Ojas Bardiya
UID: 505145284
CEE/MAE 20
07/31/20

1. Shared Birthday Problem

1.1. Introduction

We use a Monte-Carlo simulation to determine how many people we should include in a group before there is a higher probability of two people having a birthday during the same week than there isn't, i.e., **P(2 people have birthdays in the same week) > 0.5**. We do this using 10000 trials to simulate the number of people added in a group before 2 of them have a birthday in the same week.

1.2. Model and Theory

We use a 10000*1 array to hold the value for each trial, i.e., the number of people added to a group before a birthday within the same week occurs. We then find the median of the distribution which gives us our required value.

2 people are considered to have a birthday occurring in the same week if there are less than 7 days in either direction of a chosen day. For this, we determine whether either of the two expressions –

Abs(current day – new day) or
Abs(Abs(current day – new day) – 365)

are true.
Note: Abs denotes the absolute value of the difference.

1.3. Pseudocode

Set Number of Trials
Initialize array for storing results of each trial
Start the Monte Carlo Simulation
    Generate a random birthday
    Loop through all days already in the group until a match is found
        Generate another random birthday
        Determine if a match exists
    Add new birthday to the existing group
Store the length of the group in the array
Print the median
Create a histogram

1.4. Calculations

For 10000 trials and assuming each birthday has an equal probability of happening, we see that –

Median Number of People = 7

This essentially implies that if we choose a group of people with greater than/equal to 7 people it is more likely than not to find 2 people in the group whose birthdays happen in the same week.

Since we assume each birthday has an equal chance of occurring this seems reasonable.

For 10000 trials and assuming that not all birthdays are equally likely, we would expect a **smaller** median.

For example, if we assume that birthdays are more likely to occur during the 2^{nd} half of a year than the first, the distribution of birthdays would be skewed towards the last 6 months and it would generally take a fewer number of people in a group to find a birthday in the same week.

1.5. Conclusion

We observe that a group of 7 people or more is safe bet for finding 2 people having birthdays in the same week.

2. Random-Walk Collision

2.1. Introduction

We use a Monte-Carlo simulation to estimate the number of moves it takes 2 random walkers to collide in the specified 11*11 grid. We conduct 5000 trials, where in each case we determine the number of moves it takes before the two particles collide.

2.2. Model and Theory

We find the median over the set of trials to find the number of moves for which it is more likely than not they will collide.

We give each particle a probability of p = 0.2 to move North, South, West, East or stay put and generate a random number between 0 and 1 to determine which of these moves to execute.

We also set boundary conditions to make sure each particle doesn't move off the grid.

2.3. Pseudocode

For the RandWalk_2D function

Create a random number between 0 and 1 for determining the direction of movement

If the Move is north

    Update the position accordingly

    Check to see that boundary conditions are satisfied

    If not, keep in same position

If the Move is south

    Update the position accordingly

    Check to see that boundary conditions are satisfied

    If not, keep in same position

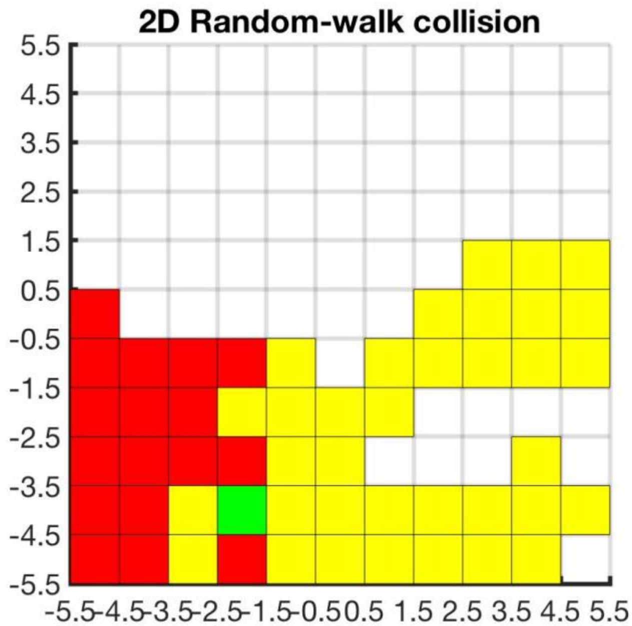
If the Move is west

Update the position accordingly
Check to see that boundary conditions are satisfied
If not, keep in same position
If the Move is east
Update the position accordingly
Check to see that boundary conditions are satisfied
If not, keep in same position
If the particle stays still
Position is unchanged
end

For the main program
Set Boundary conditions
Set number of trials
Set initial conditions for positions of particles A and B
Initialize array to hold number of moves taken per trial
Start the Monte Carlo simulation
Set the number of moves
Until a collision occurs or the number of moves exceeds the maximum
Perform a random walk for both A and B
Create a grid to visualize the random walk
Update the positions of both particles
Determine if a collision has occurred
Set the corresponding array index to the number of moves
If no collision after moves exceeds the maximum
Set the corresponding array index to the number of moves
Display the median

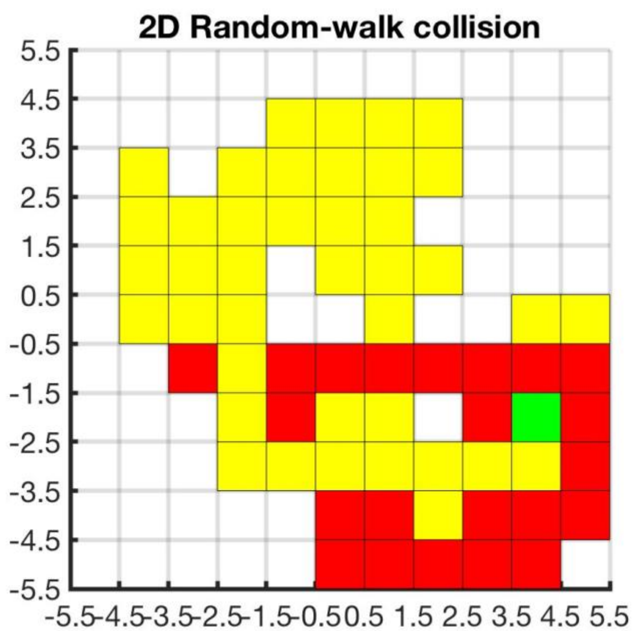
2.4. Calculations

We see that the median usually lies between **210-225** moves when both A and B perform the random walk.

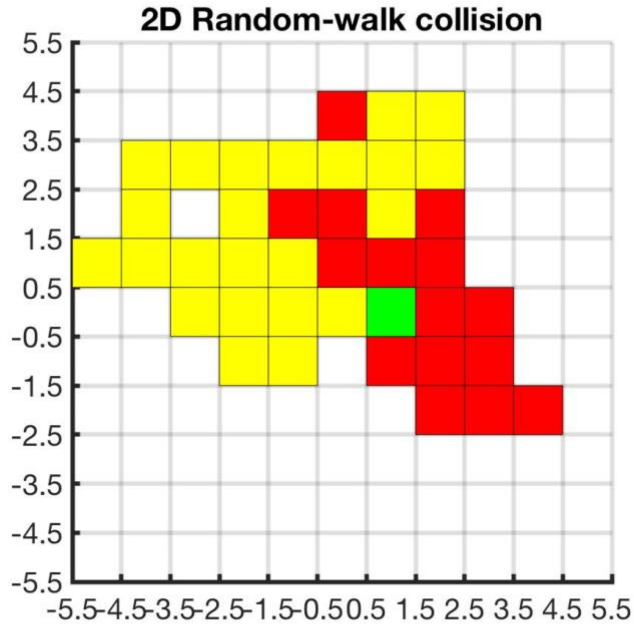


When we keep B stationary, we observe that the median usually lies between **420-430** moves. This clearly contradicts the idea that a person should stay where they are until help arrives as this would take a far greater time on average.

When the initial positions are (-4, 0) and (4, 0) we don't observe any noticeable difference in the median; it is a bit less but still lies between **210-220.**



However, when the initial positions are (-3,0) and (3, 0) we see a slight decrease in the median; it now lies in the range of **190-200.**



2.5. Conclusion

We see that it is far more efficient for two particles to move towards each other randomly rather than one staying put in order for a collision to occur.