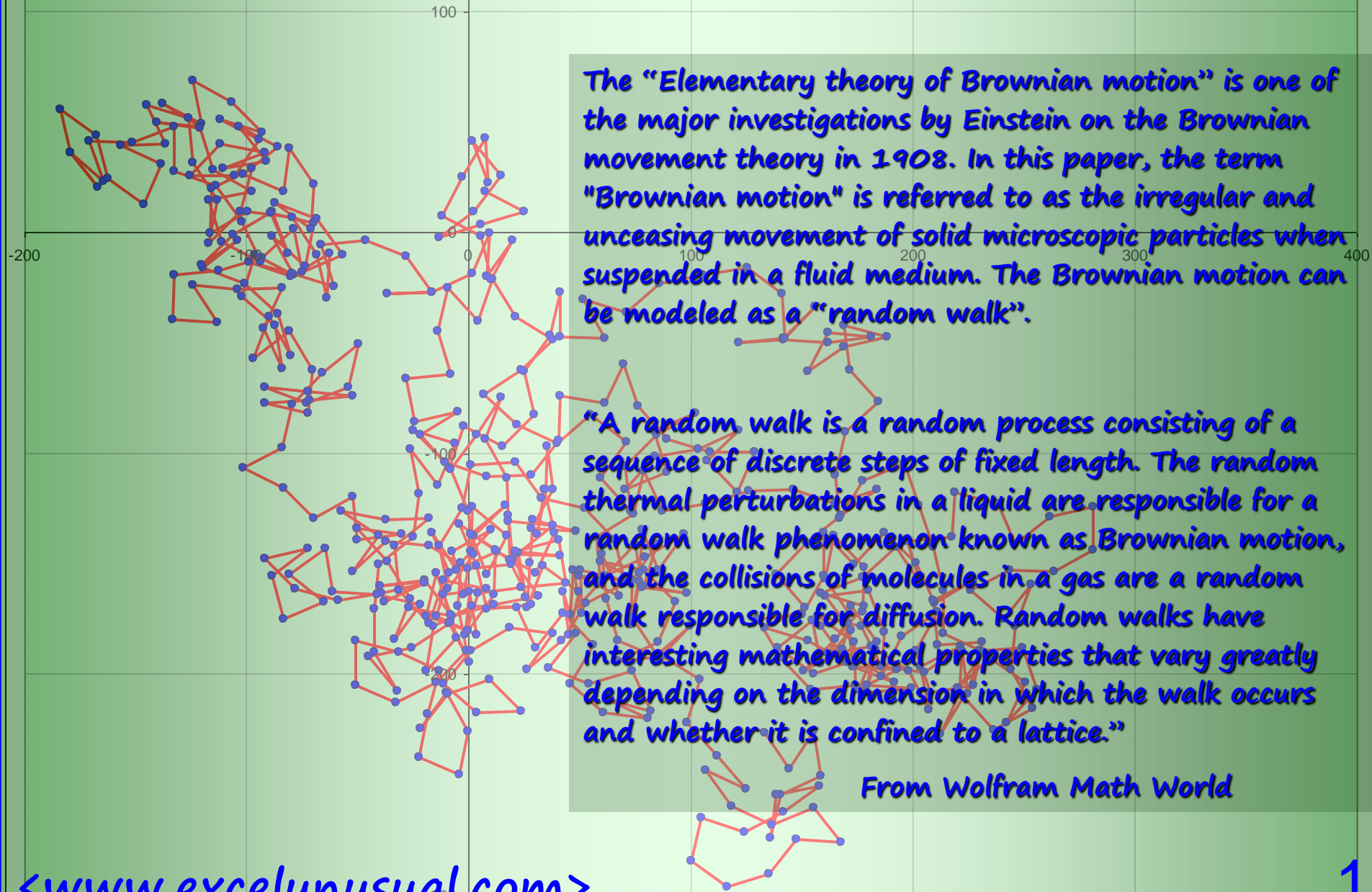


A 2D Random Walk tutorial

by George Lungu

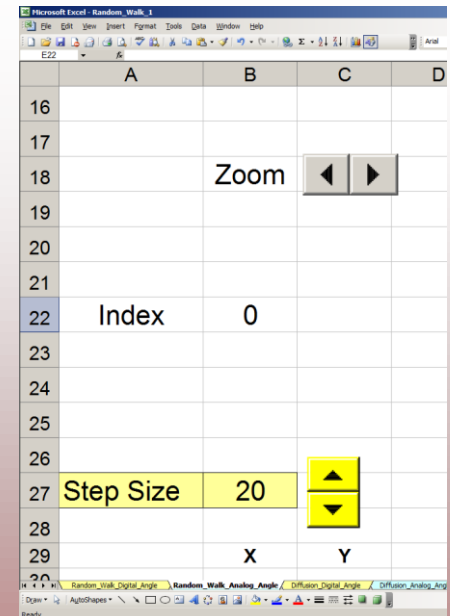


A lattice confined (digital angle) random walk:

- I refer to a random walk as “digital” if the particle is confined to a lattice. This is not a standard term, I made it up using electrical engineering jargon.
- Let's assume that the 2D lattice is square, so a particle, at any moment will jump a step equal to the lattice constant. The movement in this case can be in only one of the following directions: right, forward, left or backwards.
- We will use the random number generator spreadsheet function “rand()” which returns a random decimal number uniformly distributed between 0 and 1.
- We will also use the function “Int” which rounds a number down to the nearest integer.
- The compound function “Int(4*Rand())” will generate the numbers 0, 1, 2 and 3 in random order with equal probability.

The Excel implementation:

- Let's create two worksheets one with a digital random walk (lattice confined) and one with an analog version.
- Name the first worksheet “Random_Walk_Lattice”
- The following cells contain labels: B18, A22, A27, B29, C29
- This will be a dynamic model run as a conditional “Do” loop. Cell B27 (Index) keeps track of the number of times the loop was run since the last reset.



- There are two spin buttons, one associated with the “Step Size” input parameter which can vary that parameter from 1 to 100 in increments of “1”, and a chart zoom spin button (we will talk about that later).
- The first (yellow) spin button is named “Step_Size” and has a range of 1 to 100. The associated VBA code is shown to the right.

```
Private Sub Step_Size_Change()  
[B27] = Step_Size.Value  
End Sub
```

The active equations:

- The trajectory of the particle will be tabulated in range (B31:C10030) with the current time in range B31:C31 and the past just in the cells underneath (range B32:C32 will contain the x-y coordinates one time step in the past).
- Cell A31: “=INT(4*RAND())” this formula will generate a new random integer number between 0 and 3 every time the worksheet is recalculated
- Cell B31: “=B32+IF(A31=0,\$B\$27,0)-IF(A31=2,\$B\$27,0)” this formula will move the particle to the right by a value equal to one spatial step size if the value in cell A31 is a 0 and it will move it to the left by the spatial step size if A31 is equal to 2. This movement is with respect to the position of the particle in the previous time step whose coordinates are in range B32:C32.
- Cell C31: “=C32+IF(A31=1,\$B\$27,0)-IF(A31=3,\$B\$27,0)” this formula will move the particle forward by a value equal to one spatial step size if the value in cell A31 is a 1 and it will move it backwards by the spatial step size if A31 is equal to 3. This movement is with respect to the position of the particle in the previous time step whose coordinates are in range B32:C32.

The dynamic (animation) macros:

- The active formulas calculate the present particle coordinate recursively from the past coordinates. A copy/paste type of macro is needed to create animation.
- The "Reset" macro will delete all historical data.
- "s" is a Boolean variable and can take only two values: true or false. The purpose of this variable is to keep track if the active macro (Start_Pause) runs or it is stopped. Another purpose of this variable is to stop the active macro if the macro is triggered when the conditional "Do" loop is running.
- The "Start_Pause" macro contains a conditional loop. If the loop is not running it means "s = False". Clicking the "Start / Pause" button will flip the "s" variable from False to True and the loop will start and continue to run until the button is clicked again. We can see therefore that "s" has the main role of being able to both start and stop the macro using the same button.
- The conditional "Do" loop within the "Start_Pause" macro copies all the run data and pastes it one time step in the past (one row down), therefore, effectively simulating the passage of time and dynamically advancing the calculations in time.

Public s As Boolean

```
Sub Start_Pause()
```

```
s = Not (s)
```

```
Do Until s = False
```

```
DoEvents
```

```
[B22] = [B22] + 1
```

```
[A32:C10032] = [A31:C10031].Value
```

```
DoEvents
```

```
Loop
```

```
End Sub
```

```
Sub Reset()
```

```
[B22] = 0
```

```
[A32:C10300].Clear
```

```
End Sub
```



Botanist Robert Brown (1773-1858)
the discoverer of Brownian motion

Create two buttons:

- Create a "Reset" button and a "Start / Pause" button out of rectangles with rounded corners using "Draw" menu.
- Assign the previous macros to these buttons.

Chart the trajectory:

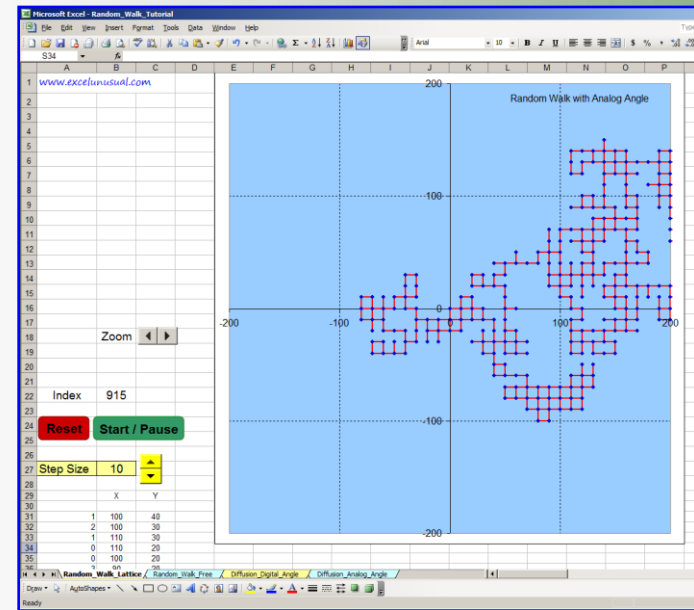
- Create a 2D scatter chart having B31:B10030 as the x-range and C31:C10030 as the y-range, this will chart 10000 points of the particle trajectory.

Rename the chart:

- Make the "Draw" toolbar visible. Click on the "Select Objects" arrow icon on the bottom of the page, then select the chart. The chart is now highlighted with small white circles in the corners.
- On right top of the page in the "Name Box" rename the chart "ChartA", then hit return

The chart scale macro:

- Create a button named Scale_XY with a range of 1 to 20. Using the macro below, the button will adjust the range of both x and y axes between [-100, 100] and [-2000, 2000].



```
Private Sub Scale_XY_Change()
```

```
ActiveSheet.ChartObjects("ChartA").Chart.Axes(xlCategory).MaximumScale = Scale_XY * 100
```

```
ActiveSheet.ChartObjects("ChartA").Chart.Axes(xlCategory).MinimumScale = -Scale_XY * 100
```

```
ActiveSheet.ChartObjects("ChartA").Chart.Axes(xlValue).MaximumScale = Scale_XY * 100
```

```
ActiveSheet.ChartObjects("ChartA").Chart.Axes(xlValue).MinimumScale = -Scale_XY * 100
```

```
End Sub
```

Verify the functionality of the model:

- Click "Start / Pause" and watch the model run. While the model is running hit "Reset" and verify that the trajectory is reset and the particle starts from coordinates (0, 0).
- Try changing the step size and watch the trajectory becoming coarser. Also verify the functionality of the chart scale adjusting macro by clicking the "Zoom".

Create a new worksheet for a free random walk model (not lattice confined):

- Copy the first worksheet "Random_Walk_Lattice" and rename it "Random_Walk_Free". This will be a model similar to the previous except that the particle is not confined to a lattice for every step it can move in any direction by a random angle uniformly distributed between 0 and 2π .
- Leave everything the same except for changing the active formulas in the range A31:C31

Cell A31: " $=2*PI()*RAND()$ " this formula will generate a new random number between 0 and 2π every time the worksheet is recalculated

Cell B31: " $=B32+\$B\$27*COS(A31)$ "

Cell C31: " $=C32+\$B\$27*SIN(A31)$ " - the last two formulas will move the particle by a value equal to one spatial step size and in a random direction based on a uniformly distributed angle (0 to 2π with the horizontal). This movement is with respect to the position of the particle in the previous time step whose coordinates are in the range B32:C32.

