

딥러닝 샘플 생성을 위한 포아존-볼츠만 방정식의 대수적 멀티그리드를 사용한 수치 예측

신광성¹ · 조광현^{2*}

An algebraic multigrids based prediction of a numerical solution of Poisson-Boltzmann equation for a generation of deep learning samples

Kwang-Seong Shin¹ · Gwanghyun Jo^{2*}

¹Associate Professor, Department of Digital Content Engineering, Wonkwang University, Iksandae-ro, Iksan, 54538 Korea

^{2*}Assistant Professor, Department of Mathematics, Kunsan National University, Guansan-si, 54150 Korea

요 약

포아존 볼츠만 방정식 (Poisson-Boltzmann equation, PBE)은 생물물리, 콜로이드 화학 등에서 등장하는 문제들을 모델링하는데 사용되는 방정식이다. 따라서 PBE의 수치해를 효율적으로 예측하는 것은 중요한 이슈이다. 저자들은 기존의 연구에서 PBE를 풀기 위한 딥러닝 방법을 제안하였으나, 딥러닝을 훈련하기 위한 샘플을 생성하는 시간이 컸다는 어려움이 있었다. 본 논문에서는 FEM 수치해를 생성하는데 걸리는 시간을 줄이는 두가지 방안을 마련하였다. 첫째로 대수 방정식을 만들 때 bilinear form에 포함되는 penalty 파라미터를 실험적으로 조정하였다. 두 번째로, 대수적 멀티그리드 기법을 활용하여 대수 방정식의 컨디션 넘버를 meshsize와 무관하게 만들었다. 따라서 PBE 방정식의 대수 방정식을 풀 때 계산 시간을 효과적으로 줄였다. 이러한 대수적 멀티그리드를 사용한 방법은 다양한 분야에서 딥러닝의 샘플을 생성하는데 효과적으로 활용될 수 있을 것으로 기대된다.

ABSTRACT

Poisson-Boltzmann equation (PBE) is used to model problems arising from various disciplinary including bio-physics and colloid chemistry. Therefore, to predict a numerical solution of PBE is an important issue. The authors proposed deep learning based methods to solve PBE while the computational time to generate finite element method (FEM) solutions were bottlenecks of the algorithms. In this work, we shorten the generation time of FEM solutions in two directions. First, we experimentally find certain penalty parameter in a bilinear form. Second, we applied algebraic multigrids methods to the algebraic system so that condition number is bounded regardless of the meshsize. In conclusion, we have reduced computation times to solve algebraic systems for PBE. We expect that algebraic multigrids methods can be further employed in various disciplinary to generate deep learning samples.

키워드 : 대수적 다중격자 방법, 포아존볼츠만 방정식, 균일 격자, 경계함유유한요소법

Keywords : Algebraic multigrid methods, Poisson boltzmann equation, Uniform grids, Immersed finite element method

Received 29 November 2021, Revised 8 December 2021, Accepted 13 December 2021

* Corresponding Author Gwanghyun Jo (E-mail: gwanghyun@kunsan.ac.kr, Tel: +82-63-469-4545)

Assistant Professor, Department of Mathematics, Kunsan National University, Guansan-si, 54150 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.2.181>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

포아즌 볼츠만 방정식 (Poisson-Boltzmann equation, PBE) 는 생물화학, 반도체공학, 콜로이드 화학 등의 분야에서 등장하는 주요한 방정식이다 [1-5]. PBE 풀기 위해서 저자들은 [6]에서 딥러닝을 활용한 기법을 소개하였다. [6]에서는 finite element method (FEM) 기반의 방법을 이용하여 PBE의 해를 생성하고, 이를 통하여 ResNet 기반 딥러닝 아키텍처를 훈련하였다. 모든 과정이 순조롭게 이루어졌지만, 계산 시간의 관점에서 유일한 bottleneck은 FEM을 이용하여 딥러닝을 훈련하기 위한 dataset을 만드는 과정이었다.

본 논문에서는 FEM을 통하여 PBE를 풀 때 걸리는 시간을 효율적으로 줄이기 위한 새로운 방안을 소개하려 한다. FEM을 통하여 PBE를 이산화 할 때 생기는 대수 방정식 ($Ax=b$) 은 symmetric positive definite (SPD) 형태이다 [4]. SPD 형태의 대수 방정식을 풀기 위해서 conjugate gradient (CG) 방법을 사용할 때 효율적인 preconditioner를 사용한다면 CG의 iteration 수가 줄어든다는 것이 알려져 있다. 이렇게 preconditioner를 사용한 conjugate gradient를 PCG라 하는데, 본 논문에서는 PCG를 통하여 대수 방정식을 풀 때 걸리는 시간을 줄이기 위한 두가지 방안을 설명한다.

첫 번째 방법은 이산화(discretization) 과정에서 bilinear form에 포함되는 penalty parameter를 조정하는 것이다. Penalty parameter가 너무 작을 경우 수치해의 convergence order가 떨어지는 현상이 보고된 반면, penalty parameter가 너무 큰 경우 system의 컨디션 넘버가 너무 커지게 된다. 따라서 실험적으로 적당한 penalty 파라미터를 찾았다.

두 번째 방법은 algebraic multigrid (AMG)를 사용하여 대수 방정식의 컨디션 넘버를 줄이는 것이다. AMG는 multigrid 방법의 변형 중 하나로, system을 residual을 대수적 coarse grid에 보내서 문제를 푼 이후에 원래 그리드에 보내는 방법이다 [7]. AMG를 통하여 타원형 방정식을 효율적으로 풀 수 있다는 결과가 알려져 있으나, PBE를 AMG를 통하여 푼 결과는 아직까지 찾기가 쉽지 않다. 따라서 AMG를 통하여 PBE를 푸는 결과를 보고하는 것은 의미가 있다. 결과 장에서는 AMG를 preconditioner로 사용하는 PCG를 통하여 PBE를 풀 경우 기존의 CG를 사용하는 것에 비해 계산 시간이 획기적으로 줄어드는 것을 보고한다. 이러한 대수적 멀티그리드를 사용한 방

법은 다양한 분야에서 딥러닝의 샘플을 생성하는데 효과적으로 활용될 수 있을 것으로 기대된다.

본 논문의 구성은 다음과 같다. 2.1 에서 PBE 방정식의 정규화 과정을 설명한다. 2.2에서 FEM의 방법중에 하나인 경계함유 유한요소법 (Immersed finite element method, IFEM)을 통하여 PBE를 이산화 하는 과정과 여기서 penalty parameter의 역할을 설명한다. 또한 AMG를 적용하는 과정을 포함하여 PBE를 수치적으로 푸는 전반적인 과정을 설명한다. 3장에서 AMG를 사용할 경우 CPU time이 줄어드는 것을 보고한다. 이어서 4장에서 결론을 요약한다.

II. 본 론

PBE를 빠른 속도를 푸는 것은 딥러닝에 필요한 샘플을 생성하는 측면에서 중요한 이슈이다. 이번 장에서 PBE를 IFEM을 기반으로 푸는 전반적인 과정을 설명한다. 대수방정식을 AMG를 통하여 풀어서 해석 시간을 효과적으로 줄일 수 있다.

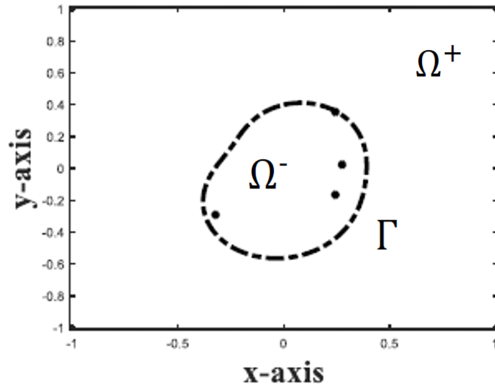


Fig. 1 Domain description. Molecular domain (Ω^-) is embedded in Ω . Solvent domain is $\Omega^+ = \Omega - \Omega^-$. Γ indicates the interface between the solvent and molecular domain. Black dots in Ω^- indicates the locations of charges.

2.1. Poisson-Boltzmann 방정식과 정규화 과정

Solvent 도메인 (Ω^+)과 molecular 도메인(Ω^-)을 갖고 있는 PBE 방정식을 기술하려 한다. 이 때 전체 도메인은 $\Omega = \Omega^+ \cup \Omega^-$ 이며, solvent-molecular의 경계는 Γ

로 표기한다 (그림 1 참고). 또한 경계와 수직인 방향인 크기가 1인 벡터를 n_r 로 표현하며, dielectric 상수 ϵ 는 Ω^+ 에서 ϵ^+ 이 되며, Ω^- 에서 ϵ^- 이 된다. PBE 방정식은 다음과 같이 기술 된다.

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla u) + \kappa^2 \sinh(u) &= \rho^f \text{ on } \Omega \quad (1a) \\ [u]_{\Gamma} &= 0 \text{ on } \Gamma, \quad (1a) \\ \left[\epsilon \frac{\partial u}{\partial n_{\Gamma}} \right]_{\Gamma} &= 0 \text{ on } \Gamma, \quad (1a) \\ u &= 0 \text{ on } \partial\Omega. \quad (1a) \end{aligned}$$

여기서 주요 변수인 u 는 전위를 나타내며, κ 는 solvent 도메인에서만 0이 아닌 상수이다. (2a)는 전위가 경계면에서 연속임을 뜻하며, (3a)는 전위의 미분 방향의 flux가 연속임을 뜻하며, (4a)는 전위가 도메인의 경계에서 0임을 뜻한다. (1a)의 우변은 N개의 charge의 합으로 $\rho^f = \sum_{i=1}^N q_i \delta(x - x_i)$ 로 계산되며, 이때 x_i 는 charge의 위치를, q_i 는 charge의 전하량을 뜻한다. 또한 $\delta(x)$ 는 direc delta 함수이다.

ρ^f 는 charge가 위치한 지역에서 무한이 됨으로, 보통 정규화 과정을 통하여 (1)에서 direc delta 형태의 singularity를 제거한다 [4]. 정규화 과정에서는 u 를 정규화 해 (u^r), singular 해(u^s)의 합으로 나타내며, u^s 는 다음과 같이 정의 한다.

$$\begin{aligned} u^s &= G \mid_{\Omega^-} \text{ on } \Omega^- \quad (2) \\ u^s &= 0 \text{ on } \Omega^+. \end{aligned}$$

이때 $G = -\sum_{i=1}^N \frac{q_i}{2\pi\epsilon^-} \log|x - x_i|$ 이며, 이와 같은 정의로 $-\epsilon \Delta u^s = \rho^f$ 가 된다. u^s 를 (1) 식에서 빼면, 정규화해 u^r 에 대한 방정식이 다음과 같이 얻어진다.

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla u^r) + \kappa^2 \sinh(u^r) &= 0 \text{ on } \Omega \quad (3a) \\ [u^r]_{\Gamma} &= J_1 \text{ on } \Gamma, \quad (3a) \\ \left[\epsilon \frac{\partial u^r}{\partial n_{\Gamma}} \right]_{\Gamma} &= J_2 \text{ on } \Gamma, \quad (3a) \\ u &= 0 \text{ on } \partial\Omega. \quad (3a) \end{aligned}$$

여기서 $J_1 = -u^s$, $J_2 = -\epsilon^- \frac{\partial u^s}{\partial n_{\Gamma}}$ 이다.

2.2. Immersed finie element method를 통한 PBE 해석

본 장에서는 IFEM을 통하여 정규화된 PBE (2번 수식)을 푸는 방법을 설명하려 한다. IFEM을 통하여 PBE를 풀게 되는 장점은 균일 격자 위에서 PBE를 풀 수 있다는 점이다. 따라서 fitted 그리드 격자에 의해서 만들어진 대수 방정식에 비해 IFEM을 통하여 만든 대수 방정식의 형태가 간단하다는 장점이 있다. 본 논문에서는 [4]에서 소개한 PBE를 풀기 위한 IFEM 기법을 사용할 것이며, 본 논문의 목표가 IFEM을 설명하는 것이 아니기 때문에 간단히 기술하려 한다.

먼저 도메인을 균일 삼각형으로 나눈 triangulation, T_h 을 사용한다. (3b), (3c)에서 확인할 수 있듯이 경계면에서 전위와 전위의 flux가 불연속이 된다. 이러한 불연속 점을 해결하기 위해서, u^r 을 연속한 해 \bar{u} 와 불연속한 해 u^* 으로 나눈다. u^* 는 IFEM에서 사용하는 discontinuous bubble 함수로 표현한다. u^* 는 요소(element)가 Γ 에 잘리게 되는 요소 위에서 다음을 만족하는 piecewise 선형 함수로 정의한다.

$$\begin{aligned} [u^*]_{\Gamma} &= J_1 \text{ on } \Gamma, \quad (4) \\ \left[\epsilon \frac{\partial u^*}{\partial n_{\Gamma}} \right]_{\Gamma} &= J_2 \text{ on } \Gamma. \end{aligned}$$

(4)을 통하여 u^* 을 구한 이후에, 연속한 해 $\bar{u} = u^r - u^*$ 는 다음을 만족하게 된다.

$$\begin{aligned} -\nabla \cdot (\epsilon \nabla \bar{u}) + \kappa^2 \sinh(\bar{u} + u^*) &= \nabla \cdot (\epsilon \nabla u^*) \text{ on } \Omega \quad (5a) \\ [\bar{u}]_{\Gamma} &= J_1 \text{ on } \Gamma, \quad (5a) \\ \left[\epsilon \frac{\partial \bar{u}}{\partial n_{\Gamma}} \right]_{\Gamma} &= J_2 \text{ on } \Gamma, \quad (5a) \\ \bar{u} &= 0 \text{ on } \partial\Omega. \quad (5a) \end{aligned}$$

\bar{u} 를 근사하기 위한 trail과 test 공간은 P1-conforming IFEM ($\hat{S}_h(\Omega)$)공간을 사용한다 [8]. 마지막으로 (5) 식을 풀기 위한 IFEM 방법을 기술하려 한다. \bar{u} 는 다음의 약 문제를 (weak problem) 풀므로 구해지게 된다: $\hat{S}_h(\Omega)$ 안의 모든 ϕ_h 에 대해서 다음을 만족하는 \bar{u} 을 찾아라,

$$\begin{aligned}
 & - \sum_{T \in T_h} \left(\int_T \epsilon \nabla \bar{u} \cdot \phi_h + \int_T \kappa^2 \sinh(\bar{u} + u^*) \phi_h \right) \quad (6) \\
 & + \sum_{e \in E_h} \left(- \int_e \epsilon \nabla \bar{u} \cdot n_e [\phi_h] ds - \int_e \epsilon \nabla \phi_h \cdot n_e [\bar{u}] ds \right. \\
 & \left. + \int_e \frac{\sigma}{|e|} [\bar{u}] [\phi_h] ds \right) = \int_{\Gamma} J_2 \phi_h ds \\
 & - \sum_{T \in T_h} \int_T \epsilon \nabla u^* \cdot \nabla \phi_h ds - \sum_{e \in E_h} \left(\int_e \epsilon \nabla u^* \cdot n_e [\phi_h] ds \right. \\
 & \left. + \int_e \epsilon \nabla \phi_h \cdot n_e [u^*] ds + \int_e \frac{\sigma}{|e|} [u^*] [\phi_h] ds \right).
 \end{aligned}$$

여기서 등장하는 $\sigma > 0$ 를 penalty parameter라 부른다. 결과 장에서 penalty parameter에 따른 컨디션 넘버를 보고 하려 한다. (6) 방정식을 풀기 위해서 발생하는 대수 방정식 ($Ax=b$)를 빠르게 풀기 위해서 AMG를 적용하려 한다. AMG를 적용할 때 계산 시간 차원의 이득을 결과장에서 기술하려 한다.

마지막으로 이와 같은 방법으로 PBE 솔루션을 구하는 과정을 요약하여 정리하면 다음과 같다. 먼저 (1) 식을 정규화 하기 위해서 (2) 수식을 따라 singular 해 u^* 를 구한다. 그 이후에 u^* 를 (1)식에서 제거한 이후에 나오는 정규화 된 수식 (3)에서 불연속 한 해 \bar{u}^* 를 구하는 과정을 (4) 수식을 통하여 구한다. 이후에 (5) 약문제를 풀으로써 \bar{u} 를 찾는다. 최종적으로, u^s, u^*, \bar{u} 를 구함으로 최종 수치해를 얻게 된다.

Algorithm 1. IFEM.

- 1) Construct u^s from (2)
 - 2) Construct u^* which satisfies (4)
 - 3) Solve (5) for \bar{u} .
 - 4) Compute $u_h := \bar{u} + u^* + u^s$
-

이때 대부분이 소요되는 과정은 Algorithm 1에서 3)에 해당한다. 결과 장에서 3) 과정에서 등장하는 대수 방정식을 효율적으로 푸는 방법을 제시할 것이다.

그림 2는 Algorithm 1. IFEM 을 통하여 PBE의 수치해를 구한 결과를 보여 준다. 이 때 domain의 정보는 그림 1에서 나타난 그대로 하였으며, (1)식에서 상수들은 다음과 같이 정하였다. $\epsilon^- = 2, \epsilon^+ = 80, \kappa^2 = 8.847$. 또한 도메인은 $\Omega = [-1, 1]^2$ 로 하였으며, node의 수는 16,641개를 사용하였다. 상수들의 값과 도메인의 크기

가 대수 방정식에서 컨디션 넘버에 영향을 끼치는 경향이 크지 않으므로, 논문에 나머지 부분들에서도 동일한 상수값과 도메인을 사용하려 한다.

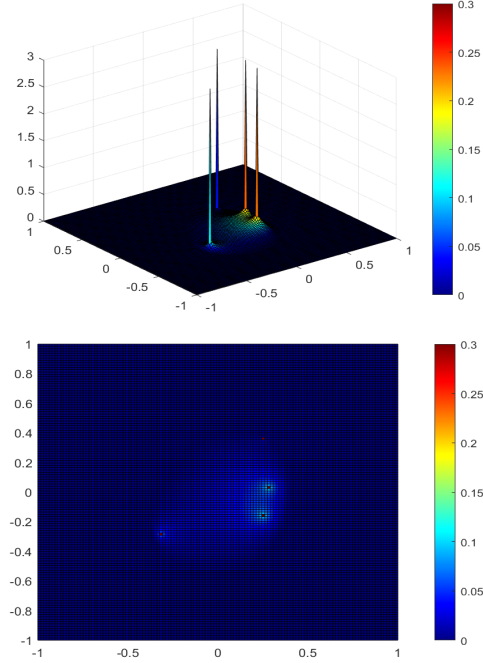


Fig. 2 Numerical solutions for PBE obtained by Algorithm 1. IFEM. Above figure shows the result looking obliquely at some angle. Bottom figure shows result in top-down view.

III. PBE 수치 실험

본 장에서 (5) 수식을 풀기 위해서 생기는 대수 방정식 ($Ax=b$)를 풀기 효율적으로 풀기 위한 두가지 측면의 방법을 설명하려 한다. 첫 번째는 (5)식의 penalty parameter를 실험적으로 바꾸며, A의 컨디션 넘버가 작아지는 값을 찾는 것이다 (section 3.1 참고). 두 번째는 AMG를 사용한 PCG를 이용한 것이다 (section 3.2 참고). AMG를 사용할 경우 A의 컨디션 넘버가 어떻게 작아지는 지를 기술하였다. 모든 경우에서 컨디션 넘버는 다음과 같이 계산 되었다.

$$\kappa(A) = \frac{\text{largest eigenvalue of } A}{\text{smalles eigenvalue of } A} = \frac{\lambda_{\max}}{\lambda_{\min}}$$

3.1. Penalty parameter에 따른 A의 컨디션 넘버 변화

본 장에서는 (6) 수식의 penalty parameter와 대수 방정식의 컨디션 넘버의 관계를 실험적으로 찾는다. 표 1은 (5)식에서 penalty parameter를 10배씩 증가시키면서 A의 컨디션 넘버를 보고하고 있다. Penalty parameter가 1일 때 값이 가장 작음을 확인 할 수 있다. penalty parameter가 0일 때 컨디션 넘버가 상대적으로 커짐으로, 단순히 penalty parameter가 작을 때 컨디션 넘버가 작은 것이 아니라는 것을 알 수 있다. 반면 penalty parameter가 100이 넘어가면 컨디션 넘버가 급격히 커짐을 확인할 수 있다.

표 2에서는 penalty parameter를 0과 2 사이에서 조정하면서 컨디션 넘버를 확인하면서 최적의 penalty parameter를 찾는 과정을 보여 준다. penalty parameter가 1 일 때 컨디션 넘버가 최적으로 작아지는 것을 확인할 수 있다. 따라서 penalty parameter를 1로 하는 것이 최적의 선택임을 확인할 수 있다.

Table. 1 Condition number of A with penalty parameter 0, 10, 100, 1000, 10000.

Penalty parameter	condition number
0	4.210E4
1	3.975E4
10	4.201E4
100	4.651E4
1000	2.477E5
10000	2.155E6

Table. 2 Condition number of A with penalty parameter in rage [0,2].

Penalty parameter	condition number
0.25	4.278E4
0.5	4.162E4
0.75	4.213E4
1	3.975E4
1.25	4.150E4
1.5	4.233E4
2	4.056E4

3.2. AMG-PCG를 통한 계산 시간

본 장에서는 AMG를 preconditioner로 활용함으로써 시스템의 컨디션 넘버를 줄이고, PCG 계산 시간을 획기적

으로 줄일 수 있음을 보여준다. Penalty parameter는 1로 고정하였다. 표 3에서 CG와 AMG를 사용한 PCG (편의상 AMG-PCG 로 표기)를 사용하여 대수방정식을 풀 결과를 CPU time과 iteration 측면에서 비교하였다. CG와 PCG 반복의 stopping criteria는 다음과 같이 하였다.

$$\frac{\|Ax - b\|}{\|b\|} < 10^{-10}.$$

CG의 경우 반복을 1379번 하였지만 AMG-PCG의 경우 7번 만에 수렴함을 확인할 수 있다. 또한 CPU 타임의 경우 CG의 경우 3.14초, AMG-PCG의 경우 1.87초로 41% 정도 감소함을 확인할 수 있다.

Table. 3 Condition number of A with penalty parameter in rage [0,2].

CG	AMG-PCG
CPU time: 3.14s Iteration: 1379	CPU time: 1.87s Iteration: 7

마지막으로 AMG를 적용할 경우 system의 condition number가 얼마나 줄어들게 되는지를 보고 하려 한다. 표 4는 AMG를 system에 적용할 경우 condition number가 얼마나 줄어드는 지를 보고 한다. AMG를 적용하기 전 시스템의 컨디션 넘버는 3.975E4인 반면, AMG를 적용한 후 시스템의 컨디션 넘버는 1.002로 줄어들음을 확인할 수 있다. 따라서, AMG-PCG 반복 수가 극적으로 줄어드는 이유가 1379에서 7번으로 줄어드는 이유를 알 수 있다.

Table. 4 Condition number of A and condition number of BA where B indicates AMG preconditioner.

A	BA
$\lambda_{\max}: 1.57E-2$	$\lambda_{\max}: 9.99E-1$
$\lambda_{\min}: 6.48E2$	$\lambda_{\min}: 9.97E-1$
$\kappa(A): 3.975E4$	$\kappa(A): 1.002$

IV. 결론

PBE 는 생물과학, 콜로이드 과학에서 등장하는 주요한 방정식이다. [6]에서 저자들은 딤러닝을 활용한 PBE의 풀이법을 발표하였으나 유일한 bottleneck은 FEM을 통하여 샘플을 생성하는데 걸리는 시간이 크다는 것이

었다. 본 논문에서는 두가지 방법으로 PBE를 빠르게 푸는 대안을 제시 하였다. 첫 번째는 약문제 (weak problem)에서 등장하는 penalty parameter를 조정하는 것이었다. 실험을 통하여, 시스템의 컨디션 넘버를 가장 줄일 수 있는 penalty parameter가 1 근처임을 확인 하였다. 두 번째 방법은 AMG을 사용한 PCG 방법을 사용하는 것이다. AMG를 사용할 경우 시스템의 컨디션 넘버가 1근처로 크게 줄기 때문에 PCG 반복수가 획기적으로 줄는 것을 확인하였다.

본 논문은 PBE를 풀기 위한 샘플을 빠르게 생성하는 방법을 제안하였다는데 의미가 있다. 따라서 [6]에서 제안한 딥러닝 아키텍처를 ion transportation 해석 등의 분야에서 효과적으로 확장할 수 있을 것으로 기대한다.

ACKNOWLEDGEMENT

The first author is supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NO. 2020R1C1C1A01005396) and second author is supported by NRF grant funded by MSIT (NO. NRF-2019R1G1A1087290).

REFERENCES

[1] G. Borlesk and Y. C. Zhou, "Enriched gradient recovery for interface solutions of the Poisson-Boltzmann equation," *Journal of Computational Physics*, vol. 421, Article ID: 109725, Nov. 2020.

[2] V. Ramm, J. H. Chaudry, and C. D. Cooper, "Efficient mesh refinement for the Poisson-Boltzmann equation with boundary elements," *Journal of Computational Chemistry*, vol. 42, no. 12, pp. 855-869, Mar. 2021.

[3] S. Wang, E. Alexov, and S. Zhao, "On regularization of charge singularities in solving the Poisson-Boltzmann equation with a smooth solute-solvent boundary," *Mathematical Biosciences and Engineering*, vol. 18, no. 2, pp. 1370-1405, Jun. 2021.

[4] I. Kwon and D. Y. Kwak, "Discontinuous bubble immersed finite element method for Poisson-Boltzmann equation," *Communications in Computational Physics*, vol. 25, no. 3, pp. 928-946, Aug. 2019.

[5] I. Kwon, D. Y. Kwak, and G. Jo, "Discontinuous bubble immersed finite element method for Poisson-Boltzmann-Nernst-Planck model," *Journal of Computational Physics*, vol. 438, Article ID: 110370, Aug. 2021.

[6] K. In, G. Jo, and K. S. Shin, "A deep neural network based on ResNet for predicting solutions of Poisson-Boltzmann equation," *Electronics*, vol. 10, no. 21, pp. 2627, Oct. 2021.

[7] J. Xu and L. Zikatanov, "Algebraic multigrid methods," *Acta numerica*, vol. 26, pp. 591-721, May. 2017.

[8] S. H. Chou, D. Y. Kwak, and K. T. Wee, "Optimal convergence analysis of an immersed interface finite element method," *Advances in Computational Mathematics*, vol. 33, no. 2, pp. 149-168, Mar. 2009.



신광성(Kwang-Seong Shin)

2005.3 전북대학교 컴퓨터공학과 석사
2014.3 군산대학교 컴퓨터정보공학과 박사
2008.8-2018.2 군산대학교 컴퓨터정보공학과
 겸임교수
2018.3-현재 원광대학교 디지털콘텐츠공학과
 부교수
※관심분야: 컴퓨터비전, 딥러닝, VR/AR



조광현(Gwanghyun Jo)

2011. 카이스트 수리과학과 학사
2013. 카이스트 수리과학과 석사
2018. 카이스트 수리과학과 박사
2018-2019. 카이스트 연구원(포스트닥터)
2019-현재: 군산대학교 수학과 조교수
※관심분야: 수치해석학, 딥러닝