18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Random Walk
  - 3-D heat equation
  - Random walk game
  - Randomized PDE solver

3-D Heat Equation

Density

Laplace operator

$$\rho \cdot C_p \cdot \frac{\partial T(x,y,z,t)}{\partial t} = \kappa \cdot \nabla^2 T(x,y,z,t) + f(x,y,z,t)$$

Thermal capacity

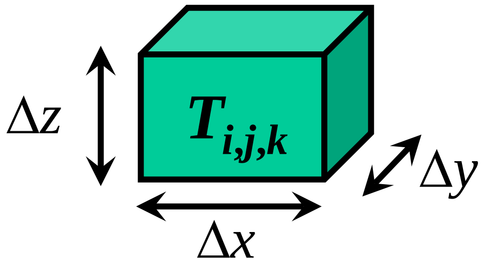
Thermal conductivity

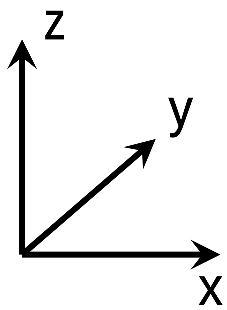
Heat source

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

Finite Difference

- A control volume





- Discretize PDE at each control volume

$$I_{i,j,k} = C \cdot \frac{\partial T_{i,j,k}}{\partial t} + G_x \cdot \left[T_{i,j,k} - T_{i+1,j,k}\right] + G_x \cdot \left[T_{i,j,k} - T_{i-1,j,k}\right]$$
$$G_y \cdot \left[T_{i,j,k} - T_{i,j+1,k}\right] + G_y \cdot \left[T_{i,j,k} - T_{i,j-1,k}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k+1}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k-1}\right]$$

$$G_x = \frac{\kappa \cdot \Delta y \cdot \Delta z}{\Delta x} \quad G_y = \frac{\kappa \cdot \Delta x \cdot \Delta z}{\Delta y} \quad G_z = \frac{\kappa \cdot \Delta x \cdot \Delta y}{\Delta z}$$

$$C = \rho \cdot C_p \cdot \Delta x \cdot \Delta y \cdot \Delta z \quad I_{i,j,k} = f_{i,j,k} \cdot \Delta x \cdot \Delta y \cdot \Delta z$$

Steady-State Solution

- Generally interested only in steady state – thermal capacitance is not considered

$$I_{i,j,k} = C \cdot \frac{\partial T_{i,j,k}}{\partial t} + G_x \cdot \left[T_{i,j,k} - T_{i+1,j,k}\right] + G_x \cdot \left[T_{i,j,k} - T_{i-1,j,k}\right]$$

$$G_y \cdot \left[T_{i,j,k} - T_{i,j+1,k}\right] + G_y \cdot \left[T_{i,j,k} - T_{i,j-1,k}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k+1}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k-1}\right]$$

$$C \cdot \frac{\partial T_{i,j,k}}{\partial t} = 0$$

$$I_{i,j,k} = G_x \cdot \left[T_{i,j,k} - T_{i+1,j,k}\right] + G_x \cdot \left[T_{i,j,k} - T_{i-1,j,k}\right]$$

$$G_y \cdot \left[T_{i,j,k} - T_{i,j+1,k}\right] + G_y \cdot \left[T_{i,j,k} - T_{i,j-1,k}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k+1}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k-1}\right]$$

Steady-State Solution

- Result in a set of linear equations

$$I_{i,j,k} = G_x \cdot \left[T_{i,j,k} - T_{i+1,j,k}\right] + G_x \cdot \left[T_{i,j,k} - T_{i-1,j,k}\right]$$
$$G_y \cdot \left[T_{i,j,k} - T_{i,j+1,k}\right] + G_y \cdot \left[T_{i,j,k} - T_{i,j-1,k}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k+1}\right] + G_z \cdot \left[T_{i,j,k} - T_{i,j,k-1}\right]$$

$$\Downarrow$$

$$I_{i,j,k} = 2 \cdot \left(G_x + G_y + G_z\right) \cdot T_{i,j,k} - G_x \cdot \left(T_{i+1,j,k} + T_{i-1,j,k}\right)$$
$$- G_y \cdot \left[T_{i,j+1,k} + T_{i,j-1,k}\right] - G_z \cdot \left[T_{i,j,k+1} + T_{i,j,k-1}\right]$$

Steady-State Solution

$$I_{i,j,k} = 2 \cdot \left(G_x + G_y + G_z\right) \cdot T_{i,j,k} - G_x \cdot \left(T_{i+1,j,k} + T_{i-1,j,k}\right) - G_y \cdot \left[T_{i,j+1,k} + T_{i,j-1,k}\right] - G_z \cdot \left[T_{i,j,k+1} + T_{i,j,k-1}\right]$$

$$\Downarrow$$

$$\begin{aligned} T_{i,j,k} = & \frac{G_x}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i+1,j,k} + \frac{G_x}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i-1,j,k} \\ & + \frac{G_y}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i,j+1,k} + \frac{G_y}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i,j-1,k} \\ & + \frac{G_z}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i,j,k+1} + \frac{G_z}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot T_{i,j,k-1} \\ & + \frac{1}{2 \cdot \left(G_x + G_y + G_z\right)} \cdot I_{i,j,k} \end{aligned}$$

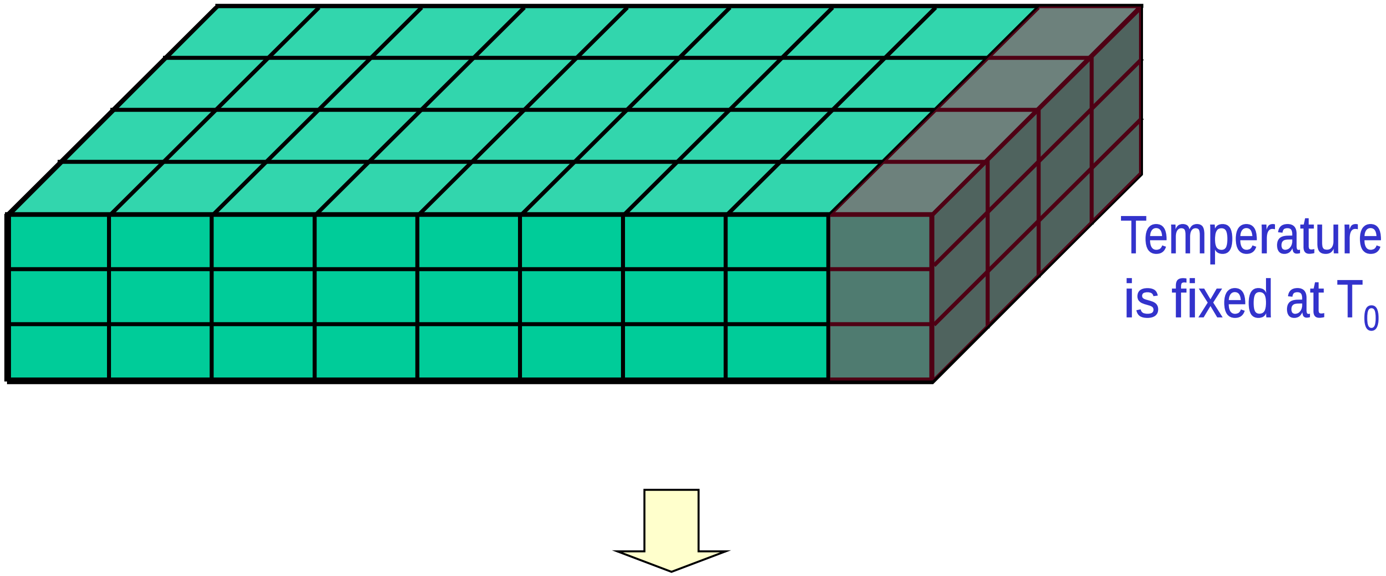
Steady-State Solution

$$T_{i,j,k} = \frac{G_x}{2\cdot(G_x+G_y+G_z)}\cdot T_{i+1,j,k} + \frac{G_x}{2\cdot(G_x+G_y+G_z)}\cdot T_{i-1,j,k}$$
$$+ \frac{G_y}{2\cdot(G_x+G_y+G_z)}\cdot T_{i,j+1,k} + \frac{G_y}{2\cdot(G_x+G_y+G_z)}\cdot T_{i,j-1,k}$$
$$+ \frac{G_z}{2\cdot(G_x+G_y+G_z)}\cdot T_{i,j,k+1} + \frac{G_z}{2\cdot(G_x+G_y+G_z)}\cdot T_{i,j,k-1}$$
$$+ \frac{1}{2\cdot(G_x+G_y+G_z)}\cdot I_{i,j,k}$$

g_x, g_y, g_z, g_I

$$T_{i,j,k} = g_x T_{i+1,j,k} + g_x T_{i-1,j,k} + g_y T_{i,j+1,k} + g_y T_{i,j-1,k}$$
$$+ g_z T_{i,j,k+1} + g_z T_{i,j,k-1} + g_I I_{i,j,k}$$

Boundary Conditions



$$T_{i,j,k} = T_0$$ for all {i,j,k} at boundary

Thermal Equation

$$T_{i,j,k} = g_x T_{i+1,j,k} + g_x T_{i-1,j,k} + g_y T_{i,j+1,k} + g_y T_{i,j-1,k}$$
$$+ g_z T_{i,j,k+1} + g_z T_{i,j,k-1} + g_I I_{i,j,k}$$

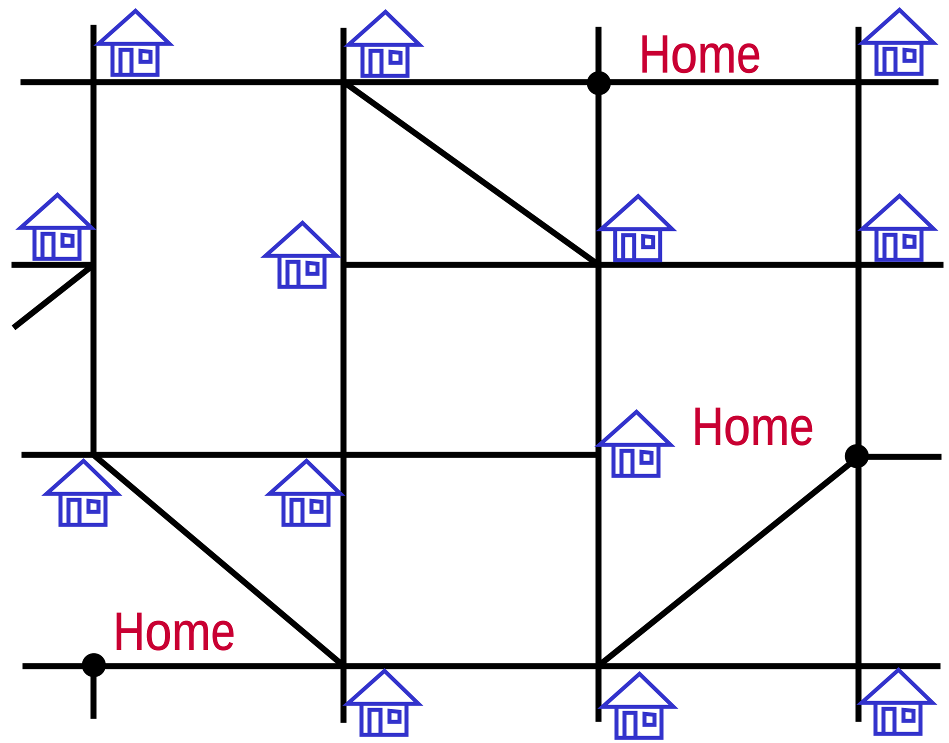
$$T_{i,j,k} = T_0 \quad @ \quad Boundary$$

- Linear thermal equation can be solved by many techniques
  - Gaussian elimination
  - Conjugate gradient method
  - Etc.

- In this lecture, we will explore a new "random" technique to solve discretized PDE

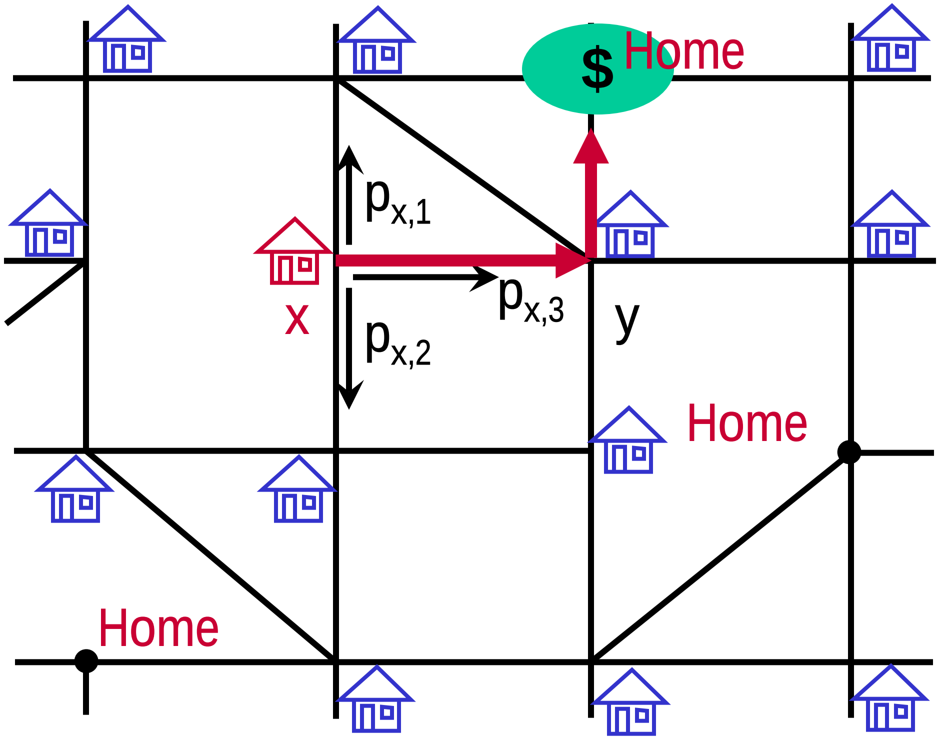
Random Walk Game

- Random walk game
  - A network of roads
  - A motel at each intersection
  - A set of homes



Random Walk Game

- Start from node x
- Walk one (randomly chosen) road every day
  - Each direction is associated with probability $p_{i,j}$
- Stay the night at motel y
- Motel charges m_y
- Keep going until home
- Get reward m_0 at each home



Random Walk Game

- Problem: find the average amount of earned money in the end as a function of the starting node x

  f(x) = E[money earned in the end | from node x]

- f(x) can be estimated by Monte Carlo analysis
  - Estimate the expected value from a number of random sampling points

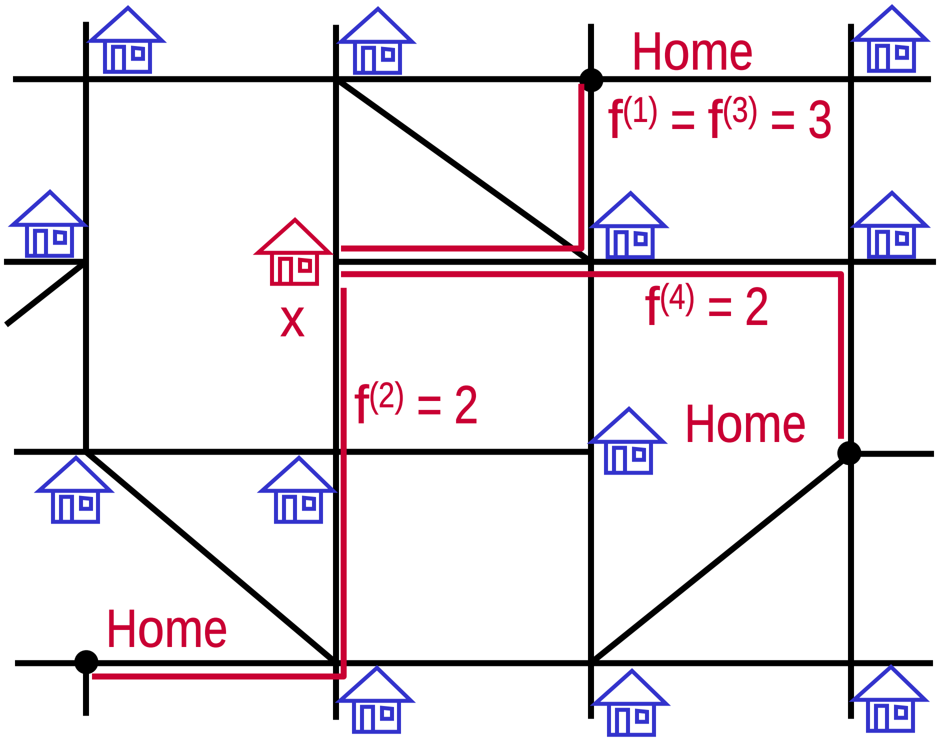
Monte Carlo Analysis

f(x) = E[money earned in the end | from node x]

- For i = 1,2,...,M
  - Start from node x
  - Perform random walk to reach home
  - Calculate the total money earned during this walk: $f^{(i)}$
- End For
- f(x) is estimated by:

$$f(x) \approx \frac{1}{M} \cdot \sum_{i=1}^{M} f^{(i)}$$

Monte Carlo Analysis

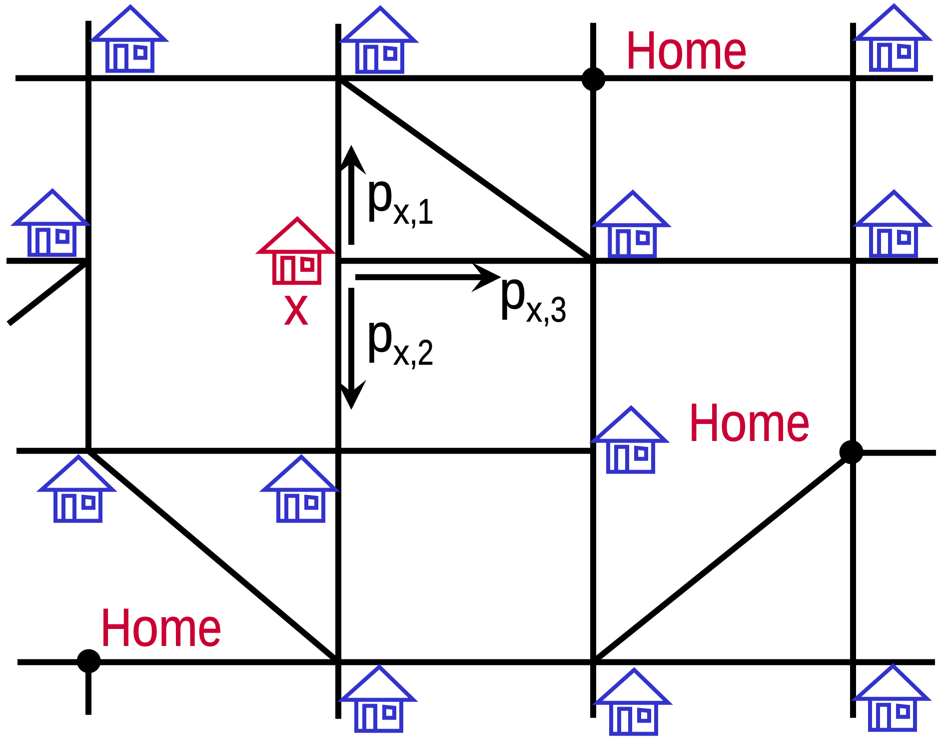


$$f(x) \approx \frac{1}{4} \cdot (3+2+3+2) = 2.5$$

- Why is random walk game related to thermal analysis?
  - To understand the connection, we need to analytically model the game as a Markov chain

Markov Chain Model

- The random walk problem can be modeled as a Markov chain
  - The transition probability from x to y is uniquely determined by x and y only
  - It is independent of any previous locations

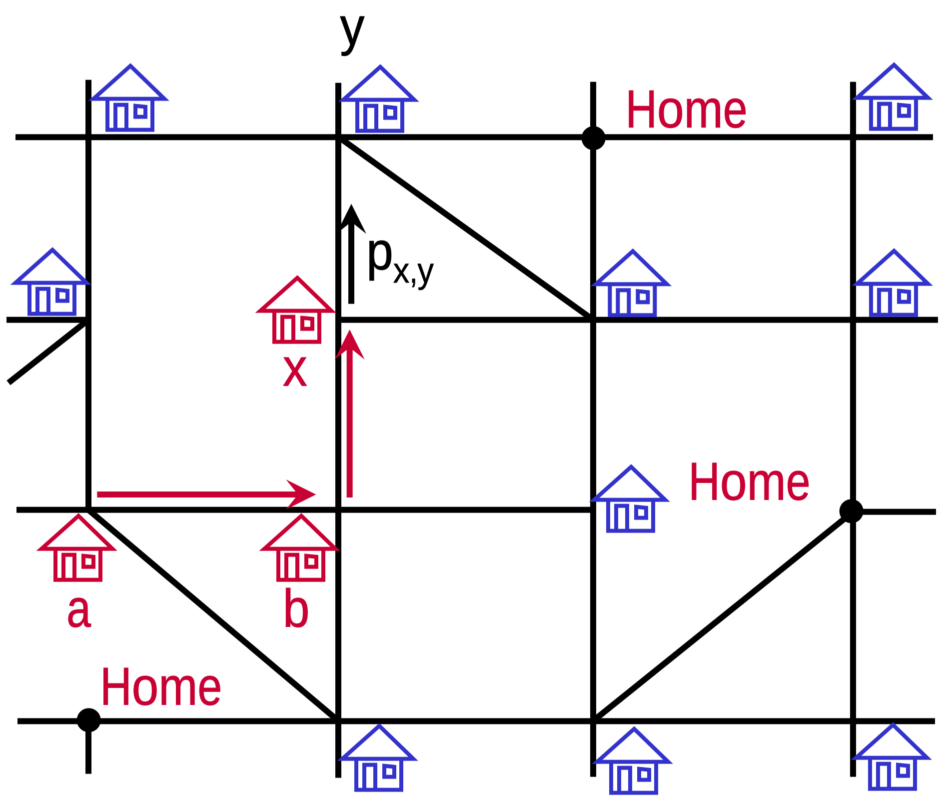


Markov Chain Model

- Mathematically, it means:

$$p(y \mid x, b, a) = p(y \mid x) = p_{x,y}$$

Probability is independent of old history



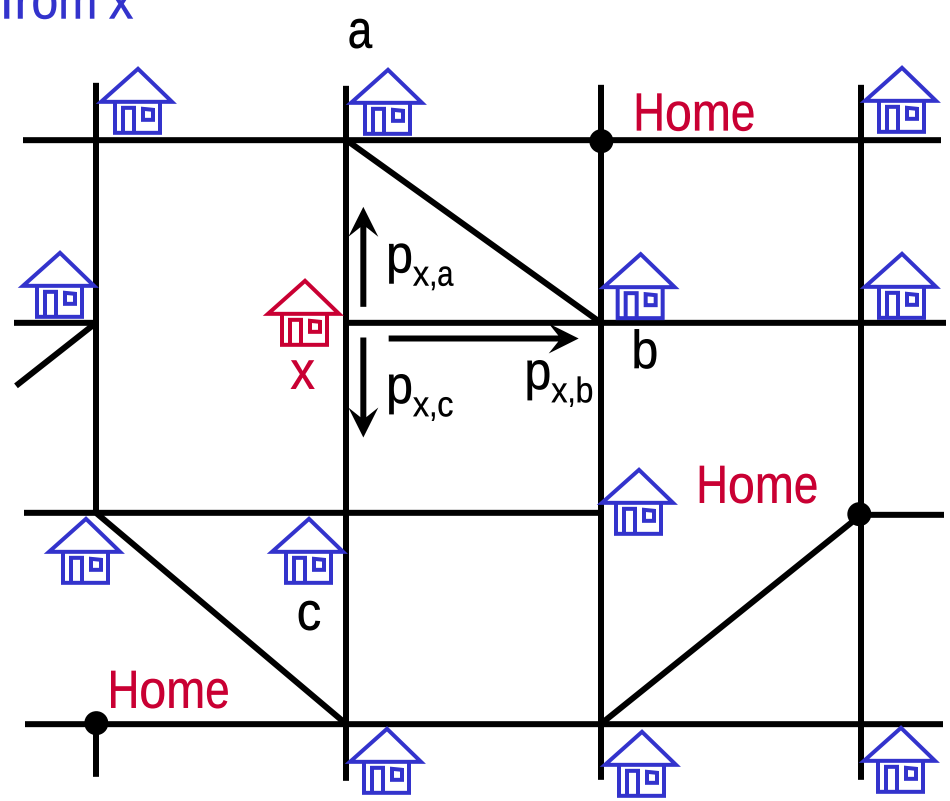
Markov Chain Model

- Linear equation for f(x)

$$f(x) = p_{x,a} f(a) + p_{x,b} f(b) + p_{x,c} f(c) - m_x$$

Money earned by starting from x

Payment made at x

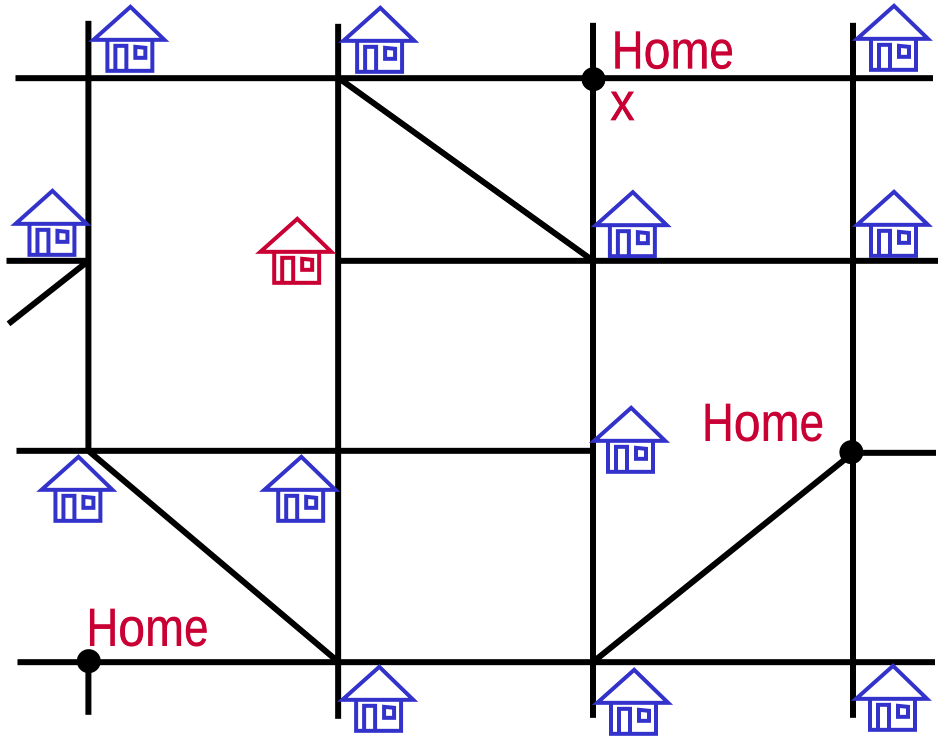


Markov Chain Model

- Linear equation for f(x)

$$f(x) = m_0$$

Reward at x

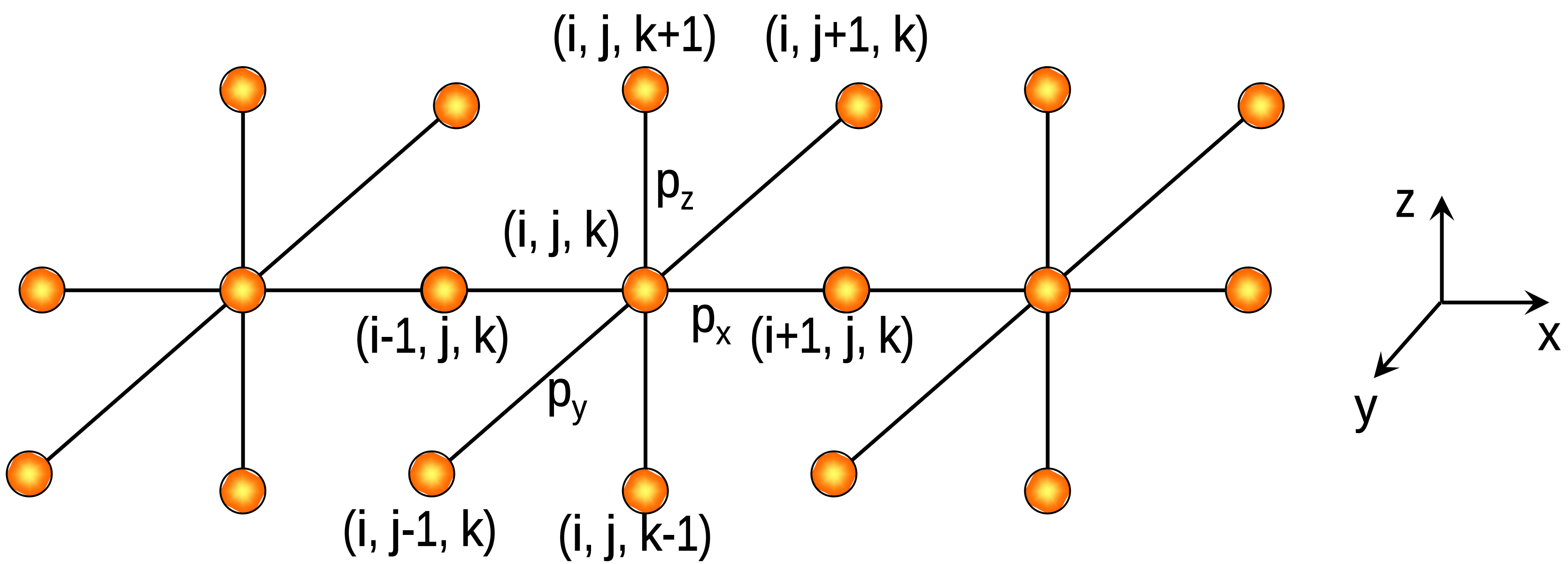


Random Walk Game for 3-D Grid

- If the road network is a 3-D grid

$$f_{i,j,k} = p_x f_{i+1,j,k} + p_x f_{i-1,j,k} + p_y f_{i,j+1,k} + p_y f_{i,j-1,k} + p_z f_{i,j,k+1} + p_z f_{i,j,k-1} - m_{i,j,k}$$

{i,j,k} is not home

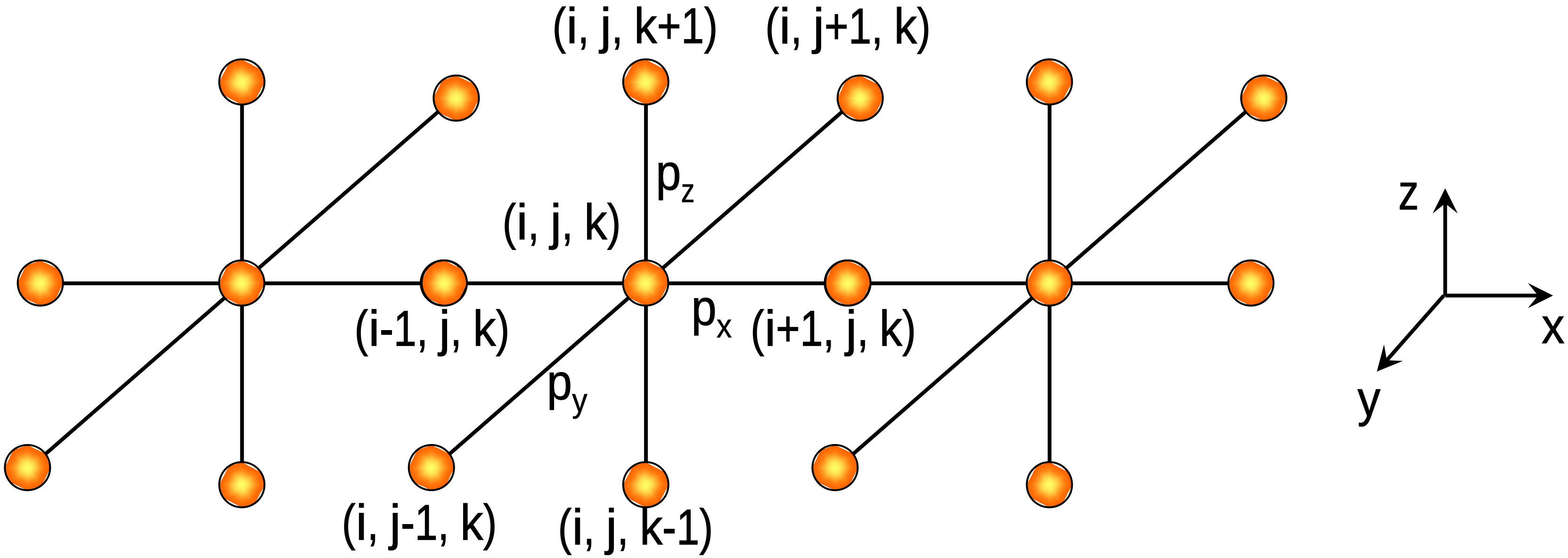


Random Walk Game for 3-D Grid

- If the road network is a 3-D grid

$$f_{i,j,k} = m_0$$

{i,j,k} is home



Thermal Analysis vs. Random Walk Game

- Both thermal analysis and random walk game can be modeled by similar linear equations

$$T_{i,j,k} = g_x T_{i+1,j,k} + g_x T_{i-1,j,k} + g_y T_{i,j+1,k} + g_y T_{i,j-1,k} + g_z T_{i,j,k+1} + g_z T_{i,j,k-1} + g_I I_{i,j,k}$$

$$T_{i,j,k} = T_0$$

@ boundary

Thermal analysis

$$f_{i,j,k} = p_x f_{i+1,j,k} + p_x f_{i-1,j,k} + p_y f_{i,j+1,k} + p_y f_{i,j-1,k} + p_z f_{i,j,k+1} + p_z f_{i,j,k-1} - m_{i,j,k}$$

$$f_{i,j,k} = m_0$$

@ home

Random walk game

Thermal Analysis vs. Random Walk Game

- This observation has a two-fold meaning

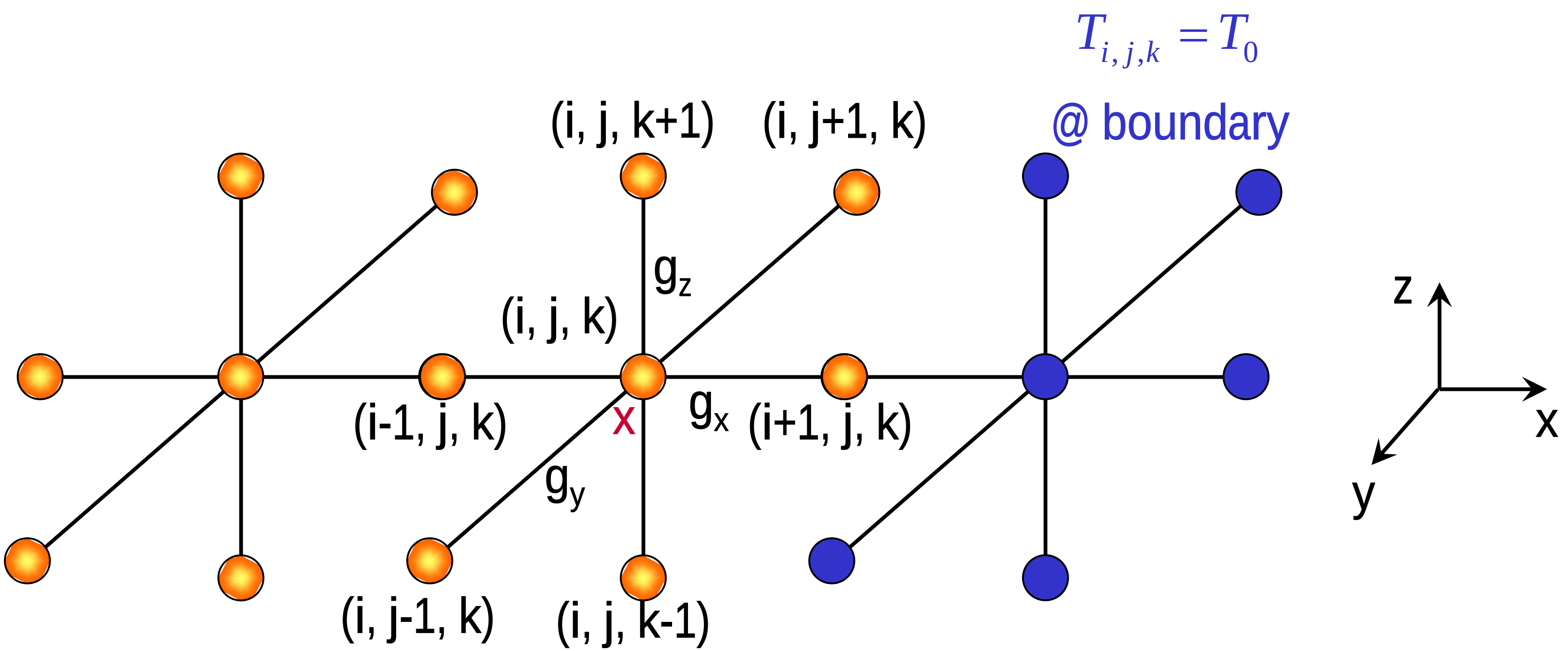
- #1: We do not have to play random walk game by Monte Carlo
  - It can be solved deterministically based on linear equation

- #2: We do not have to solve thermal analysis problem deterministically
  - Temperature solution can be found by Monte Carlo analysis (i.e., a randomized algorithm)

Random Walk for Thermal Analysis

- Problem: find temperature at x by random walk

$$T_{i,j,k} = g_x T_{i+1,j,k} + g_x T_{i-1,j,k} + g_y T_{i,j+1,k} + g_y T_{i,j-1,k} + g_z T_{i,j,k+1} + g_z T_{i,j,k-1} + g_I I_{i,j,k}$$



Random Walk for Thermal Analysis

- Random walk
  - Start from node x with reward $g_I I_x$
  - Walk across one (randomly chosen) edge where each walking direction is associated with a probability g_x, g_y or g_z
  - Reach node {i,j,k} and get reward $g_I I_{i,j,k}$
  - Keep going until reaching boundary
  - Get reward T_0 at boundary
  - Calculate the total reward earned during this walk

$$T_{i,j,k} = g_x T_{i+1,j,k} + g_x T_{i-1,j,k} + g_y T_{i,j+1,k} + g_y T_{i,j-1,k} + g_z T_{i,j,k+1} + g_z T_{i,j,k-1} + g_I I_{i,j,k}$$

Random Walk for Thermal Analysis

- Monte Carlo analysis

$$T_x = E[\text{ reward earned in the end} \mid \text{from node } x\ ]$$

- For i = 1,2,...,M
  - Start from node x
  - Perform random walk to reach boundary
  - Calculate the total reward earned during this walk: $f^{(i)}$
- End For
- T_x is estimated by:

$$T_x \approx \frac{1}{M} \cdot \sum_{i=1}^{M} f^{(i)}$$

Deterministic Solver vs. Random Walk

- The efficacy of both algorithms is problem-dependent

- In general, random walk is preferable if we are only interested in local temperature

  T_x = E[reward earned in the end | from node x]

  - We do not have to solve the complete linear equation
  - Random walk quickly tells us the temperature at “a” location x

- Random walk can also be used to generate “good” pre-conditioner for conjugate gradient method

Summary

- Random walk
  - 3-D heat equation
  - Random walk game
  - Randomized PDE solver