런 관점에서 이번 연구에서는 설계 모듈의 성능을 비교하였다. 상용 설계소프트웨어들은 서로 다른 여러 설계 모듈을 제공하고 있다. 그 중 최적설계와 실험계획법 모듈을 공통적으로 가지고 있다. 특히 최적 설계는 여러 분야에서 연구되고 사용되는 설계방식이기 때문에, 이번 연구에서는 최적 설계 모듈의 성능을 비교해 보았다.

최적 설계는 반복적인 함수계산을 이용하여 최적의 설계점을 찾아가는 방식이기 때문에, 최적점을 찾아가갈 때까지 반복적으로 함수계산을 수행하게 된다. 만일 수학적으로 정식화되어 있거나 해석에 필요한 시간이 작은 경우에는 해석의 횟수는 중요한 문제가 아니다. 그러나, 해석에 많은 시간이 필요한 복잡한 문제에서는 해석의 횟수가 설계 시간과 직결되는 문제이기 때문에 매우 중요한 부분이라고 할 수 있다. 이런 관점에서 설계 모듈의 성능을 비교하기 위하여 최적설계를 수행하여 얻어진 결과와 최적의 결과를 찾는 데 사용된 해석의 횟수를 비교하였다. 몇 가지 수학 문제와 해석소프트웨어를 이용한 설계 문제들이 설계 모듈의 성능을 비교하기 위하여 이용되었다.

본 연구에서 비교된 설계소프트웨어의 버전 등의 정보를 Table 1에 표시하였다.

2. 설계소프트웨어의 기능 및 구성

설계소프트웨어는 Fig. 1과 같이 크게 설계 환경을 정의하고 제어하는 부분과 설계를 수행하는

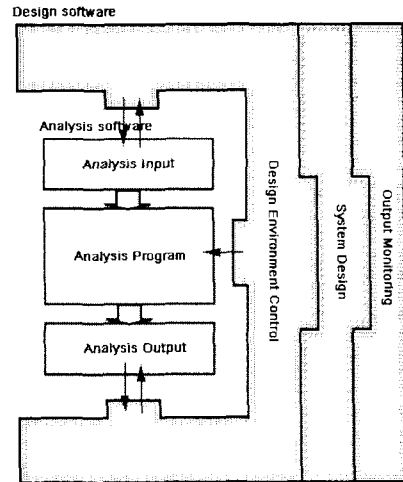


Fig. 1 Functions of the design software packages

부분, 설계 결과를 사용자에게 보여주는 부분의 세 부분으로 구성되어 있다.

설계 환경을 제어하는 부분은 해석소프트웨어를 직접 실행하고, 결과를 얻는 부분이다. 그리고, 해석소프트웨어와 설계 변수나 목적함수와 같은 데이터를 교환하는 부분이기 때문에 설계소프트웨어에서 중요한 부분이다⁽⁷⁾. 이 부분은 설계소프트웨어의 특징에 따라 조금씩 다른 방식을 사용하고 있다. 이런 차이점들이 소프트웨어들의 사용상의 특징과 장단점을 가지도록 한다.

설계 환경이 정의되면, 정의된 해석소프트웨어와 데이터들은 설계소프트웨어에 내장되어 있는 설계 모듈과 연결되어 설계를 수행하게 된다. 설계소프트웨어는 최적설계와 같이 이미 체계적으로 정립된 설계 방식에 대한 설계 모듈을 가지고 있다. 소프트웨어에 따라 제공하는 설계 방식은 조금씩 차이가 있지만, 최적 설계와 실험계획법을 이용한 설계 방식을 공통적으로 제공하고 있다.

위의 두 과정을 거쳐 얻어진 설계 결과들은 설계자에게 표시된다. 이렇게 결과를 표시하는 부분은 설계 성능이나 결과의 정확성과는 거리가 있는 부분이다. 그러나, 설계를 통하여 얻어진 결과들을 설계자들에게 표시해 주는 부분이기 때문에 결코 무시할 수 없는 부분이다.

3. 설계소프트웨어의 특징

설계소프트웨어는 개발 시 중점을 둔 부분에 따라 서로 다른 특징들을 보여주고 있다. 설계소프트웨어의 두드러지는 특징들을 살펴 보면 다음과 같다.

먼저, VisualDOC은 해석 소프트웨어의 제어 기

능을 포함한 설계 환경이 정의되어 있는 함수를 설계자들에게 친숙한 Fortran, C 등의 소스 코드로 생성해 준다. 이 경우, 설계자는 컴파일에 대한 부담을 가지게 된다. 그러나, Fortran 이나 C 와 같은 프로그래밍언어에 익숙한 설계자라면, 함수에 직접 코드를 추가함으로써 목적에 맞도록 설계 환경을 조절할 수 있다는 장점이 있다.

다음으로, iSIGHT 는 사용자의 입장에서 매우 편리한 기능이 많고, 기능들이 매우 안정되어 있다는 장점이 있다. 특히, iSIGHT 는 MDOL 이라는 자체 스크립트를 이용하여 유연하고, 다양한 설계 환경을 정의할 수 있다는 장점을 가지고 있다⁽⁴⁾.

OPTIMUS 는 설계 변수와 입력 파일, 해석 소프트웨어, 출력 파일과 결과값들의 관계를 도식적으로 표현하여 사용자가 정의된 문제의 구조를 쉽게 파악할 수 있도록 구성되어 있다는 장점을 가지고 있었다.

ModelCenter 는 설계 환경을 정의하기 위해 사용되는 Analysis server 라는 소프트웨어가 존재한다. Analysis server 는 설계 환경을 정의하기 위해 사용되는 소프트웨어이다⁽¹¹⁾. ModelCenter 는 네트워크로 연결된 여러 Analysis server 에서 정의된 설계 환경을 가져와 설계 문제를 정의하고, 설계를 수행하는 기능을 한다. 이런 특징 때문에 ModelCenter 는 네트워크 상에 존재하는 해석 소프트웨어를 이용하여 설계를 수행하기에 매우 편리한 소프트웨어이다.

4. 최적설계 모듈의 성능 비교

설계 문제가 정의되면, 설계소프트웨어는 자체 설계모듈을 이용하여 설계를 수행한다. 설계소프트웨어는 Table 2 와 같은 몇 가지의 설계방식을 제공하고 있지만, 이번 연구에서는 공통적으로 제공하고 있는 최적 설계 모듈에 대한 성능을 비교해 보았다. 최적 설계에는 많은 다양한 알고리즘들이 존재하고 있고, 소프트웨어들은 이런 알고리즘들 중에서 몇 가지를 제공하고 있다. Table 3 에 각 소프트웨어가 제공하는 알고리즘들을 표시하였다. 이번 연구에서는 이 가운데 자주 사용되는 몇 가지 알고리즘의 성능을 비교해 보았다. 최적 설계 모듈의 성능 비교를 위해 알고리즘들의 기본 설정을 그대로 이용하였다. 그리고, 몇 가지 수학 문제와 공학문제를 이용하여, 함수계산횟수와 최적 설계 결과값의 크기를 비교하였다.

먼저, 식 1, 2 와 같이 정식화된 수학 문제, 예제 1, 2 에 대해 알고리즘의 성능을 비교해 보았다.

Table 2 Optimization algorithm of design software

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
Optimization	○	○	○	○
Design of Experiments	○	○	○	○
Parameter study	○	○	○	○
Quality Engineering	×	○	×	×

Table 3 Optimization algorithm of design software packages^(1,8,9,12)

	VisualDOC	iSIGHT	OPTIMUS	ModelCenter
1	BFGS	MFD	SQP	VMM
2	FR	MMFD	GRG	CGM
3	MMFD	SLP		MFD
4	SLP	SQP		SLP
5	SQP	SAM		SQP
6		DHS		
7		EP		
8		GA		
9		HJ		
10		SA		
11		GRG		

BFGS : Broyden-Fletcher-Goldfarb-Shannon algorithm

CGM : Conjugate Gradient Method

DHS : Direct Heuristic Search

EP : Exterior Penalty

FR : Fletcher-Reeves algorithm

GRG : Generalized Reduced Gradient

GA : Genetic Algorithm

HJ : Hooke-Jeeves algorithm

MFD : Method of Feasible Directions

MMFD: Modified Method of Feasible Directions

SLP : Sequential Linear Programming

SQP : Sequential Quadratic Programming

SA : Simulated Annealing

SAM : Successive Approximation Method

VMM : Variable Metric Method

Example 1. Rosenbrock's valley problem

$$\begin{aligned} \text{Design Variable} & \quad x_1, x_2 \\ \text{Minimize} & \quad f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \end{aligned} \quad (1)$$

Example 2. Spring design problem⁽⁸⁾

$$\begin{aligned} \text{Design Variables} & \quad d, D, N \\ \text{Minimize} & \quad f = (N + 2)Dd^2 \\ \text{Subject to} & \quad g_1 = 1.0 - \frac{D^3 N}{71875d^4} \leq 0 \\ & \quad g_2 = \frac{D(4D - d)}{12566d^3(D - d)} + \frac{2.46}{12566d^2} - 1.0 \leq 0 \\ & \quad g_3 = 1.0 - \frac{140.54d}{D^2 N} \leq 0 \\ & \quad g_4 = \frac{D + d}{1.5} - 1.0 \leq 0 \end{aligned} \quad (2)$$

식 1 은 구속조건이 없는 문제이고, 식 2 는 구속조건을 가지는 문제이다. VisualDOC 과 ModelCenter 의 경우 구속조건이 있는 문제와 없는 문제에 대해, 그에 적합한 알고리즘을 사용하였으며, iSIGHT 와 OPTIMUS 의 경우에는 일반적으로 많이 사용되는 알고리즘인 MMFD 와 SQP 알고리즘을 이용하였다. Fig. 2, 3 은 최적설계를 통하여 얻어진 설계 결과와 함수계산횟수를 비교한 그림이다.

Fig. 2 는 예제 1 에 대한 결과이다. iSIGHT 의 SQP 알고리즘의 경우에는 함수 계산횟수가 다른 알고리즘들에 비해 상대적으로 많은 것을 볼 수 있었다. Fig. 3 은 예제 2 의 결과를 나타낸 그림이다. 두 번째 예제의 경우, 모든 알고리즘이 비슷한 설계 결과를 얻어내고 있다. 그러나, OPTIMUS 의 SQP 알고리즘과, 특히 iSIGHT 의 SQP 알고리즘은 함수 계산횟수가 상당히 많다는 것을 알 수 있다.

다음으로, 실제 해석소프트웨어와 연결하여 최적설계를 수행한 결과를 비교해 보았다. 예제로는 Fig. 4, 5 의 3 부재 트러스 구조물과 포털프레임 구조물이 사용되었다. 두 예제 모두 구속 조건을 가지고 있고, 질량을 최소화하는 문제이다. 3 부재 구조물에 대한 최적화 문제는 식 3 과 같이 정식화 하였고, 포털프레임 구조물에 대한 최적화 문제는 식 4 와 같이 정식화 하였다. 3 부재 트러스 구조물은 초기 설계가 구속 조건을 만족하고, 포털프레임 구조물의 경우 초기 설계가 구속 조건을 위배하고 있다는 차이가 있다. 이상의 두 예제들도 수학 문제들의 경우에서와 같이 최적 설계로 얻어진 결과와 함수계산횟수를 비교하였다. 해석 소프트웨어로는 Genesis 6.0⁽¹¹⁾을 사용하였다.

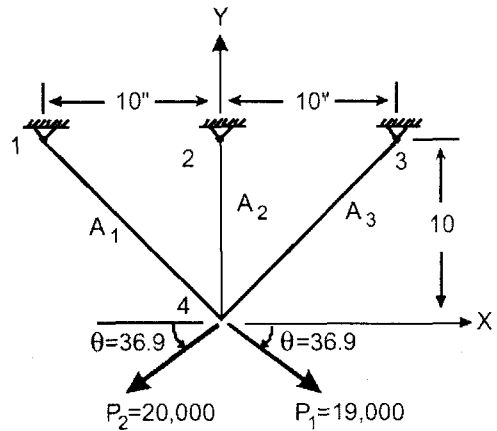


Fig. 4 Problem Specification of three bar truss

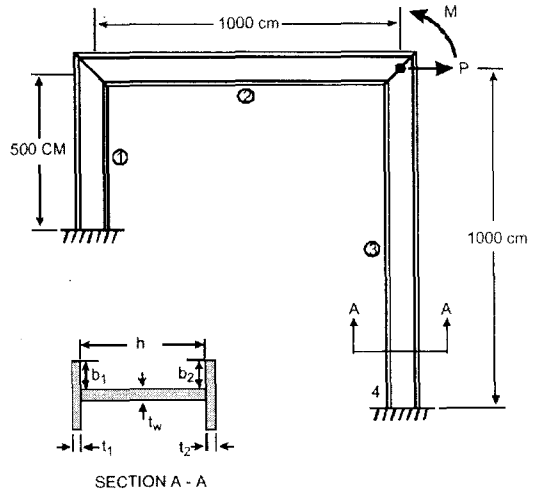


Fig. 5 Problem Specification of portal frame

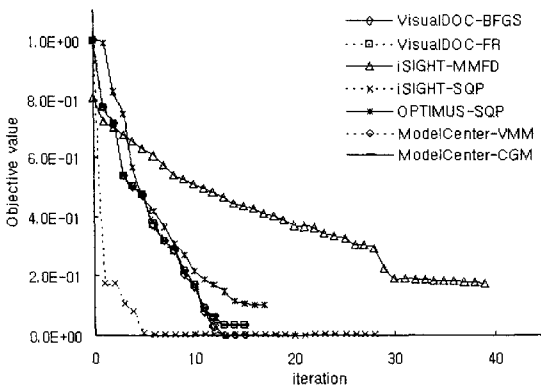


Fig. 2 Optimization results of Example 1

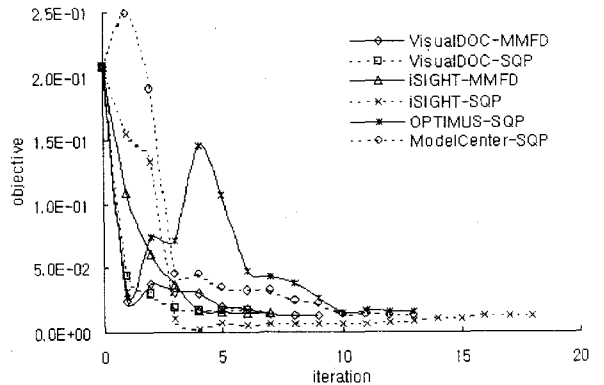


Fig. 3 Optimization results of Example 2

Example 3. Optimization problem for three bar truss structure^(7,11)

Design Variable	A_1, A_2, A_3	(3)
Minimize	Mass	
Subject to	$-15000 \leq \sigma_i \leq 20000 \quad i = 1,2,3$ $-2.0 \leq \delta_{4,j} \leq 2.0 \quad j = x, y$	

where)

σ_{ij} is the stresses of each elements

δ_3 is the displacement in the x and y direction at grid point 4

Example 4. Optimization problem for portal frame structure^(11,12)

Design Variables	Shape of crosssections of the frame	(4)
Minimize	Mass	
Subject to	$-20000 < \sigma_y < 20000 \quad j = 1,2,3, \quad j = 1,2$ $-4.0 < \delta_1 < 4.0$ $-0.015 < \theta_1 < 0.015$	

where)

σ_{ij} is the axial stresses on the top and bottom at each end of the elements

δ_3 is the displacement in the x-direction at grid point 3

θ_3 is the rotation in the z-direction at grid point 3

Fig. 6 과 Fig. 7 에 설계 결과를 표시하였다. 두 결과는 수학 예제들에서의 결과와 비슷한 양상을 나타낸다. Fig. 6 는 3 부재 트러스 구조물의 최적 설계 결과이다. 이 경우에는 iSIGHT 의 경우의 결과가 상대적으로 좋지 못하며, 해석 횟수도 많은 것을 알 수 있다. 특히 Fig. 7 의 포털프레임 구조물의 경우, 목적함수 값이 $2.0E+5$ 부근에 국부 최소값이 존재한다. VisualDOC 과 iSIGHT 의 경우에는 이 값 부근에서 함수계산횟수가 증가하는 것을 볼 수 있다. 이는 알고리즘이 목적함수값의 변화를 섬세하게 파악하고 있기 때문이다. 그러나, OPTIMUS 와 ModelCenter 는 그렇지 못하다는 것을 알 수 있다. 이런 관점에서, OPTIMUS 와

ModelCenter 의 최적화 알고리즘은 목적함수의 변화에 민감하게 대응할 수 있도록 성능을 개선할 필요가 있다고 판단된다.

위의 수학 예제와 공학적 문제들의 결과를 살펴 보면, VisualDOC 의 최적설계 알고리즘들과 iSIGHT 의 SQP 알고리즘이 전반적으로 좋은 결과를 얻어내고 있는 것을 알 수 있다. 하지만 iSIGHT 의 SQP 알고리즘의 경우에는 함수 계산의 횟수가 상당히 많은 특징을 가지고 있다는 것을 알 수 있었다. 전체적인 결과를 통하여 살펴보면 함수계산횟수와 해의 정확성 관점에서 최적설계 모듈의 성능은 VisualDOC 이 가장 우수하다는 결과를 얻을 수 있다.

5. 소프트웨어들의 한계

최근 다양한 분야에 대한 해석소프트웨어가 많이 개발되고, 다분야통합설계와 같이 여러 분야에 대해 동시에 고려하여 설계 하는 경우도 많이 발생하고 있다. 즉, 구조 해석과 유체 해석이 동시에 고려되어야 하는 등의 설계에 대한 연구가 활발히 이루어지고 있다.

본 연구에서 다루어지고 있는 설계소프트웨어 역시 여러 종류의 해석소프트웨어를 연결하여 설계를 수행할 수 있도록 제작되어 있다. 그러나, Fig. 8 의 (a)와 같이, 설계 문제를 정의하는 과정에서 단지 하나의 설계 모듈만을 연결할 수 있다. 이들은 여러 소프트웨어들의 입력과 출력의 상호관계를 정의하여 설계 환경을 정의하고, 정의된 설계 환경에 하나의 설계 모듈만을 연결하여 설계를 수행할 수 있도록 하는 방식을 사용하고 있다.

그러나, 문제의 특징에 따라 Fig. 8 의 (b)와 같이 각각의 해석 모듈에 대해 설계 모듈을 정의하고, 이렇게 정의된 전체 설계 환경에 대하여 설계를 수행해야 하는 경우도 발생할 수 있다. 그러나, 현재의 소프트웨어들로는 이러한 설계 과정을 직

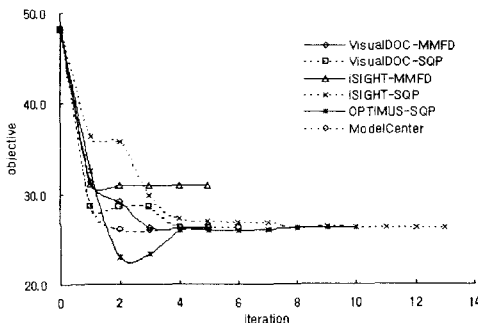


Fig. 6 Optimization results of three bar truss

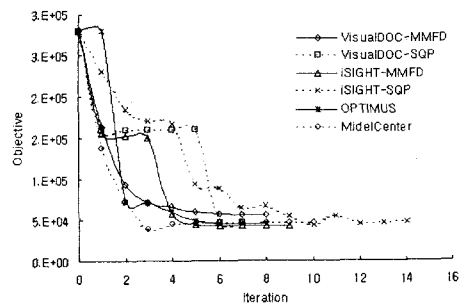


Fig. 7 Optimization results of portal frame

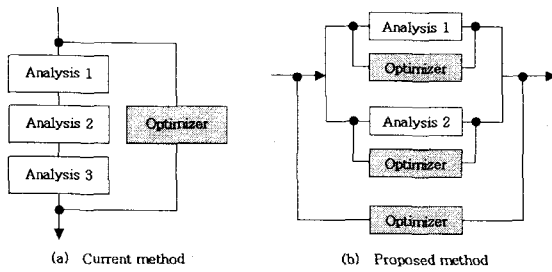


Fig. 8 Comparison of current method and proposed method

접적으로 수행할 수 없다. 최근 다분야통합최적설계에 대한 연구가 활발하게 이루어지고 있고, 이런 부분에 대한 요구가 빈번히 발생하고 있다는 점에서, 설계소프트웨어의 설계에 이런 부분도 고려되어야 할 필요가 있다고 판단된다.

6. 결론

설계를 위한 모듈이 없는 해석소프트웨어와 연결하여 설계를 할 수 있도록 설계된 상용 설계 소프트웨어 VisualDOC, iSIGHT, OPTIMUS, ModelCenter 의 특징을 살펴보고, 그 성능을 비교해 보았다. 특히, 수학적 예제와 공학적 예제들을 통하여 얻어진 설계 결과와 함수계산횟수를 이용하여, 최적 설계 모듈들의 성능에 대해 증점적으로 비교해 보았다.

- I. VisualDOC 은 최적 설계에 비중을 두고 있는 소프트웨어로서 최적설계 모듈의 성능이 매우 뛰어난 장점을 가지고 있었다
- II. iSIGHT 는 설계자가 소프트웨어를 쉽게 사용할 수 있는 특징이 있었다. 그러나, 최적 설계 모듈의 경우 수렴 조건등의 값들에 대한 조정을 통해 최적설계 모듈의 성능을 개선할 필요가 있다고 판단된다.
- III. OPTIMUS 와 ModelCenter 는 해석소프트웨어와의 연결관계에서 사용자의 편의를 고려하여 만들어져 있다. 그러나, 최적설계의 관점에서 많은 부분이 지나치게 간략화 되어있다. 그리고, 설계 모듈의 성능이 문제에 따라 많이 차이가 나는 등, 최적화 부분에 대한 고려가 적다. 따라서, 최적화 관점에 대해 개선이 필요하다고 판단된다.

그리고, 비교 대상이 된 소프트웨어들은 하나의 설계 환경 내에서는 단지 하나의 설계 모듈만을 사용하는 방식을 사용하고 있다는 사실을 알 수 있었다. 최근 연구가 활발히 이루어지고 있는 다분야 통합 설계의 경우에는 하나의 설계 문제 내

부에서 여러 개의 설계 모듈이 사용되는 경우도 존재하고 있다. 이런 문제들의 경우, 비교 대상 소프트웨어들로부터 직접적으로 정의될 수 없었다. 설계소프트웨어는 이런 부분에 대해서 고려를 해야 할 필요가 있다고 판단된다.

후기

이 연구는 한국과학재단지정 최적설계신기술센터와 교육부 지원 BK21 사업 연구비 지원으로 수행되었습니다.

참고문헌

- (1) Arora, J. S., 1989, "Introduction to Optimum Design", Mcgraw-Hill book company, International edition.
- (2) Sobieszczanski-Sobieski, J., Haftka, R.T., 1996, "Multidisciplinary Aerospace Design Optimization : Survey of Recent Developments", AIAA Paper 96-0711, pp. 1-32.
- (3) VisualDOC Manual : version 1.0, 1998, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO.
- (4) iSIGHT Designer' s Guide : version 5.0, 2000, Engineous Software, Inc., Centregreen Way, Suite 100, Cary, NC 27513, North Carolina.
- (5) OPTIMUS Users Manual : version 2.1, 1999, LMS International, Inc., Interleuvenlaan 68-70, B-3001 LEUVEN, BELGIUM.
- (6) ModelCenter Training and Help, 1999, Phoenix Integration, Inc., 1750 Kraft Drive, Suite 2200, Blacksburg, Virginia.
- (7) Ghosh, D.K., Garcelon, J.H., Balabanov, V.O., Vanderplaats G.N., 1999, "Development Of A Flexible Design Optimization Study Tool", presented at the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louse, MO, Sept. 2-4.
- (8) Haug, E.J., Arora, J.S., 1979, "Applied Optimal Design", Wiley & Sons, New York, U.S.A.
- (9) Haftka, R.T. and Gurdal, Z., 1991, "Elements of Structural Optimization", Kluwer Academic Publishers, The Netherlands.
- (10) Brett Malone and Michael Papay, 1999, "ModelCenter: An Integration Environment for Simulation Based Design", SISO workshop, Mar.
- (11) Genesis User Manual : version 6.0, 1998, Vanderplaats Research & Development, Inc., 1767 S. 8th Street, Suit M-210, Colorado Springs, CO.
- (12) Vanderplaats, G. N., 1984, "Numerical Optimization Techniques for Engineering Design", McGraw-Hill book company, New York.