Solving Linear Equation $Ax = b$:

Techniques and Examples

Sungwoo Jeong

Department of Mathematics, MIT

Solving Scientific Problem

Steps of numerical solving

- Formulation
- Selecting right discretization algorithm(Finite Difference, Finite Element, …)
- Solving Linear Equation $Ax = b$

Solving Linear Equation Ax = b

- In most of large-scaled problems, solving linear equation takes majority of computation time
- PROBLEM : Can we solve them faster?
- Software solution : Numerical Algorithms
- Hardware solution : GPU, Multiprocessor, …
  - WITH appropriate numerical linear algebra

Oil Reservoir Simulation

- Maximizing total production by optimizing production and oil recovery strategies
- Understanding behavior of fluid and reservoir under various production/well plans
- Simulate multiphase flow through porous media

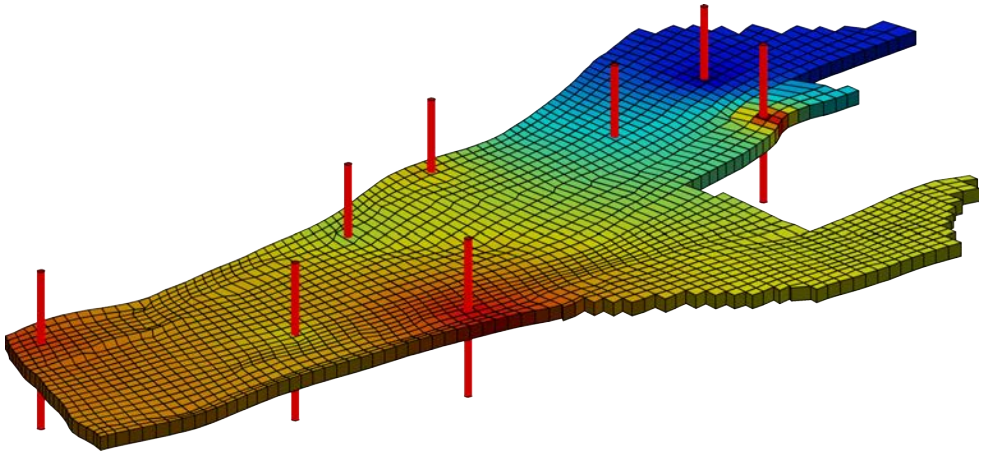

Figure 1. Reservoir Simulation overview

A. Hasan, *Optimization and Control of Petroleum Reservoirs,* Doctoral thesis, NTNU, 2013

Oil Reservoir Simulation

- Extremely heterogeneous nature of the earth
- Resulting linear systems are both huge and ill-conditioned
  - 80-90% of simulation time is used for Linear Solve
- 2 Solutions
  - Software Solution : Preconditioning
  - Hardware Solution : GPU computing

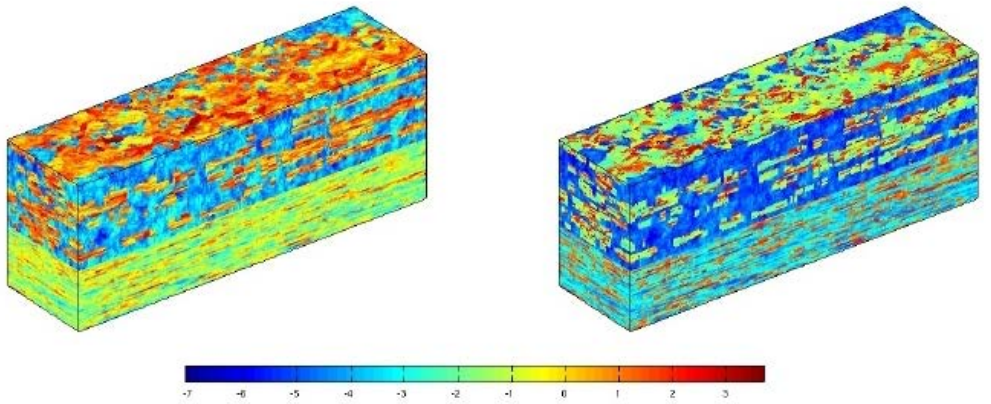

Figure 2. Heterogeneous Horizontal/Vertical Permeability field of oil reservoir

Source: https://www.sintef.no/projectweb/geoscale/results/msmfem/spe10

Software Solution : Preconditioning

- Solving Ax = b with A having large condition number

- So we propose a new equation, equivalent to $Ax = b$

- We will now solve $Ax = b$ with following steps
  1. $APy = b$ (The matrix AP should have better condition number than A)
  2. $Py = x$

Software Solution : Preconditioning

$$Ax = b \Leftrightarrow \begin{cases} APy = b \\ Py = x \end{cases}$$

- Why are we doing this? What is good preconditioning?
  1. AP should have (relatively) small condition number
  2. Multiplying P (solving second equation) should be cheap

- Simple simulation with Julia! (Live notebook)

Hardware Solution : GPU computing

- What is GPU? A hardware that has thousands of small CPUs inside
  - They are very simple CPUs which can only do simple operations (+, -, *, /…)
- So instead of making our main CPU to work on 1000 jobs serially, what if we can tell GPU to do 1000 jobs simultaneously, with its small CPUs?

Hardware Solution : GPU computing

- Simple example : Diagonal Matrix multiplied by Diagonal Matrix
  - Just 1000 multiplications

- CPU : Need to do 1000 operations
- GPU : Need only 1 operation each

Hardware Solution : GPU computing

- A famous specialty of GPU : Matrix multiplication
- Good algorithm for solving $Ax = b$ which is good for GPU : Krylov Method
- Krylov Methods : Solving Linear Equation by series of matrix multiplications
  - GMRES
  - CG
  - BiCGStab
  - Many more

Take home lesson

- Solving $Ax = b$ is another serious problem
  - not just an easy linear algebra homework problem!

- Software solutions are choosing right algorithm : ex) Preconditioning

- Hardware solutions are using good machines with appropriate numerical algorithms : ex) GPU with Krylov methods

18.085 Computational Science and Engineering I
Summer 2020