<학술논문> **DOI:10.3795/KSME-A.2011.35.8.921** ISSN 1226-4873

비정렬 다변수 데이터의 B-스플라인 근사화 기법

박 상 근*†
* 충주대학교 기계공학과

On B-spline Approximation for Representing Scattered Multivariate Data

Sangkun Park*†
* Dept. of Mechanical Engineering, Chungju Nat'l Univ.

(Received December 28, 2010 ; Revised May 23, 2011 ; Accepted May 23, 2011)

Key Words : B-spline Hypervolume(B-스플라인 하이퍼볼륨), Scattered Data Approximation(비정렬 데이터 근사법), Least Squares Minimization(최소자승 최소화)

초록: 본 연구는 B-스플라인 하이퍼볼륨을 사용하여 주어진 비정렬 데이터를 근사화하는 데이터 근사 기법에 관한 것이다. 개발 구현을 위한 B-스플라인 하이퍼볼륨의 자료 구조가 기술되며 해당 메모리 크기의 측정을 통해 간결한 표현 모델임을 보인다. 제안하는 근사 기법은 두 가지 알고리즘으로 구성된다. 하나는 B-스플라인 하이퍼볼륨의 절점 벡터 결정에 관한 것이고, 다른 하나는 조정점 결정에 관한 것으로 최소자승 최소화 문제의 해를 구함으로써 얻게 된다. 여기서 구한 해는 데이터 복잡성에 의존하지 않는다. 본 연구 방식은 다양한 형태의 데이터 분포를 가지고 근사 정밀도, 메모리 사용량, 계산 시간 등의 근사화 성능(수준)을 평가한다. 더불어 기존 방법과의 비교를 통해 유용성을 보이며, 비구속 최적화 예제를 통하여 다양한 응용 분야로의 가능성을 보여준다.

Abstract: This paper presents a data-fitting technique in which a B-spline hypervolume is used to approximate a given data set of scattered data samples. We describe the implementation of the data structure of a B-spline hypervolume, and we measure its memory size to show that the representation is compact. The proposed technique includes two algorithms. One is for the determination of the knot vectors of a B-spline hypervolume. The other is for the control points, which are determined by solving a linear least-squares minimization problem where the solution is independent of the data-set complexity. The proposed approach is demonstrated with various data-set configurations to reveal its performance in terms of approximation accuracy, memory use, and running time. In addition, we compare our approach with existing methods and present unconstrained optimization examples to show the potential for various applications.

1. 서 론

1.1 연구 배경

본 연구는 비정렬 다변수 데이터를 B-스플라인 기반의 표현모델에 의해 근사적으로 표현하는 근사화 기법에 관한 것이다. 여기서 비정렬 데이터란 구조화된 규칙적 배열을 가진 격자형 형식이 아닌 불규칙적으로 순서 없이 산만하게 분포된 랜덤 데이터를 말하며, 다변수 데이터란 데이터 존재 공간의 차원이 다차원(데이터 독립변수의 개수가 다수 개)임을 뜻한다. 즉 데이터 분포에 의존하지 않으며 데이터 차원에 무관한 근사 표현 모델을 제시하고 이에 기반을 둔 근사화 기법에 관해 기술하고자 한다.

본 연구에서 다루는 데이터 근사화 문제는 기초과학을 포함하여 여러 공학 분야에서 자주 등장하며 그 물리적 의미가 다를 뿐 공통적으로 고민하는 기본 문제이며, 이에 기초한 다양한 형태의 응용 사례들이 존재한다. 예를 들어, 1) CAD 분야에서 곡선 적합 및 근사 곡면의 생성, 2) 의료분야에서 MRI 및 CAT 스캐닝 장비로부터 획득한 의료 영상으로부터 인체 형상 모델링, 3) 역공학 분야에서 3D 스캐너로 추출된 측정점을 가지고 삼각형 형태의 폴리곤 곡면 재건, 4) 최적화 분야에서 근사 최적화를 위한 반응 표면의 계산,

† Corresponding Author, skpark@cjnu.ac.kr

5) CAE 분야에서 유한 개의 노드 점에서 계산한 등가 응력 데이터를 가지고 임의의 특정 위치에서의 응력 분포 계산 및 데이터 가시화, 6) 지질학 분야에서 측정된 지질 정보로부터 등고선 지도 생성, 7) 유동 가시화 분야에서 계산된 유속 데이터로부터 와류 등과 같은 특징적 유동 구조 추출. 이상과 같이 본 연구에서 다루는 데이터 근사화 문제(해법)는 여러 응용 분야에서 효과적인 데이터 모델링 방법을 지원하며, 원시 데이터 내부에 내재된 특이 정보의 추출 및 분석, 그래픽 처리를 통한 데이터 가시화 등 과학적 데이터의 효과적 처리를 위한 기반 기술로서 그 활용 가치가 높다고 말할 수 있다.

일반적으로 데이터 근사화 기법의 성능 혹은 유용성을 여러 관점에서 평가하기 위하여 고려되어야 하는 사항(논점)[1,2]들을 살펴보면 다음과 같다.

○ **근사 정밀도:** 근사 모델이 얼마나 정확하게 원시 데이터를 근사하고 있는지에 관한 문제로서, 대개 원시 데이터의 복원 정밀도를 RMS 방식에 의해 측정한다.

○ **데이터 복잡성:** 처리 가능한 원시 데이터의 개수 및 데이터 분포 형태에 관한 것으로, 대용량 데이터의 경우 또는 비균일 데이터 분포의 경우 계산 처리가 불가능할 수 있다.

○ **메모리 효율:** 근사화 과정 혹은 최종 근사 모델이 필요로 하는 메모리 크기에 관한 것으로, 대용량 데이터 저장 크기와 근사 모델 사용 메모리 크기를 상대 비교하여 메모리 사용량 효율을 측정할 수 있다.

○ **계산 복잡성:** 근사화 과정에 소요된 계산 시간 및 임의 위치에서의 근사 모델 evaluation 시간에 관한 것으로, 빠른 계산 결과를 필요로 하는 실시간 작업의 경우에 매우 중요하다.

○ **계산 안정성:** 원시 데이터의 크기 및 분포 등에 무관하게 근사화 과정이 수치적으로 안정되게 계산되는 지에 관한 것으로, 대부분의 기존 근사 모델의 경우, 수치적 어려움을 가지고 있다.

○ **근사 완만성:** 근사 모델의 완만성(smoothness)에 관한 것으로, 근사 정밀도와 상충하는 부분이다. 즉 근사 정밀도가 높다고 최적의 근사 결과라고 평가할 수 없다. 교차검증(cross validation)[3]에 관한 고려가 필요하다.

○ **국부 수정:** 기존 데이터가 일부 변경되거나 새로운 데이터가 삽입되었을 때 이를 효율적으로 처리할 수 있는지에 관한 것으로, 근사 모델의 국부적 성질과 관련이 많다.

이상의 논점 측면에서 과거 40 년 동안 제시되어 왔던 기존 방법들을 살펴보면 위에서 기술한 사항 모두를 만족시켜 주는 최선의 근사 모델(혹은 기법)은 존재하지 않는다.[2] 또한 Nielson 논문[4]에 의하면 모든 응용 분야에 활용 가능한 단 하나의 최적 근사 모델은 없으며, 응용 분야별로 각 모델마다 장단점이 있으며, 데이터의 차원, 크기, 분포 등에 의존하여 경우마다 다르다.

본 연구에서 제시하는 근사화 기법도 마찬가지 경우이다. 위의 열거한 모든 측면에서 기존의 타 방법들보다 항상 우수한 결과를 보여주는 것은 아니다. 그러나, 이들 논점 가운데, 기존 방법들의 치명적 결함이며 이로 인해 다른 논점 사항들조차 문제를 일으키는 항목이 바로 메모리 효율이다. 사용하고 있는 근사 모델의 표현 구조와 관련되기 때문에 모델 정의 자체를 변경하지 않는 한 극복하기 어렵다. 바로 이 메모리 효율이 본 연구 제안 모델의 강점으로 현저히 나타난다. (추후 소개할 적용 예제들을 통하여 확인할 수 있다.) 일반적으로 근사 모델 자체가 필요로 하는 메모리 크기가 작을수록, 대규모 데이터의 간결한 근사 표현이 가능하며, 이로 인해 신속한 데이터 처리와 쾌속의 계산을 기대할 수 있다. 결국 위에서 언급한 여러 논점 가운데 메모리 효율은 다른 논점들에 비해 그 영향력이 크며 실용적 측면에서 반드시 고려되어야 할 사항으로써 본 연구 제안 모델의 유용성을 보여주는 것이다. 참고로 근사 정밀도 역시 근사 기법의 성능과 직결되는 사항으로 메모리 효율과는 무관하나 근사 완만성과 반드시 연계하여 함께 고려되어야 한다.

1.2 관련 연구

현재까지 알려진 대표적인 데이터 근사화 모델 혹은 방법으로 Shepard[5] 방식과 RBF(radial basis function)[6] 모델이 있다. Shepard 방식은 거리에 반비례하는 가중치를 사용하여 데이터 값을 보간하는 방식이다. 이 방식은 입력된 데이터 위치(독립변수)에 국부적인 극대 혹은 극소 모양의 데이터 값(종속변수) 분포가 나타나기 때문에 데이터 내부에 내재된 어떠한 국부적 정보도 추출할 수 없는 결함을 가지고 있다. 또한 데이터의 국부적 수정 시 이를 반영시키기 위해 모든 데이터 위치에서의 가중치 값을 재계산해야 하는 문제점을 가지고 있다. 이러한 문제점을 해결하기 위해 Franke와 Nielson 은 MQS(modified quadratic Shepard)[7]을 제시하였는데, Shepard 방식에서 가중치 계산을 국부화시켰고, 데이터 값 자리에 가중치 최소자승법

에 의해 계산되는 2 차 다항식을 삽입하여 사용하였다.

RBF 모델은 관련 기저함수의 형태에 따라 다양한 형태의 개선 모델이 가능하여 최근까지도 여러 분야에서 다양한 호칭으로 사용되고 있다. 대부분 거리 개념의 3 차 기저함수를 사용하는데 비정렬 데이터가 존재하는 볼륨 공간을 보간하는 일종의 볼륨 스플라인 모델이다. 주어진 데이터를 RBF 모델에 대입하여 선형대수 방정식을 유도하고, 이로부터 미지의 계수를 계산함으로써 최종 근사 모델을 생성한다. 여기서 대용량 데이터의 경우, 유도된 선형대수 방정식의 행렬이 이론적으로 정칙 행렬이지만 수치적으로 불량조건(ill-conditioned) 문제이기에 실제 계산이 매우 느리며 불안한 수치 결과를 보여준다. 이러한 문제점을 극복하기 위해 제안된 것이 CS-RBF(compactly supported RBF)[8]이다. 이것은 기존의 무한 영역의 기저함수 영향 범위를 한정된 유한 영역으로 조정함으로써 국부적 성질을 가지게 하여 안정된 수치 계산을 유도한 것이다. 이밖에 RBF 와 유사한 볼륨 표현 모델로서 Hardy 가 multiquadratics[9] 모델을 제안하였고, Duchon 는 곡률 에너지 최소화 개념의 thin plate splines[10]을 제안하였다.

2. B-스플라인 하이퍼볼륨

비정렬 다변수 데이터의 효율적 근사화를 위해 본 연구에서 사용한 근사 모델은 B-스플라인 하이퍼볼륨(B-spline hypervolume)[11]이다. 이 표현 모델은 다차원의 스칼라 장(scalar field)을 격자 형태의 조정점 망(control net)에 의해 정의하는 일종의 텐서곱 스플라인(tensor product spline) 형태를 갖는다.

2.1 수학적 정의

B-스플라인 하이퍼볼륨(B-spline hypervolume)은 마치 B-스플라인 곡선의 매개변수(parameter) 공간을 한 차원 확장하면 2 차원 곡면이 되는 것처럼, 매개변수 공간을 임의의 e 차원으로 확장하고, 나아가 3 차원의 조정점(control point) 벡터의 크기를 임의의 s 개로 확장한, 일반화된 형태의 매개변수형 볼륨 표현 모델이다. 식 (1)은 이 볼륨 표현 모델의 수학적 정의를 보여주고 있다.

e 차원 유클리드 공간 상에 s 개의 스칼라 장(scalar field)이 존재할 때, 이를 식 (1)과 같이 B-스플라인 하이퍼볼륨으로 표현할 수 있다.

$$\mathbf{A}(u_1,\cdots,u_e) = \sum_{i_1=0}^{n_1-1}\cdots\sum_{i_e=0}^{n_e-1} \mathbf{A}_{i_1\cdots i_e} N_{i_1,\mathbf{T}_1}^{k_1}(u_1)\cdots N_{i_e,\mathbf{T}_e}^{k_e}(u_e) \tag{1}$$

여기서 u_j 는 e 차원 유클리드 공간 상의 j 방향 좌표이며, $\mathbf{A}_{i_1\cdots i_e} \in \mathbf{R}^s$ 는 인덱스 좌표 $(i_1,\cdots,i_e)$에 해당하는 조정점이며, n_j 와 k_j 는 u_j 방향으로의 조정점 개수 및 차수(order)를 말하고, $N_{i_j,\mathbf{T}_j}^{k_j}(u_j)$ 는 차수가 k_j 인 정규화된 B-스플라인 기저함수로서 아래의 절점 벡터(knot vector) 상에서 정의된다.

u_j 방향 절점 벡터 $\mathbf{T}_j = \left\{ t_{i_j}^{(j)} \right\}_{i_j=0}^{i_j=n_j+k_j-1}$ $(j=1,\cdots,e)$

위에서 소개한 B-스플라인 하이퍼볼륨은 추후에 소개될 근사법 알고리즘의 간결한 수식 전개를 위하여 아래의 식 (2)와 같이 간결한 표현으로 재작성할 수 있다.

$$\mathbf{A}(\mathbf{u}) = \sum_{I=0}^{N-1} \mathbf{A}_I \mathbf{N}_I(\mathbf{u}) \tag{2}$$

여기서

$N = n_1 \times \cdots \times n_e$

$\mathbf{A}_I = \mathbf{A}_{i_1\cdots i_e}$

$\mathbf{N}_I(\mathbf{u}) = N_{i_1,\mathbf{T}_1}^{k_1}(u_1) \times \cdots \times N_{i_e,\mathbf{T}_e}^{k_e}(u_e)$

$I = i_1 + (n_1)\cdot i_2 + (n_1 \times n_2)\cdot i_3 + \cdots + (n_1 \times \cdots \times n_{e-1})\cdot i_e$

2.2 자료구조 및 메모리 사용량

본 연구에서 구현한 B-스플라인 하이퍼볼륨의 자료 구조를 살펴보면 Fig. 1 과 같으며, 이것에 근거하여 식 (1) 혹은 (2)가 요구하는 메모리 크기는 Table 1 과 같다.

```
class B-spline_Hypervolume
{
  int      m_e;   // space dimension (= e)
  int      m_s;   // vector size of A_I (= s)
  int*     m_n;   // number of control points (= n_j)
  int*     m_k;   // order of B-spline function (= k_j)
  float**  m_T;   // knot vectors (= T_j)
  float**  m_A;   // control points (= A_I)
};
```

Fig. 1 Data structure of a B-spline hypervolume with the class in C++

Table 1 Memory usage of a B-spline hypervolume.

자료형	구성요소	개수
정수형 (4byte)	m_e	1
	m_s	1
	m_n	e
	m_k	e
실수형 (4byte)	m_T	$\sum_{j=1}^{e}(n_j+k_j)$
	m_A	$s\times(n_1\times\cdots\times n_e)$

Table 1 과 같이 B-스플라인 하이퍼볼륨은 정수형 자료로서 $2(1+e)\times 4$ bytes 의 메모리가 사용되며, 실수형 자료로서 $(\sum_{j=1}^{e}(n_j+k_j)+s\times(n_1\times\cdots\times n_e))\times 4$ bytes 의 메모리가 필요하다. 한편, 메모리 사용량 효율을 살펴보면, $e=3$, $s=1$, $n_1=n_2=n_3=7$, $k_1=k_2=k_3=4$ 의 경우에 B-스플라인 하이퍼볼륨의 메모리 크기는 384 byte 로서, 대략 686 개의 데이터 점을 효과적으로 근사할 수 있다(4.3 절의 "조정점 개수의 결정" 부분 참조). 이는 입력 데이터가 차지하는 메모리 크기 대비 14%에 해당하는 크기이다.

3. 데이터 근사화 기법

순서없이 산만하게 분산된 M 개의 데이터 집합 $\mathbf{D}$ 가 주어졌을 때,

$$\mathbf{D}=\{(\mathbf{x}_i,\mathbf{f}_i)\,\big|\,\mathbf{x}_i\in\mathbf{R}^e,\mathbf{f}_i\in\mathbf{R}^s,i=0,\cdots,M-1\}$$

이를 식 (2)의 B-스플라인 하이퍼볼륨에 의해 근사적으로 표현하는 근사화 기법은 다음과 같다.

B-스플라인 하이퍼볼륨을 생성하기 위해서는 Fig. 1 과 같이 e, s, n_j, k_j, $\mathbf{T}_j$, $\mathbf{A}_I$가 정의되어야 한다. 여기서 $j\in\{1,\cdots,e\}$, $I\in\{0,\cdots,N-1\}$이다. M 개의 입력 데이터 $\mathbf{D}$ 로부터 e 와 s 는 결정되며, n_j와 k_j가 주어졌을 때(추후 추가설명 있음), $\mathbf{T}_j$ 는 아래와 같은 알고리즘[12,13]에 의해 결정하며, $\mathbf{A}_I$는 3.1~3.3 절의 제시 방법에 의해 계산한다.

○ 절점벡터 $\mathbf{T}_j=\{t_i^{(j)}\}_{i=0}^{i=n_j+k_j-1}$ 결정 알고리즘

(1) 절점벡터 앞부분 $(i=0,\cdots,k_j-1)$

$$t_0^{(j)}=\cdots=t_{k_j-1}^{(j)}=x_{\min}^{(j)}$$

(2) 절점벡터 뒷부분 $(i=n_j,\cdots,n_j+k_j-1)$

$$t_{n_j}^{(j)}=\cdots=t_{n_j+k_j-1}^{(j)}=x_{\max}^{(j)}$$

(3) 절점벡터 중간부분 $(i=k_j,\cdots,n_j-1)$

$\mathrm{d}=(M-1)/(n_j-k_j+1)$

$l=\mathrm{int}\,((k_j-1+i)\times\mathrm{d})$

$\alpha=(k_j-1+i)\times\mathrm{d}-l$

$$t_i^{(j)}=(1-\alpha)\cdot x_l^{(j)}+\alpha\cdot x_{l+1}^{(j)}$$

여기서 $x_{\min}^{(j)}$ 와 $x_{\min}^{(j)}$ 는 $x_{\min}^{(j)}\le x_i^{(j)}\le x_{\max}^{(j)}$ $(i=0,\cdots,M-1)$로서 입력된 데이터 집합으로부터 계산된다.

○ 조정점 $\mathbf{A}_I$ 계산 알고리즘

조정점은 아래의 식 (3)과 같은 최소화 문제의 해를 구함으로써 계산된다.

$$\underset{\mathbf{A}_I}{\mathrm{Minimize}}\sum_{i=0}^{M-1}\left\|\mathbf{f}_i-\sum_{I=0}^{N-1}\mathbf{A}_I\mathbf{N}_I(\mathbf{x}_i)\right\|_2^2 \tag{3}$$

여기서 식 (3)은 식 (4)과 같이 선형대수 방정식 형태로 재 작성할 수 있다.

$$\underset{\{\mathbf{A}\}}{\mathrm{Minimize}}\;\|\{\mathbf{f}\}-[\mathbf{N}]\{\mathbf{A}\}\|_2^2 \tag{4}$$

여기서

$$\{\mathbf{f}\}=\{\mathbf{f}_0,\cdots,\mathbf{f}_{M-1}\}^T\in\mathbf{R}^{M\times s}$$

$$[\mathbf{N}]=[(\mathbf{N}_I(\mathbf{x}_i))_{i,I}]\in\mathbf{R}^{M\times N}$$

$$\{\mathbf{A}\}=\{\mathbf{A}_0,\cdots,\mathbf{A}_{N-1}\}^T\in\mathbf{R}^{N\times s}$$

3.1 과결정 문제

식 (4)에서 $M\ge N$ 이고, 행렬 $\mathbf{N}$ 의 계수 $\mathrm{rank}[\mathbf{N}]=N$ 일 때, 식 (4)의 최소화 문제는 유일 해(unique solution)를 가지며, 이를 과결정(overdetermined) 문제라 부르며, 여기서 구한 해를 최소제곱해(least squares solution)라 부른다. 목적함수가 양수이고 2 차이므로 목적함수가 최소가 될 필요충분 조건은 다음의 식 (5)와 같이 편미분식이 0 일 때이다.

$$\begin{aligned}&\frac{\partial}{\partial\mathbf{A}}\|\{\mathbf{f}\}-[\mathbf{N}]\{\mathbf{A}\}\|_2^2\\&=2\frac{\partial}{\partial\mathbf{A}}\big(\{\mathbf{f}\}-[\mathbf{N}]\{\mathbf{A}\}\big)\cdot\big(\{\mathbf{f}\}-[\mathbf{N}]\{\mathbf{A}\}\big)\\&=2[\mathbf{N}]^T\big(\{\mathbf{f}\}-[\mathbf{N}]\{\mathbf{A}\}\big)=0\end{aligned} \tag{5}$$

위의 식 (5)에서 미지수 {**A**}에 관해 정리하면 다음의 식 (6)을 얻을 수 있어 결국 N 개의 조정점을 구하게 된다.

$$\{\mathbf{A}\} = ([\mathbf{N}]^T[\mathbf{N}])^{-1}[\mathbf{N}]^T\{\mathbf{f}\} \tag{6}$$

3.2 미결정 문제

식 (4)에서 $M < N$ 이고 rank[**N**] = M 일 때, 식 (4)의 목적함수 값이 0 이 되는 해가 무한 개 존재하는데, 이를 미결정(underdetermined) 문제라 부르며, 흔히 유클리드 노름(Euclidean norm)이 최소인 해를 구한다. 이 해는 유일 해로서 최소노름해(minimum norm solution)라 부른다. 수치적으로 Pseudo-inverse 기법을 적용하면 다음의 식 (7)를 얻을 수 있으며, 이를 통하여 해에 해당하는 N 개의 조정점 {A}를 구한다.

$$\{\mathbf{A}\} = [\mathbf{N}]^T([\mathbf{N}][\mathbf{N}]^T)^{-1}\{\mathbf{f}\} \tag{7}$$

앞 절에서 소개한 과결정 문제와 본 절의 미결정 문제는 모두 행렬 [**N**]이 완전계수(full rank)라 가정하였을 때의 경우로서 수치적으로 양호조건(well-conditioned)에 해당하는 문제이다. 대개 $M >> N$ 인 경우 또는 $M << N$ 인 경우에 그러하다. 그러나, 일반적인 경우에 행렬 [**N**]의 계수(rank)가 부족하다.

3.3 불량조건 문제

행렬 [**N**] $\in \mathbf{R}^{M\times N}$ 의 계수(rank)가 부족할 경우, 즉 rank[**N**] < M 이며 rank[**N**] < N 인 경우, 일반적으로 식 (4)의 목적함수 값과 미지수의 유클리드 노름이 동시에 최소인 해를 찾는다. 이 문제는 수치적으로 안정된 해를 구하기 어려운 불량 조건(ill-conditioned) 문제에 해당한다. 본 연구에서는 수치해를 구하기 위해 SVD(singular value decomposition) 알고리즘과 Pseudo-inverse 기법을 사용하여 근사해를 구한다. 대개 불량 조건은 행렬 [**N**]의 대각선 주변의 요소 크기가 비대각선 요소의 크기에 비해 작을 때 나타난다. 추후 소개될 본 연구 예제들의 경우, $M \approx N$ 일 때 자주 나타나며, 데이터의 분포 형태와도 관련된다. 예를 들어 균일하게 데이터 점들이 분포되어 있지 않으면, 즉 데이터가 존재하는 공간 내부에 데이터 점이 존재하지 않는 영역의 공간이 크다면, 불량 조건일 경우가 크며 이때의 근사 정밀도는 대체적으로 낮게 나타난다. 이러한 사실은 본 연구 적용 예제를 통하여 확인할 수 있다.

4. 수치결과 및 토의

4.1 테스트 함수 및 근사 정밀도 측정

본 연구 근사화 기법의 수행 결과 및 성능 분석을 위하여, 아래와 같은 6 개의 테스트 함수[4]를 사용하였다. 데이터의 독립변수로 (x, y, z)가 있고, 종속변수로 f가 있다. ($e = 3, s = 1$ 인 경우임.)

$$\begin{aligned} f_1(x,y,z) = {} & 0.75\exp\left[-\frac{(9x-2)^2+(9y-2)^2+(9z-2)^2}{4}\right] \\ & + 0.75\exp\left[-\frac{(9x+1)^2}{49}-\frac{(9y+1)}{10}-\frac{(9z+1)}{10}\right] \\ & + 0.5\exp\left[-\frac{(9x-7)^2+(9y-3)^2+(9z-5)^2}{4}\right] \\ & - 0.2\exp\left[-(9x-4)^2-(9y-7)^2-(9z-5)^2\right] \end{aligned}$$

$$f_2(x,y,z) = \left[\tanh(9z-9x-9y)+1\right]/9$$

$$f_3(x,y,z) = \left[(1.25+\cos(5.4y))\cos(6z)\right]/\left[6+6(3x-1)^2\right]$$

$$f_4(x,y,z) = \exp\left[-\frac{81}{16}\left((x-0.5)^2+(y-0.5)^2+(z-0.5)^2\right)\right]/3$$

$$f_5(x,y,z) = \exp\left[-\frac{81}{4}\left((x-0.5)^2+(y-0.5)^2+(z-0.5)^2\right)\right]/3$$

$$f_6(x,y,z) = \sqrt{64-81\left((x-0.5)^2+(y-0.5)^2+(z-0.5)^2\right)}/9-0.5$$

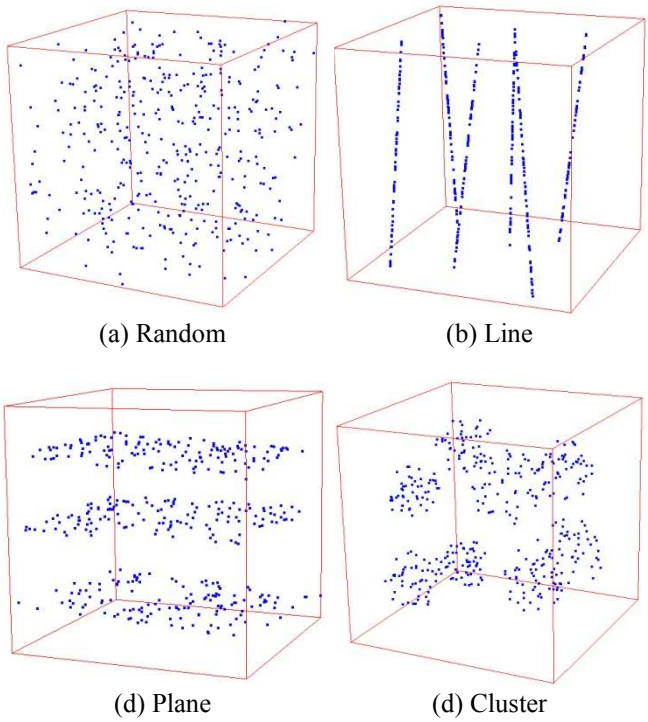
그리고 테스트 함수에서 사용된 데이터 점의 개수(M)는 4.1 절에서 소규모 크기로 M = 50 ~ 400 개를 사용하였고, 4.2 절에서 대규모 크기로 M = 1000 ~ 4000 개를 사용하였다. 또한 데이터 분포 형태로서 Fig. 2 와 같이 (a) 랜덤 형태, (b) 라인 형태, (c) 평면 형태, 그리고 (d) 클러스터 형태로 구분하여 사용하였다.

한편, 이렇게 생성된 데이터의 근사 정밀도는 RMS(root mean square) 오차에 의해 측정하는데, 식 (8)은 입력된 데이터 위치(점)에서의 RMS 오차를, 식 (9)는 데이터가 존재하는 볼륨 공간 내에서 일정 간격으로 생성된 격자 점 위치에서의 RMS 오차를 나타낸다.

○ 데이터 점에서의 정규 RMS 오차

$$RMS_{data} = \frac{\sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\| f(\mathbf{u}_i) - A(\mathbf{u}_i)\|^2}}{(f_{\max} - f_{\min})} \tag{8}$$

여기서 f는 테스트 함수를, $f_{\min}$와 $f_{\max}$는 f의 최소값과 최대값을 나타내며, A 는 본 연구 모델을, $\mathbf{u}_i = (x_i, y_i, z_i)$는 데이터 점을 나타낸다.



(a) Random (b) Line

(d) Plane (d) Cluster

Fig. 2 Data set configurations

○ 격자 점에서의 정규 RMS 오차

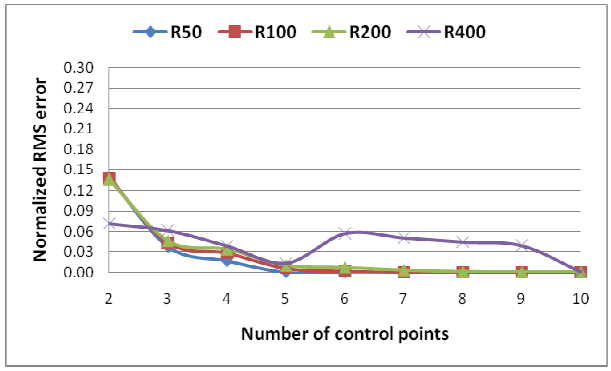
$$RMS_{grid} = \frac{\sqrt{\sum_{i=0}^{M_i}\sum_{j=0}^{M_j}\sum_{k=0}^{M_k} \| f(\mathbf{u}_i) - A(\mathbf{u}_i) \|^2}{(M_i+1)(M_j+1)(M_k+1)}}{(f_{\max} - f_{\min})} \quad (9)$$

여기서 $\mathbf{u}_i = \left(\frac{i}{M_i}, \frac{j}{M_j}, \frac{k}{M_k} \right)$ 는 격자 점을 말하며, $M_i = M_j = M_k = 20$ 을 사용하였다.

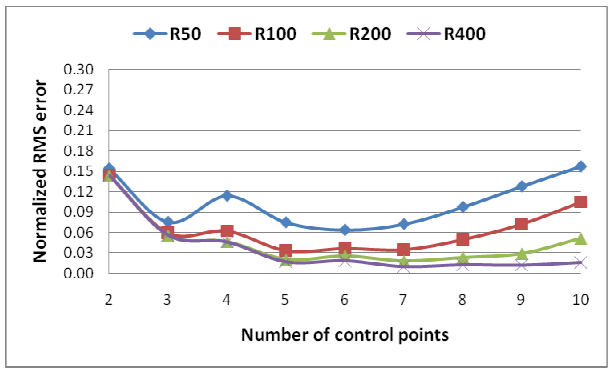
4.2 조정점 개수와 RMS 오차

Fig. 3 은 조정점 개수 변화에 따른 RMS 오차를 보여주고 있다. 사용된 입력 데이터는 R50, R100, R200, R400 이고, 각 경우마다 앞 절에서 소개한 6 개 테스트 함수 각각에 관해 RMS 오차를 계산한 후 이를 평균하여 사용하였다. Fig. 3(a)는 입력 데이터 점에서 오차를 측정한 경우이고, (b)는 격자 점에서 측정한 결과이다.

데이터 점에서 측정한 경우(Fig. 3(a)), 조정점 개수(N)가 많을수록 측정 RMS 오차가 감소됨을 확인할 수 있다. 이것은 B-스플라인 하이퍼볼륨의 조정점 개수가 증가하면 그만큼 기저함수의 개수가 증가, 다시 말하면, 데이터 근사를 위한 모델의 표현 능력이 증가됨을 뜻하며, 이로 인해 측정



(a) Average errors measured at data points



(b) Average errors measured at grid points

Fig. 3 Normalized RMS errors averaged over 6 test functions for random data sets, R50, R100, R200, and R400

오차가 감소됨을 의미한다.

반면에, 격자 점에서 측정한 경우(Fig. 3(b)), 조정점 개수가 증가하면 측정 오차는 감소하다가 증가됨을 확인할 수 있다. 그 이유는 조정점 개수가 증가하면 그만큼 각 조정점의 지배 영역은 좁아지는데, 좁아진 각 조정점이 오직 데이터 점 근처에서의 오차 감소에 활용되기 때문이다.

여기서 주목해야 할 점은 바로 근사 완만성이다. 조정점 개수가 많아지면 국부적 성질이 강해지며 이로 인해 전체적인 완만성을 놓치기 쉽다. 전체적으로 부드러운 완만성을 가지기 위해서는 각 조정점은 함께 변화해 주어야 한다. 현재 이에 관한 개선 안을 개발 중에 있다. Table 2(a)는 Fig. 3 작성에 사용되었던 수치 자료이고, (b)는 근사기법 수행에 소요된 계산 시간이다. 여기서 계산 시간은 조정점 개수가 증가할 때 선형적으로 증가하며, 데이터 점의 경우는 1) 조정점 개수가 2~6 일 때 선형적으로 증가하며, 2) 7~10 일 때 기하급수적으로 증가한다. 참고로 본 연구의 모든 계산 작업은 3.06GHz Intel Xeon CPU 에서 수행되었다.

Table 2 Numerical results shown in Fig. 3. (The symbol 'N' stands for the number of control points.)

(a) Normalized RMS errors

	Data points				Grid points			
N	R50	R100	R200	R400	R50	R100	R200	R400
2	0.139	0.138	0.137	0.072	0.154	0.145	0.144	0.144
3	0.037	0.043	0.046	0.062	0.075	0.059	0.056	0.056
4	0.017	0.028	0.033	0.039	0.114	0.062	0.047	0.046
5	0.000	0.006	0.010	0.013	0.075	0.033	0.021	0.017
6	0.000	0.002	0.007	0.057	0.064	0.037	0.026	0.018
7	0.000	0.000	0.003	0.051	0.072	0.035	0.018	0.010
8	0.000	0.000	0.001	0.045	0.097	0.050	0.023	0.012
9	0.000	0.000	0.000	0.039	0.128	0.073	0.029	0.012
10	0.000	0.000	0.000	0.000	0.157	0.105	0.052	0.016

(b) CPU time measured in seconds

N	R50	R100	R200	R400
2	0.0027	0.0027	0.0102	0.0103
3	0.0027	0.0052	0.0157	0.0290
4	0.0103	0.0207	0.0313	0.0600
5	0.0155	0.0417	0.0833	0.1772
6	0.0208	0.0547	0.1925	0.3697
7	0.0285	0.0832	0.2602	1.0492
8	0.0338	0.0938	0.3153	2.1277
9	0.0467	0.1198	0.4247	2.0678
10	0.0620	0.1640	0.5338	2.4742

4.3 데이터 분포 형태와 RMS 오차

Fig. 4 는 테스트 함수 f_1에 관해 데이터 분포 형태에 따른 RMS 오차를 보여주고 있다. 사용된 데이터는 아래와 같다. 즉,

- 랜덤 형태로서 R1000, R2000, R3000, R4000
- 라인 형태로서 L1000, L2000, L3000, L4000
- 평면 형태로서 P1000, P2000, P3000, P4000
- 클러스터 형태로서 C1000, C2000, C3000, C4000

Fig. 4 (a)는 데이터 점에서 측정한 오차를 보여주고 있으며, (b)는 격자 점에서의 오차를 측정한 결과를 보여주고 있다.

데이터 점에서 오차 측정한 경우(Fig. 4 (a))에 데이터 분포 형태에 무관하게 비교적 작은 RMS 오차가 측정됨을 확인할 수 있다. 반면에 격자 점에서의 오차 측정인 경우(Fig. 4 (b))엔 데이터 점의 경우에 비해 큰 RMS 오차가 측정되었다. 그리고 데이터 분포 형태에 따라 차이가 나타남을 볼 수 있는데, 랜덤 및 클러스터의 경우보다 라인 및 평면의 경우가 상대적으로 큰 오차가 발생했음을 확인할 수 있다. 이는 Fig. 2 에서 보는 바와 같이 라인 및 평면의 경우엔 데이터 점이 존재하지 않는 영역이 존재하기 때문이다. 영역 내에 데이터 점이 없으므로 그 영역에 관한 정보가 없어 이 영역에서 오차가 커지는 것은 당연한 일이다. 즉 정밀한 근사를 위해서는 해당 영역에 데이터 점들이 균등하게 분포되어 있어야 한다. Table 3 (a)은 Fig. 4 작성에 사용되었던 수치 자료이며, (b)는 데이터 분포 형태 별로 소요된 계산 시간을 보여준다. Table 3 (b)의 결과로부터 근사화 소요 시간은 데이터 분포 형태와 무관하며 데이터 개수(M)가 증가할수록 기하 급수적으로 증가함을 확인할 수 있다.

○ 조정점 개수의 결정

본 절의 입력 데이터들을 근사화하기 위해 사용된 B-스플라인 하이퍼볼륨의 조정점 개수는 식 (10)에 의해 결정하였다. 즉 입력 데이터의 개수 M에 관하여, 조정점 개수 N를 경험적으로 $N = M / 2$ 에 의해 결정하였고, N 을 각 방향으로 균등하게 분배한 것이다.

$$n_1 = \cdots = n_e = \mathrm{int}\left(\sqrt[e]{M/2}\right) \tag{10}$$

여기서 $N = n_1 \times \cdots \times n_e$ 이며, n_i 은 i 방향의 조정점 개수이다. 그리고 int(a)는 실수 a 보다 작은 최대 정수를 반환하는 함수를 나타낸다.

대규모 데이터를 가지고 근사 모델을 구할 때, 미결정(3.2 절)의 경우보다 식 (10)과 같은 과결정(3.1 절)의 경우가 메모리 사용량 측면과 계산 시간 측면에서 유리하다. 그 이유는 다음과 같다. 데이터 개수 M 에 관해, 과결정 조정점 개수 N_{over} 과 미결정 조정점 개수 N_{under} 사이에 $N_{over} < M < N_{under}$ 가 성립하므로 생성된 근사 모델의 메모리 크기는 과결정 경우가 적다. 또한 근사화 과정 중에 다루는 행렬 크기를 비교해 보면, 과결정 경우에 $N_{over} \times N_{over}$ 이며 미결정 경우에 $M \times M$ 이므로 역시 과결정 경우가 사용하는 메모리 크기도 작으며 동시에 계산 시간도 적게 걸린다.

4.4 기존 연구와의 비교

입력 데이터 R200 에 관하여 RMS 오차 측면에서 기존의 근사화 방법[4]과 본 연구 방법을 비교한 결과가 Table 4 와 같다. 기존 연구 방법의 각 RMS 오차를 평균한 값과 B 로 표시된 본 연구 방법을 비교해 보면, 본 연구가 f_3, f_4, f_6의 경우 우세하며 나머지는 열세이다. 즉 대등함을 확인할 수 있다.

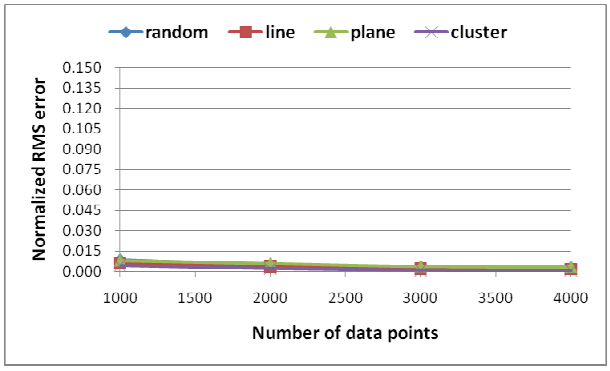
Table 3 Numerical results shown in Fig. 4. (The symbol 'M' denotes the number of data points.)

(a) Normalized RMS errors

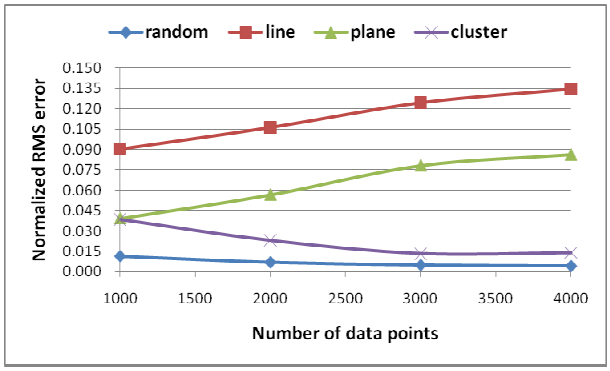
	Data points				Grid points			
M	R	L	P	C	M	L	P	C
1000	0.0088	0.0060	0.0082	0.0045	0.0112	0.0905	0.0392	0.0386
2000	0.0049	0.0037	0.0061	0.0022	0.0068	0.1063	0.0567	0.0231
3000	0.0037	0.0020	0.0039	0.0009	0.0048	0.1243	0.0781	0.0133
4000	0.0033	0.0018	0.0036	0.0008	0.0041	0.1343	0.0862	0.0139

(b) CPU time measured in seconds

M	R	L	P	C
1000	1.7650	1.5470	1.4530	1.7660
2000	13.3440	12.3440	10.5160	12.2660
3000	65.0780	65.7340	55.8750	59.0320
4000	169.6560	212.5780	165.4380	172.6560



(a) Errors measured at data points



(b) Errors measured at grid points

Fig. 4 Normalized RMS errors for data set configurations for test function f_1.

그리고, 테스트 함수 f_1에 관하여 다양한 데이터 분포 형태를 가지고 RMS 오차를 비교한 결과가 Table 5 와 같다. 앞에서 비교한 방식 데로 비교하면, 본 연구가 R1000, C200 의 경우 우세하며 나머지는 열세이다.

Table 4 RMS errors produced by the previous methods compared to the proposed approach for data set R200

	f_1	f_2	f_3	f_4	f_5	f_6
Q	0.0237	0.0134	0.0127	0.0033	0.0089	0.0012
V	0.0120	0.0014	0.0068	0.0006	0.0020	0.0015
H	0.0109	0.0126	0.0042	0.0003	0.0007	0.0018
N	0.0190	0.0121	0.0128	0.0019	0.0045	0.0030
R	0.0131	0.0117	0.0078	0.0008	0.0026	0.0017
B	0.0183	0.0118	0.0052	0.0013	0.0038	0.0012

Q = modified quadratic Shepard
V = volume spline
H = multiquadric
N = volume minimum norm network
R = local volume splines
B = B-spline hypervolume (proposed in this paper)

한편, 메모리 사용량 측면에서 기존 연구와 비교해 보면 다음과 같다. 1.2 절의 관련 연구에서 언급했던 식 (11)의 Shepard 모델과 식 (12)의 RBF 모델의 메모리 요구량은 다음과 같이 계산된다. 여기서 주목할 부분은 두 식 모두 모델 정의를 위하여 입력 데이터가 필요하다는 것이다.

$$S(\mathbf{x}) = \frac{\sum_{i=0}^{M-1} \frac{f_i}{\| \mathbf{x} - \mathbf{x}_i \|^2}}{\sum_{i=0}^{M-1} \frac{1}{\| \mathbf{x} - \mathbf{x}_i \|^2}} \tag{11}$$

$$\phi(\mathbf{x}) = \sum_{i=0}^{M-1} c_i \| \mathbf{x} - \mathbf{x}_i \|^3 + ax + by + cz + d \tag{12}$$

여기서 $\mathbf{x}_i$는 입력 데이터 위치, f_i 는 데이터 값, c_i 및 a, b, c, d 는 데이터로부터 계산되는 계수이다.

e 차원 공간에 데이터 종속변수의 크기가 s 인 M 개의 데이터가 존재할 경우, 식 (11)과 식 (12)의 메모리 크기는 각각 아래와 같다.

Shepard 모델 : $(e+s) \times M \times 4$ bytes
RBF 모델 : $((e+s) \times M + s \times 4) \times 4$ bytes

예를 들어, $e = 3$, $s = 1$, $M = 686$ 인 경우에 Shepard 모델은 10,976 bytes 가 사용되며, RBF 모델은 10,992 bytes 가 사용된다. 이 결과는 2.2 절에서 언급한 본 연구 모델인 B-스플라인 하이퍼볼륨의 메모리 사용량 크기의 약 29 배에 해당한다. 결국 메모리 측면에서 본 연구 방식이 식 (11)의 Shepard 모델과 식 (12)의 RBF 모델보다 상당히 우세함을 확인할 수 있다.

Table 5 RMS errors compared between the existing methods and the proposed approach for test function f_1

	R125	R200	R1000	L200	P200	C200
Q	0.0293	0.0237	0.0150	0.0328	0.0332	0.0454
V	0.0265	0.0120	-	0.0237	0.0206	0.0268
H	0.0218	0.0109	-	0.0177	0.0135	0.0349
N	0.0327	0.0190	0.0040	0.0346	0.0257	0.0429
R	0.0259	0.0131	0.0014	0.0235	0.0217	0.0257
B	0.0276	0.0183	0.0045	0.0693	0.0256	0.0237

4.5 타 분야 응용 사례

본 연구에서 제시한 데이터 근사 방법은 서론에서 언급한 바와 같이 다양한 형태의 응용 분야에 활용될 수 있다. 그 응용 사례로서 비구속 최적화 문제로의 활용 예를 살펴보면 다음과 같다.

설계변수가 2 개인 비구속 최적화 문제는 식 (13)과 같이 정의할 수 있으며, 최적 해를 구하기 위한 필요조건은 식 (14)와 같다.

$$\text{Minimize } f(x_1, x_2) \tag{13}$$

$$\frac{\partial f}{\partial x_1} = 0,\ \frac{\partial f}{\partial x_2} = 0 \tag{14}$$

그리고 식 (14)의 해를 구하기 위한 수치적 방법으로 식 (15)와 같은 Newton-Raphson 방법을 사용할 수 있다.

$$\begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_1}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_1}\right) \\ \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right) & \frac{\partial}{\partial x_2}\left(\frac{\partial f}{\partial x_2}\right) \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \end{Bmatrix} = \begin{Bmatrix} -\frac{\partial f}{\partial x_1} \\ -\frac{\partial f}{\partial x_2} \end{Bmatrix} \tag{15}$$

이 방법은 계산 속도 면에서 비교적 빠르지만 이를 위해선 최적 해 근처의 초기치가 필요하며, 단점으로 목적함수 f의 2 차 미분 값을 요구한다.

그러나, 본 연구 근사 모델은 이러한 초기치와 2 차 미분 값을 수치적으로 안정되며 신속하게 제공할 수 있다. 먼저 주어진 설계 공간 상에서 아래의 식 (16)과 같은 M 개의 데이터를 구성한다.

$$\mathbf{D} = \{(\mathbf{x}_i, f_i) \mid \mathbf{x}_i \in \mathbf{R}^2, f_i \in \mathbf{R}, i = 0, \cdots, M-1\} \tag{16}$$

그리고 식 (16)의 입력 데이터를 가지고 3 장의 근사 기법을 사용하여 식 (17)의 B-스플라인 하이퍼볼륨 근사 모델을 생성한다.

Step 1 Find p and q such that
$f_{pq} = \min(f_{ij})(i = 0, \cdots, n_1 - 1,\ j = 0, \cdots, n_2 - 1)$

Step 2 Compute the Greville abscissa by using

$$\xi_1 = \frac{t^{(1)}_{p+1} + \cdots + t^{(1)}_{p+k_1-1}}{k_1 - 1}$$

$$\xi_2 = \frac{t^{(2)}_{q+1} + \cdots + t^{(2)}_{q+k_2-1}}{k_2 - 1}$$

Step 3 Return the initial guess, $\mathbf{x}_0 = (\xi_1, \xi_2)$

Fig. 5 Finding initial guess to be used for Eq. (15)

Step 1 Set $\mathbf{x}_0 = (\xi_1, \xi_2)$ and $\mathbf{x} = \mathbf{x}_0$
Repeat until $k < k_{\max}$
{
Step 2 Compute the partial derivatives, $\frac{\partial^{i+j} f}{\partial^i x_1 \partial^j x_2}$
Step 3 Solve the system of linear equations Eq. (15)
Step 4 If $\|\Delta\mathbf{x}\| < \varepsilon_x$ or $\|\partial f / \partial \mathbf{x}\| < \varepsilon_f$
Then set $\mathbf{x}^* = \mathbf{x}$ and return the solution, $\mathbf{x}^*$
Step 5 Set $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}$
}

Fig. 6 Newton-Raphson algorithm for solving Eq. (15)

$$f(x_1, x_2) = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} f_{ij} N_i^{k_1}(x_1) N_j^{k_2}(x_2) \tag{17}$$

식 (17)로부터 Fig. 5 의 초기치 결정 알고리즘에 의해 최적 해에 근접한 초기 해를 구한다. 즉, 조정값이 최소인 조정점의 인덱스 좌표 (p, q)를 구한 후, 이에 해당하는 Greville abscissa[14]를 구하면 이것이 바로 초기 해가 된다.

마지막으로 구해진 초기 해를 가지고 Fig. 6 의 반복순환 알고리즘에 의해 최종 해를 구한다. Fig. 6 에서 k 는 반복 회수를 나타내며, ε_x 와 ε_f 는 10^{-6} 로 설정하였다. 그리고 step 2 의 편미분 계산은 deBoor 알고리즘[13]에 의해 구한다.

결국, 식 (13)의 비구속 최적화 문제는 B-스플라인 하이퍼볼륨에서 제공하는 Greville abscissa 를 사용한 초기치와 deBoor 알고리즘에 의한 편미분 계산에 의해 쉽게 해결된다. 여기서 목적함수가 반드시 명시적으로 정의될 필요는 없다. 즉 입력 데이터만 있어도 식 (17)의 근사 모델을 생성할 수 있다.

○ 간단한 수학 문제

앞에서 기술한 B-스플라인 하이퍼볼륨 기반의 최적화 기법을 사용하여, 식 (18)과 같은 수학 문제의 최적 해를 구해보면 Table 6 (a)와 같이 상대오차가 0.0621 %인 비교적 정확한 해를 빠르게 얻을 수 있었다. 본 예제에서 사용된 비정렬 데이터는 R100 이며, 차수는 $k_1 = k_2 = 4$ 로 설정하였고, 조정점 개수는 식 (10)에 의해 $n_1 = n_2 = 7$ 을 사용하였다.

$$f(x_1, x_2) = (x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{18}$$

○ 2 개 막대로 이루어진 트러스 문제

기계공학 분야의 전형적인 샘플 문제로서 Fig. 7 (a)와 같이 2 개의 막대로 이루어진 트러스에 힘 P_1 과 P_2 가 가해졌을 때 점 R 의 변위요소 x_1 과 x_2 을 계산하는 문제이다. 이 문제의 해는 아래에 주어진 위치 에너지 f 를 최소화함으로써 얻어진다. 본 예제의 해를 구하기 위해 사용된 입력 데이터는 R32 이고, 차수는 $k_1 = k_2 = 4$ 로 설정하였고, 조정점 개수는 식 (10)에 의해 $n_1 = n_2 = 4$ 를 사용하였다. 수치 결과는 Table 6 (b)와 같이 단 한번의 반복순환에 의해 정답을 구할 수 있었다.

$$f(x_1, x_2) = \frac{EA}{s}\left(\frac{L}{2s}\right)^2 x_1^2 + \frac{EA}{s}\left(\frac{H}{s}\right)^2 x_2^2 - P_1 x_1 - P_2 x_2 \tag{19}$$

여기서 E 는 탄성계수, A 는 각 막대의 단면적, s 는 각 막대의 길이, L 은 밑면의 길이, P_1 과 P_2 는 수평과 수직방향으로 작용하는 하중이다. (H = 30 in, E = 30×10^6 psi, A = 1 in^2, $L = s = 20\sqrt{3}$ in)

○ 1 차원 핀 열전달 문제

본 예제는 간단한 1 차원 열전달 문제로서, Fig. 7 (b)와 같이 1 차원 핀의 점 1 과 2 에서 정상 상태의 온도 T_1, T_2 를 구하는 문제이다. 이 문제는 식 (20)를 최소화함으로써 해를 얻을 수 있는데, 이를 위해 사용된 입력 데이터는 R50, 차수는 $k_1 = k_2 = 4$, 조정점 개수는 식 (10)에 의해 $n_1 = n_2 = 5$ 을 사용하였다. 수치 결과는 Table 6 (c)와 같이 한번의 반복순환에 의해 비교적 정확한 근사 해를 구할 수 있었다.

$$f(T_1, T_2) = 0.4380 \cdot T_1^2 - 0.0810 \cdot T_1 \cdot T_2 + 0.2190 \cdot T_2^2 - 39.9114 \cdot T_1 - 14.2857 \cdot T_2 + 2292.4020 \tag{20}$$

Table 6 Iteration history for the sample unconstrained optimization problems

(a) Simple mathematical problem, Eq. (18)

Iteration	x_1	x_2	Absolute Error [2]	Relative Error [3]
0 [1]	1.211729	1.659639	0.692786	0.489874
1	1.047018	1.132305	0.140411	0.099286
2	0.969576	0.920460	0.085160	0.060217
3	1.001091	1.000172	0.001104	0.000781
4	1.000146	0.999135	0.000877	0.000620
5	1.000145	0.999134	0.000878	0.000621
$\mathbf{x}^{true}$	1.000000	1.000000		

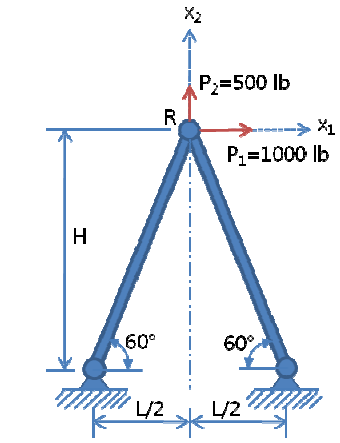
(b) Two bar truss problem, Eq. (19)

Iteration	x_1	x_2	Absolute Error [2]	Relative Error [3]
0 [1]	0.016495	0.016802	0.021770	9.355220
1	0.002309	0.000289	0.000000	0.000000
$\mathbf{x}^{true}$	0.002309	0.000289		

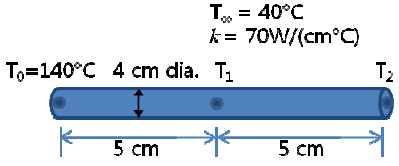
(c) One dimensional pin problem, Eq. (20)

Iteration	x_1	x_2	Absolute Error [2]	Relative Error [3]
0 [1]	48.397880	41.611682	1.033759	0.015978
1	49.421905	41.755421	0.000306	0.000005
$\mathbf{x}^{true}$	49.421600	41.755400		

1) initial guess computed from the algorithm shown in Fig. 5
2) absolute error measured from $\| \mathbf{x}^* - \mathbf{x}^{true} \|$
3) relative error measured from $\| \mathbf{x}^* - \mathbf{x}^{true} \| / \| \mathbf{x}^{true} \|$



(a) Two bar truss



(b) One dimensional pin

Fig. 7 Engineering sample problems

5. 결 론

본 연구에서는 비정렬 다변수 데이터를 효율적으로 근사화하기 위하여, 근사화 모델로서 B-스플라인 하이퍼볼륨에 관해 소개하였고, 이에 기반을 둔 근사화 기법에 관해 기술하였다. 그리고 조정점 개수 변화에 따른 RMS 오차의 변화 추이를 살펴보았고, 4 가지의 데이터 분포 형태 별로 측정된 RMS 오차를 가지고 본 연구 기법의 유용성을 확인할 수 있었고, 또한 기존 연구와의 비교를 통하여 근사 정밀도 및 메모리 사용량 측면에서 경쟁력이 존재함을 확인할 수 있었다. 한편, 본 연구 근사화 기법의 타 분야 응용 사례로서 비구속 최적화 문제의 근사해 계산 과정을 소개하였고, 샘플 문제의 수렴성 확인을 통해 본 연구 방식의 응용 가능성을 확인하였다.

본 연구 근사화 기법의 주요 특징을 살펴보면 다음과 같이 요약된다. 1) 본 연구 근사화 기법은 데이터 크기(개수)에 제한을 받지 않으며 그 순서에도 무관하다. 또한 2) 데이터 분포 형태가 비균일한 경우(예: Fig. 2 의 라인 형태, 평면 형태, 클러스터 형태)에도 즉 불량 조건의 문제임에도 안정된 근사 결과를 제공한다. 3) 비교적 적은 양의 메모리를 가지고 기존 방법과 대등한 근사 정밀도를 제공할 수 있다. 4) 조정점 개수의 증감을 통해 근사 정밀도를 조정할 수 있다. (과결정 문제의 경우에 조정점 개수가 증가시켜 근사 정밀도는 높일 수 있다.) 5) 데이터 공간 상의 임의의 위치에서 0 차 미분을 포함하여 1 차 이상의 미분 값을 저렴하게 제공할 수 있다. 6) 데이터 근사화 이후, B-스플라인 연관 분야에서 기 연구된 유용한 성질(예: convex hull) 등을 활용하여 데이터 상에 내재된 유용한 정보를 추출할 수 있다. (예: 최적화 수행에 필요한 초기해 계산)

향후 연구 과제로서 1) 근사 정밀도와 더불어 근사 완만성도 함께 고려하는 가중치 근사 기법에 관한 연구, 2) 본 연구 방식의 반복 적용을 통한 다중 레벨 형태의 계층적 근사화 기법에 관한 연구 등이 근사화 품질 개선을 위해 필요하다. 더불어 3) 주어진 오차 범위 내에서 데이터를 효과적으로 근사화 하는 오차 유계(error-bounded) 근사화 기법에 관한 연구도 필요하다.

후 기

이 논문은 2011 년도 충주대학교 교내학술연구비의 지원을 받아 수행한 연구임.

참고문헌

(1) Park, S., 2007, “Multiresidual Approximation of Scattered Volumetric Data With Volumetric Non-Uniform Rational B-Splines,” *Trans. of the Society of CAD/CAM Engineers*, Vol. 12, No. 1, pp. 27~38.

(2) Haber, J., Zeilfelder, F., Davydov, O. and Seidel, H-P, 2001, “Smooth Approximation and Rendering of Large Scattered Data Sets,” *12th IEEE Visualization 2001*, pp. 341~571.

(3) Kohavi, R., 1995, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” *Proc. of International Joint Conference on Artificial Intelligence.* Vol. 2, No. 12, pp. 1137–1143.

(4) Nielson, G. M., 1993, “Scattered Data Modeling,” *IEEE Computer Graphics and Applications*, Vol. 13, No. 1, pp. 60~70.

(5) Shepard, D., 1968, “A Two Dimensional Interpolation Function for Irregularly Spaced Data,” *Proc. of ACM 23rd National Conference*, pp. 517~524.

(6) Schaback, R., 1995, “Multivariate Interpolation and Approximation by Translates of a Basis Function,” in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, World Scientific Publishing, Singapore, pp. 491~514.

(7) Franke, R. and Nielson, G. M., 1980, “Smooth Interpolation of Large Sets of Scattered Data,” *International Journal of Numerical Methods in Engineering*, Vol. 15, pp. 1691~1704.

(8) Wendland, H., 1995, “Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree,” *Advances in Computational Mathematics*, Vol. 4, pp. 389~396.

(9) Hardy, R., 1971, “Multiquadric Equations of Topography and Other Irregular Surfaces,” *J. Geophysical Research*, Vol. 76, No. 8, pp. 1905~1915.

(10) Duchon, J., 1975, “Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces,” in *Multivariate Approximation Theory*, Basel, Switzerland: Birkhauser, pp. 85~100.

(11) Park, S., 2009, “A Rational B-spline Hypervolume for Multidimensional Multivariate Modeling,” *J. Mech. Sci. and Tech.*, Vol.23, pp. 1967~1981.

(12) Piegl, L. and Tiller, W., 1995, *The NURBS Book*, Springer-Verlag.

(13) De Boor, C., 1978, *A Practical Guide to Splines*, New York, Springer-Verlag.

(14) Farin, G., 1990, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego.