# 안테나 특성 고속 계산을 위한 병렬화 행렬 연산

## Parallelized Matrix Operation for Fast Computations of Antenna Characteristics

조 용 희
목원대학교 정보통신융합공학부

Cho, Yong-Heui
Mokwon University

요약

밀리미터파 대역에서 사용하는 대형 안테나 해석 속도를 개선하기 위한 병렬형 행렬 연산법을 제안한다. 기존의 가우스 소거법을 병렬화하기 위해 행렬 분해와 반복법을 이용한다. 또한, 반복법의 수렴성을 높이기 위해 이전 행렬해를 부분적으로 사용하여 분해 행렬을 구성하는 방식도 제시한다. 본 제안법은 OpenMP, MPI, CUDA 등의 병렬법과 함께 사용할 수 있다.

## Ⅰ. 서론

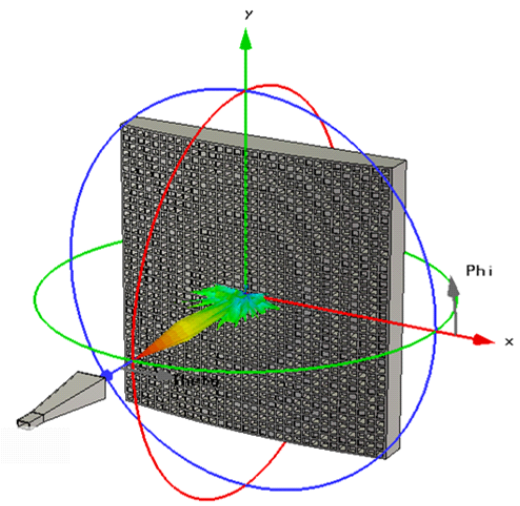
30에서 300 [GHz] 주파수 대역은 파장의 단위가 밀리미터이므로 밀리미터파(Millimeter Wave) 대역이라 부른다. 밀리미터파 대역은 파장이 밀리미터 단위이므로, 안테나 구동 주파수의 파장이 매우 짧아져서 전자파의 도달 범위가 줄어드는 문제가 있다. 이러한 밀리미터파 대역의 단점을 해결하기 위해 통신용 안테나는 이득이 매우 큰 대형 안테나를 사용한다[1].

밀리미터파 대역의 대형 안테나는 파장에 비해 물리적 크기가 매우 크기 때문에 안테나 해석과 설계에 막대한 시간이 소요된다. 현재 인기를 얻고 있는 병렬화 기법[2]-[4]을 사용하면 안테나 계산 시간을 줄일 수 있지만, 안테나 해석에 사용하는 가우스 소거법으로 인해 병렬화의 장점이 상쇄되는 문제가 있다. 본 발표에서는 가우스 소거법을 병렬화하기 위한 행렬 분해와 반복법에 기반을 둔 병렬화 행렬 연산법을 제안한다.

## Ⅱ. 병렬화 행렬 연산법

현재 많이 사용하는 병렬화 기법은 OpenMP(Open Multiple-Processing), MPI(Message Passing Interface), CUDA(Compute Unified Device Architecture)[2]-[4] 등이 있다. 병렬화 기법의 기본 원리는 다수 개의 계산 코어(Compute Core)가 할당된 업무를 분해하여 동시에 계산이 이루어지도록 하는 것이다. 계산 코어는 동일한 CPU, GPU, 계산 노드(Compute Node) 내부에 각각 위치할 수 있다. 계산 코어가 위치한 장소에 따라 병렬화 기법은 OpenMP, MPI, CUDA 등으로 불린다.

그림 1과 같은 안테나는 파장에 비해 물리적 크기가 매우 크기 때문에 밀리미터파 대역 초대형 안테나라고 한다. 파장에 비해 큰 안테나는 수치 해석을 위한 메쉬 개수가 매우 커지기 때문에 계산량이 안테나 크기에 따라 지수적으로 매우 빠르게 증가한다. 대형 안테나 수치 계산을 위해서는 CPU 계산 능력과 메모리 크기가 향상되어야 한다. 현재 대부분의 CPU와 운영체제는 64비트이므로 메모리 크기 향상은 어려움이 없다. 문제는 제한된 CPU의 계산 능력이다. CPU의 멀티코어 개수 한계로 인해 발생하는 문제점은 다양한 병렬화 기법을 통해 비약적으로 향상될 수 있다.




▶▶ 그림 1. 밀리미터파 대역용 고이득 안테나

그림 1과 같은 대형 안테나의 복사 특성은 전자파 해석을 통해 식 (1)의 행렬 방정식 형태로 표현할 수 있다.

$$MP = S \tag{1}$$

여기서 $M$은 안테나를 구성하는 매질의 특성을 담고 있는 행렬, $S$는 안테나의 전원을 표현하는 행렬, $P$는 안테나의 복사 특성을 결정하는 모드 계수 행렬이다. 안테나의 물리적 크기가 작은 경우는 초보적인 가우스 소거법을 이용하여 식 (1)의 해인 행렬 $P$를 쉽게 결정할 수 있다. 하지만, 그림 1의 구조처럼 안테나의 물리적 크기가 매우 크다면 가우스 소거법은 식 (1)을 풀기 위한 적절한 방법이 아니다. 왜냐하면 가우스 소거법은 행이나 열 연산을 순차적으로 진행하는 알고리즘이므로 병렬화 기법을 적용하더라도 계산 가속화가 매우 미미하게 일어나기 때문이다.

이러한 문제점을 해결하기 위해 행렬 분해와 반복법에 기반을 둔 아래와 같은 병렬화 행렬 연산법을 제안한다.

- 행렬해의 초기값은 $P^{(0)}$이라 표기하고 모두 0으로 초기화한다.
- 반복이 $n$번 진행되면 $P^{(0)}$은 $P^{(n)}$으로 바뀐다.
- 원래 행렬 $M,S,P^{(n)}$을 작은 행렬인 $m,s,p^{(n)}$으로 분해한다.
- 분해 행렬 간의 연관성을 고려하기 위해 $P^{(n-1)}$을 이용하여 $m,s,p^{(n)}$으로 구성된 행렬 방정식을 보정한다.
- 작은 행렬인 $m,s,p^{(n)}$을 가우스 소거법으로 풀어서 $p^{(n)}$을 결정한다.
- 이 과정을 계속 반복하여 원래 행렬해인 $P^{(n)}$이 수렴하도록 한다.

위의 알고리즘은 전체 행렬을 계산하지 않고 작은 행렬인 $p^{(n)}$에 대해서만 가우스 소거법을 적용하므로, 행렬 $P$의 크기가 커지더라도 병렬화를 할 수 있다. 다만 기본적으로 반복법에 기반을 두고 있으므로, 병렬화 행렬 연산법의 매개 변수를 조정하여 $p^{(n)}$이 발산하지 않도록 해야 한다.

## ■ 감사의 글 ■

본 저자는 밀리미터파 대역 고이득 안테나 구조의 CAD를 공유해준 국방과학연구소 이민우 박사에게 깊이 감사합니다.

## ■ 참 고 문 헌 ■

[1] Y. H. Cho, W. J. Byun, and M. S. Song, "High gain metal-only reflectarray antenna composed of multiple rectangular grooves," *IEEE Trans. Antennas Propag.*, vol. 59, no. 12, pp. 4559-4568, Dec. 2011.

[2] B. Chapman, G. Jost, and R. van der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming,* The MIT Press, 2007.

[3] 이홍석 , 김정한, 이승우, 이식, 멀티코어 시대에 꼭 알아야할 MPI 병렬 프로그래밍, 어드북스, 2010.

[4] 정영훈, *CUDA 병렬 프로그래밍 - 고성능 GPGPU를 이용한 NVIDIA 병렬 컴퓨팅 아키텍처,* 프리렉, 2011.