

2021 인공지능 소수전공

21차시: 수치 미분

2021.07.26 18:30~22:15

- 수치해석(Numerical Analysis)이란?
 - 어떤 함수나 방정식의 해를 컴퓨터를 이용해 수치적으로 근사해서 구하는 알고리즘에 대한 연구를 하는 학문
- 수치해석을 사용하는 이유
 - 사람의 손으로는 풀지 못하는, 또는 풀기 어려운 문제를 해결하기 위해서.

- 주어진(특히 수학적인) 문제를 풀기 위한(해를 찾기 위한) 방법
 - 해석적 방법 (Analytic Method)
 - 추상적인 이론을 바탕으로 논리적인 추론을 통해 문제를 풀어나가는 방법
 - 수학과에서 주로 다루며 학교에서 주로 해왔던 수학 문제 풀기 등이 해석적 방법에 해당
 - 수치적 방법 (Numerical Method)
 - 근사값을 이용해 해를 구하는 방법 및 알고리즘
 - 해석적 방법으로 직접 풀기 힘든 복잡하고 어려운 수학/공학문제를 풀기 위해 등장
 - 수학의 여러 분야에서 사용하는 방법과 결과를 이용하여 단계적인 산술연산(+, -, \times , \div)으로 수식화 하고, 미리 설정한 허용오차 범위 내에서 만족하는 근사값을 구하는 방법

- 해석적 방법과 수치적 방법의 비교 예시

- 주어진 문제: $x(t) = t^2, x'(2) = ?$

- 해석적 방법: x 함수의 도함수를 구한 뒤 도함수에 2라는 값을 넣어서 해결

- 수치적 방법: $\frac{(2+\delta)^2 - (2-\delta)^2}{2\delta}, \delta = 10^{-10}$ 과 같이 쉽게 구할 수 있는 근사값을 이용하여 계산

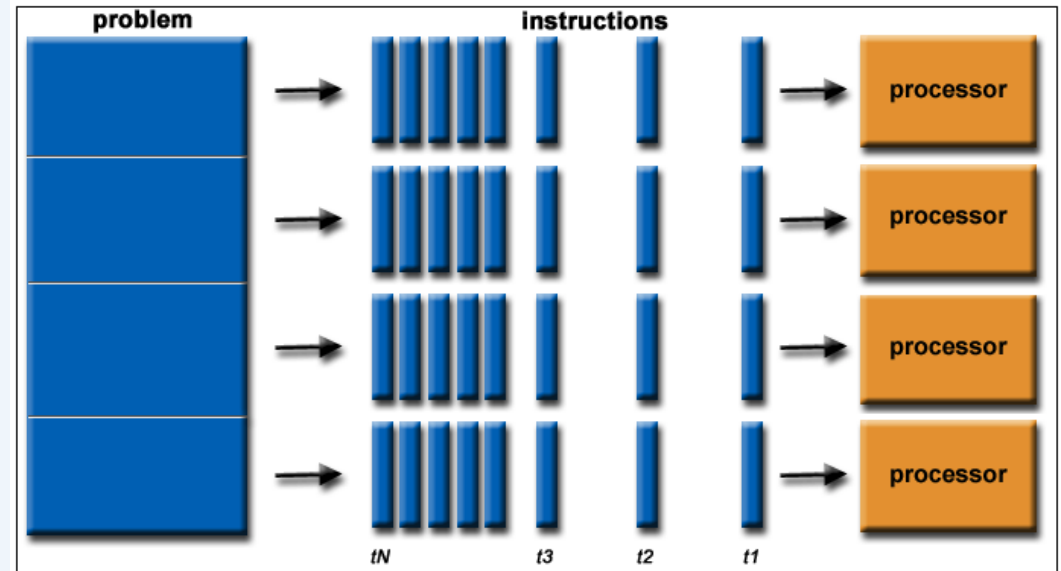
→ 참값과 완벽하게 같지는 않지만 거의 정확하게 구할 수 있음

- 왜 서로 다른 방법을 사용하는가?
 - 수학에서는 주로 어느 방정식의 **해가 존재한다는 것(해의 존재성과 유일성)**을 **증명하는 것이 목적**
- 공학에서는 방정식에 존재하는 **해의 값을 구하는 것이 주된 목적**
 - 쉽게 구할 수 있는 간단한 계산식들을 조합, 반복하여 해의 값에 매우 가까운 근사값을 계산 → 컴퓨터에게 일을 시키기 좋은 형태로 구성

- 수치해석 기술이 가장 잘 어울리는 시스템 → 병렬 컴퓨팅 시스템

- 병렬 컴퓨팅

- 하나의 문제를 해결하기 위한 계산을 여러 개로 나누어 각각을 다수의 처리 장치(프로세서)에 맡기는 계산 방식
- 프로그램의 계산량이 클 때, 프로그램의 실행 시간을 줄이기 위해 이용
- 현재의 거의 모든 슈퍼컴퓨터는 병렬 컴퓨팅 방식을 기본 구조로 채택 중



[사진 출처] ©OmniSci

- 병렬 컴퓨팅의 대표적인 이용 분야

- 수치해석

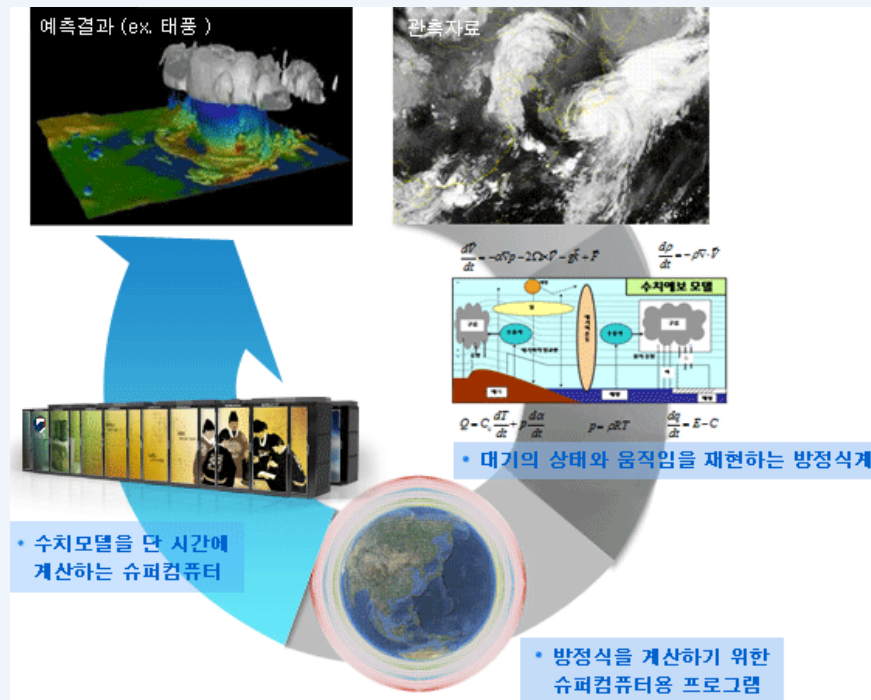
- 대표 사례: 기상 정보 예측, 유체 역학 등
 - 추진 중인 사례: 곤충의 날갯짓을 분석하여 모사한 새로운 비행체 개발
→ 전 세계적으로 연구, 개발이 시도 중이나 기술적 한계로 아직 미완성 상태

- 인공지능

- 특히 신경망, 딥러닝 모델의 경우 엄청난 양의 연산이 요구됨
 - 병렬 연산에 특화된 GPU의 등장으로 인해 본격적으로 성과를 내기 시작함

• 기상 정보 예측의 경우

- 하나의 답이 존재하는 수학 문제 풀이와 달리, 거대한 데이터를 이용한 유한 차분의 수치해석적 풀이에 기반한 수치예보 모델 사용

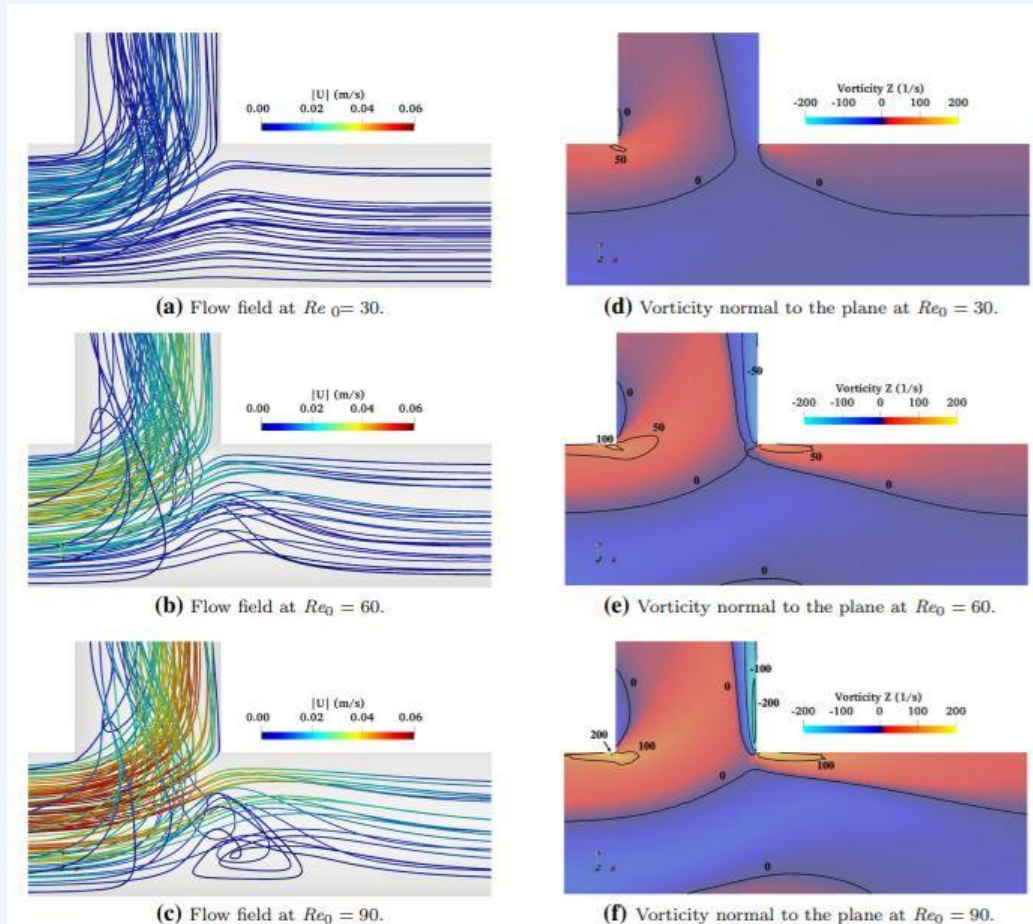


연속적인 대기 데이터를 일정한 간격으로 잘라 큐브 형태로 만들고 이를 토대로 방정식을 풀어 나감

[사진출처: 기상청 수치예보모델, 기상청]

[자료출처: “병렬 컴퓨팅, 컴퓨터 속도에 혁신을 불러오다”, 과학기술정보통신부 블로그, 2021.07.21]

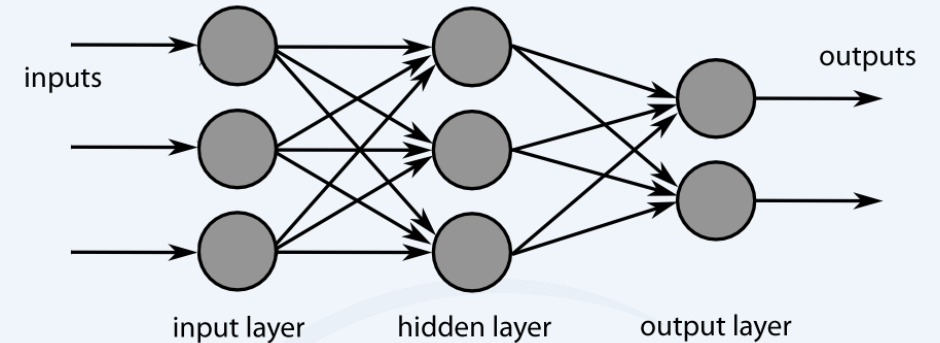
• 유체역학 분석의 예



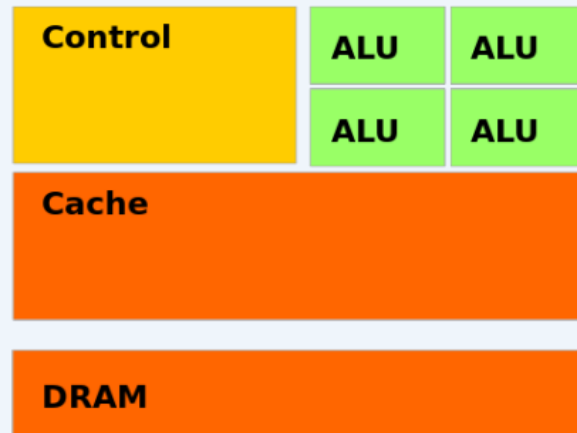
심혈관 내부 혈액의 유동을 수치 해석적으로 분석한 결과

[자료출처: “병렬 컴퓨팅, 컴퓨터 속도에 혁신을 불러오다”, 과학기술정보통신부 블로그, 2021.07.21]

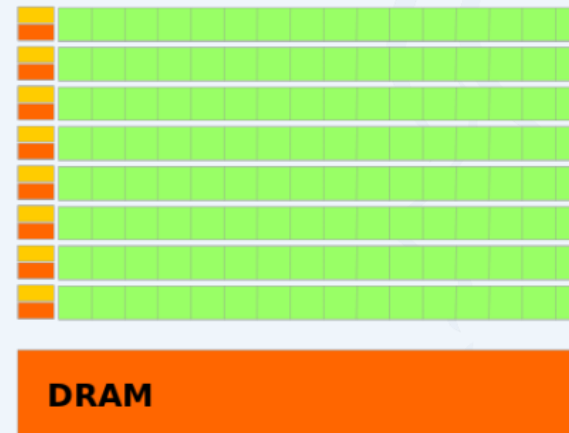
- 인공지능, 특히 딥러닝 기술의 경우
 - 뇌의 각 뉴런의 역할을 담당하는 각각의 노드에 대한 엄청난 양의 연산 요구
 - GPU 등장 전까지는 거의 개념만 존재



딥러닝 아키텍처 [사진출처: ©Wikimedia commons]



CPU

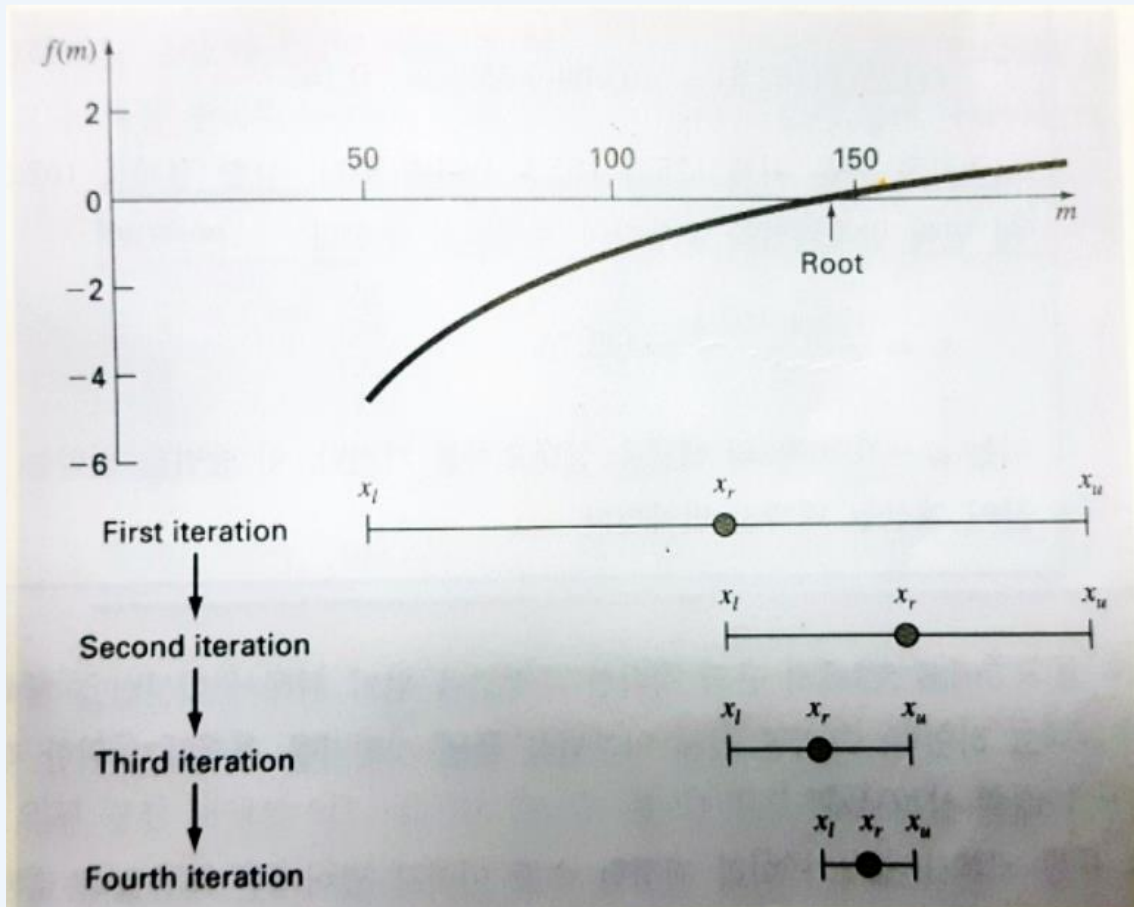


GPU

CPU와 GPU 차이 [사진출처: ©Wikimedia commons]

- 프로그래밍 대회에서 유용하게 사용되는 수치해석의 대표적인 기법
 - 이분법
 - 말 그대로 이등분으로 분해하는 방법을 말함
 - 함수의 근을 모를 때, 근이 있다고 판별할 수 있는 근거는 함수 값의 2개의 곱이다.
→ “함수 값 2개를 구해서 그것의 곱이 0보다 작다면 그 사이에 적어도 근이 한 개 이상 존재한다”라는 개념을 이용함
 - 수치해석이라는 개념 자체가 근을 모르는 상태에서 근을 찾아가는 방식을 말하기 때문에 수치해석은 기본적으로 “반복법”에 기초하여 근을 찾음
 - 근이 있을 것 같은 위치를 계속적으로 2개의 구간으로 나누어 가면서 위치를 찾는 방법

• 이분법



$$f(x_{lower}) \times f(x_{upper}) < 0$$

이므로 두 점 사이에는 근이 하나 이상 존재함

주어진 함수의 영역을 반복적으로 2등분하여
근이 있을 위치를 좁혀 나감

특이한 함수의 경우에는 근을 찾지 못하는 경
우도 있으나 예제처럼 연속이며 부드러운 곡
선에서는 일반적으로 근을 찾는데 유용함

- 수치해석적 접근

- 이분법의 예시에서 본 것처럼 컴퓨팅 알고리즘은 수학적인 공식과는 접근 방법이 다름
- 정통 수학에서는 수치해석을 수학이라고 인정하지 않으며, 이런 이유로 수치해석은 컴퓨터 공학 등의 영역으로 분류되기도 함
- 인공지능과 같은 수많은 단순 연산을 요구하는 분야의 경우 수치해석을 기반으로 한 접근이 필요하며, 알고리즘의 구성도 기존 수학과는 다른 형태일 가능성이 높음을 인식할 필요가 있음