

お父ちゃん無事家に帰り着くか？

いわゆる2次元酔歩問題です。Excel VBA でゲーム風にプログラム (Excel「酔歩問題.xlsx」) してみました。

筋書き

とある居酒屋で美酒にのどを潤しつつ、ふと腕時計を見るともう夜も更けている。そろそろ家に帰ろうかと支払いを済ませ、外にでるとなんと蒼白いばかりの煌々とした月明かりが夜道を照らしている。北東方向に位置する我が家までは歩いてすぐの距離だ。

さあ、家に帰ろう！

と歩きはじめると、なんと足は千鳥足。右への一步も左への一步も、前進も後退も、どの向きに一步を踏みだすかは1/4の確率。これでは運を天に任せるようなもの、家に帰れないかもしれん。。。酔ってふらふらする頭で

わが家は北東の方向だ

と思い直したお父ちゃんは思いつめた気持ちでそちらの方向に足を向けるべく夜道を歩きはじめるのでした。果たしてお父ちゃんは無事家に帰りつくことができるか。。¹

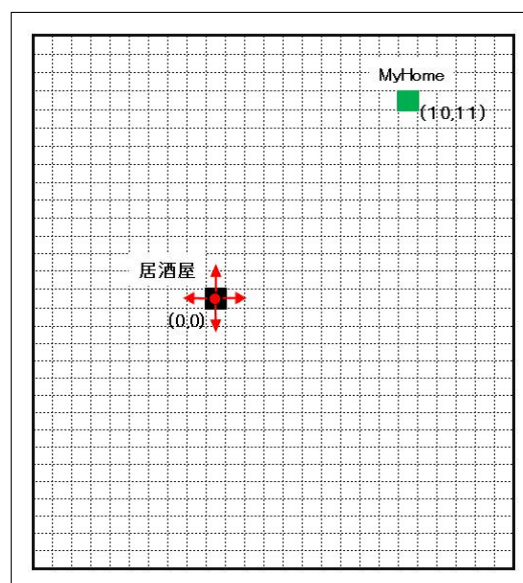


図 1: ゲームステージ

最短経路の数：酔っていなければ目をつむっていても帰れる！

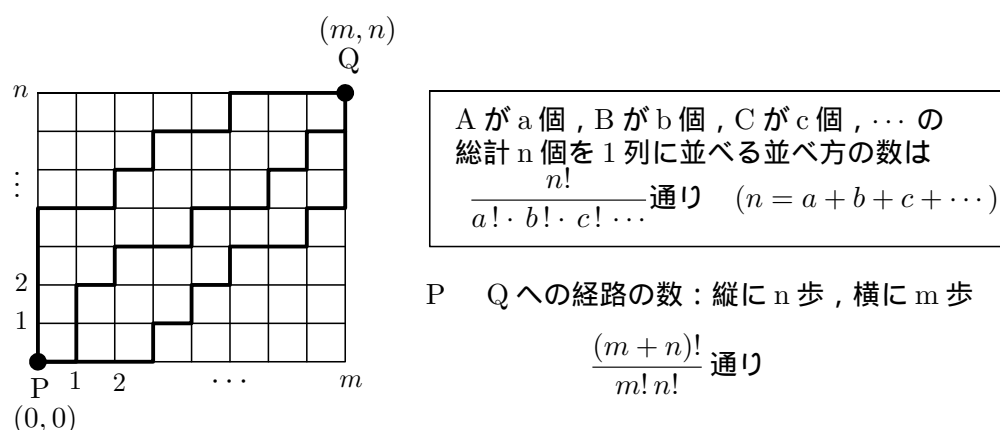


図 2: 最短経路の数

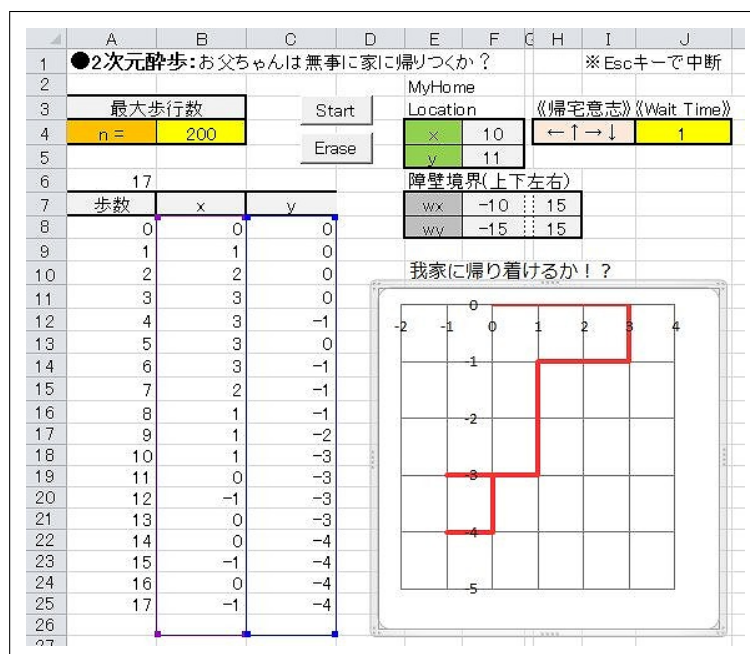
居酒屋の位置を原点 (0, 0) とすると MyHome は (10, 11) に位置する。つまり, 居酒屋からは右に 10 歩, 縦に 11 歩の位置にある。したがって, MyHome へたどり着く経路の数は

$$\frac{(11+10)!}{11! \cdot 10!} = 352716 \text{ 通り}$$

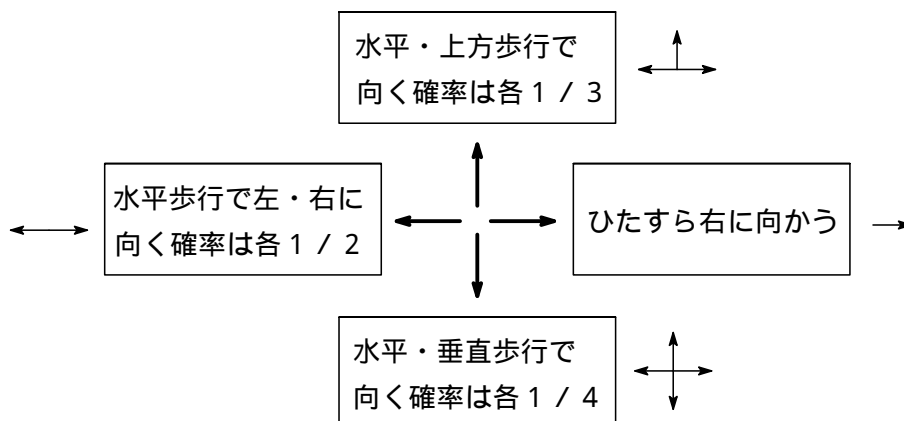
¹原点から歩み始めた酔歩者はいつか必ず原点に戻ります (再帰性)。ただし, 3 次元以上の酔歩問題では自由度が増した影響で酔歩者は永遠に彷徨い続けます。。

もあるので、酔っていなければ頭で縦・横の歩数をカウントしていれば間違いなく我が家へ帰ることができる。しかし今宵はうまい酒にめぐり合い不覚にも深酒をした。意識もぼんやりとし、足は千鳥だ。「ええい、儘（ママ）よ」とお父ちゃんは意識を北東に向かわせながら我が家に向けて歩み続けるのでした。お父ちゃん大丈夫かなあ～。。

2次元酔歩：「酔歩問題.xlsx」の「2次元酔歩 GM」シート



- ・最大歩行数：歩数を入力。
- ・Wait time：次の一步を踏みだすまでの遅延時間。1～5の数値（単位は0.1秒）を入力。
- ・Start ボタン：ボタンを押すと酔歩が始まります。Esc キーを押すと終了。
- ・Erase ボタン：歩行記録とグラフの消去。
- ・帰宅意識：上下左右の矢印キーを叩くことでお父ちゃんの歩む向きをリアルタイムに修正（セルに意識をリアルタイム表示）。エクセルシートの図1の画面を見ながらキー操作をうまくコントロールしてください。



- ・MyHomw Location：我が家の座標 (10, 11)。ちなみに居酒屋は原点 (0, 0)
- ・障壁境界：上下左右に障壁を設けました。壁に当たると正反対方向に向きます。

<ソースコード>

```
Option Explicit
Public n As Integer
```

```
'Windows API 宣言 (押されたキーの状態を取得する関数)
```

```
Private Declare PtrSafe Function GetAsyncKeyState Lib "User32.dll" (ByVal vKey As Long) As Long
```

```
'PlaySound API (VBA でサウンド再生)
```

```
Private Declare PtrSafe Function PlaySound Lib "winmm.dll" Alias "PlaySoundA" _
    (ByVal lpszName As String, ByVal hModule As Long, ByVal dwFlags As Long) As Long
```

```
Sub RandomWalk()
```

```
    Dim x As Integer, y As Integer
```

```
    Dim Mx As Integer
```

```
    Dim My As Integer
```

```
    Dim wx0 As Integer, wx1 As Integer
```

```
    Dim wy0 As Integer, wy1 As Integer
```

```
    Dim Mnd As Integer
```

```
    Dim i As Integer, rd As Integer
```

```
    Dim cnt As Integer
```

```
    Dim wt As Single
```

```
    Mnd = 4
```

```
    n = Cells(4, 2).Value
```

```
' 最大歩行数
```

```
    wt = 100 * Cells(4, 10).Value
```

```
' wait time(msec)
```

```
    cnt = 0
```

```
    x = 0: y = 0
```

```
' 出発点の座標
```

```
    Mx = Cells(4, 6).Value
```

```
' MyHome 地点
```

```
    My = Cells(5, 6).Value
```

```
    wx0 = Cells(7, 6).Value
```

```
' 障壁の座標
```

```
    wy0 = Cells(8, 6).Value
```

```
    wx1 = Cells(7, 8).Value
```

```
    wy1 = Cells(8, 8).Value
```

```
    Range("H4").Value = " "
```

```
' 帰宅意識
```

```
    Cells(8, 1) = cnt
```

```
' 歩行数
```

```
    Cells(8, 2) = x
```

```
' 酔歩者の位置
```

```
    Cells(8, 3) = y
```

```
    For i = 1 To n
```

```
' 最大既定歩数までの繰り返し
```

```
        ' 矢印キー入力で歩む方向の確率を偏らせる
```

```
        If GetAsyncKeyState(vbKeyEscape) Then
```

```
' Esc キーで中断
```

```
            GoTo Out 'Exit For
```

```
        ElseIf GetAsyncKeyState(vbKeyRight) Then
```

```
            Mnd = 1: Range("H4").Value = " "
```

```
        ElseIf GetAsyncKeyState(vbKeyLeft) Then
```

```
            Mnd = 2: Range("H4").Value = " "
```

```
        ElseIf GetAsyncKeyState(vbKeyUp) Then
```

```
            Mnd = 3: Range("H4").Value = " "
```

```
        ElseIf GetAsyncKeyState(vbKeyDown) Then
```

```
            Mnd = 4: Range("H4").Value = " "
```

```
        End If
```

```
' 1 から Mnd までの整数の乱数を発生
```

```
        rd = Application.WorksheetFunction.RandBetween(1, Mnd)
```

```
        Select Case rd
```

```
' 酔歩者の移動
```

```
            Case 1
```

```
                x = x + 1
```

```

Case 2
  x = x - 1
Case 3
  y = y + 1
Case 4
  y = y - 1
End Select

```

’ 縦の側壁跳ね返される

```

If x >= wx1 Then x = x - 1
If x <= wx0 Then x = x + 1

```

’ 横の側壁に跳ね返される

```

If y >= wy1 Then y = y - 1
If y <= wy0 Then y = y + 1

```

```

Cells(i + 8, 1).Value = i          ’ 歩数

```

```

Cells(i + 8, 2).Value = x        ’ 酔歩者の x 座標

```

```

Cells(i + 8, 3).Value = y        ’ 酔歩者の y 座標

```

```

Application.Wait [Now()] + wt / 86400000 ’Wait(msec)

```

```

Cells(15 - y, 21 + x) = " "

```

```

If (x = Mx And y = My) Then Exit For ’ 我が家に到着したら

```

```

Range("A6").Value = i

```

```

Next i

```

```

If i <= n Then

```

```

  Range("E11").Value = i

```

```

  PlaySound "c:\Windows\media\tada.wav", 1, 1

```

```

  Range("F11").Value = "歩で我が家にたどり着いた!!"

```

```

Else

```

```

  PlaySound "c:\Windows\media\chord.wav", 1, 1

```

```

  Range("E11").Value = " Ah! 我が家にたどり着けず。。。"

```

```

End If

```

```

Out:

```

```

  PlaySound "c:\Windows\media\notify.wav", 1, 1

```

’ 近似解曲線を散布図で描く

```

With ActiveSheet.Shapes.AddChart.Chart

```

```

  .ChartType = xlXYScatterLinesNoMarkers

```

’ 平滑線付き散布図マーカーなし

```

  .SetSourceData Range(Cells(8, 2), Cells(8 + i, 3))

```

’ 作図範囲

```

  .Axes(xlValue).HasMajorGridlines = True

```

’ False で補助線消去

```

  .Axes(xlCategory).HasMajorGridlines = True

```

```

  .HasLegend = False

```

’ 凡例なし

```

  .Axes(xlCategory).MajorUnit = 1

```

’ x 軸のスケール

```

  .Axes(xlValue).MajorUnit = 1

```

’ y 軸のスケール

```

  .SeriesCollection(1).Format.Line.Weight = 3

```

’ 歩行軌跡の線の太さ

```

  .SeriesCollection(1).Format.Line.ForeColor.RGB _

```

```

= RGB(255, 50, 50)

```

’ 線の色 (橙色)

```

  .ChartArea.Select

```

’ グラフ外枠の設定

```

    With Selection

```

```

      .Width = 200

```

```

      .Height = 200

```

```

    End With

```

```

  .PlotArea.Select

```

’ 描画領域の設定

```

        With Selection
            .Left = 0
            .Top = 0
        End With
    End With

    ' グラフの表示位置の設定
    With ActiveSheet.ChartObjects
        .Top = Range("E15").Top           ' 縦位置
        .Left = Range("E15").Left        ' 横位置
    End With

End Sub

' 計算結果とグラフの消去
Sub Erase_Click()
    Dim i As Long
    Range(Cells(8, 1), Cells(8 + n, 3)).Clear ' セルに格納したデータを消去
    Range(Cells(1, 11), Cells(30, 35)).ClearContents
    Range("A6").Clear
    Cells(4, 8).Value = "          "
    Range("E11:F11").Clear
    With ActiveSheet                               ' 複数埋込グラフ全削除
        For i = .ChartObjects.Count To 1 Step -1
            .ChartObjects(i).Delete
        Next i
    End With
End Sub

```

2次元8方向のランダムウォークはExcelの「2d 酔歩八方向」シートに載せています。
特に何の加工もしていませんのでいろいろコードを改変するのも面白い。。。かな？

「Start ボタン」で酔歩開始。酔歩終了後「r : N」ボタンを押すと原点からの移動距離と歩数の平方根の関係グラフが表示されます。「Erase ボタン」でデータとグラフを一括消去。

数学的補足

A) 1次元ランダムウォーク

2次元ランダムウォークは碁盤の目を上・下・左・右の方向にランダムに一步を進めました。ここでは数学的補足ということで次元を落として直線上を「右」か「左」の2つの方向の内のいずれかにランダムに歩む1次元ランダムウォークから話から始めてみましょう。これはコインを投げて表がでれば右へ一歩、裏が出れば左へ一歩という問題に焼き直せますね。そこでしばらくは確率分布のお話をして、それから1次元ランダムウォークの話に入ることになります。

- ・ベルヌイ試行：コインを投げると「表」か「裏」の2種類のうち必ずいずれかになり、表か裏いずれかとなるかはそれまでの結果と全く無関係です（表となったから次は裏とはならず、前後の因果関係はありません）。このような試みのことをベルヌイ（Bernoulli）試行と呼んでいます。例えば5回のコイン投げを1セット（ベルヌイ試行）とし、1セット試行のたびに「表」が何回出たかを記録します。ベルヌイ試行を何セットも繰り返し、そのたびに「表」が出た回数を記録していくと、2回であったり、1回であったり、4回であったりと色々な回数となりますね。それではベルヌイ試行を多数回行ったとき、「表」が $k(\leq n)$ 回でる確率は？（注参照）。理論的にはこの確率は2項分布で与えられます（Excel「2項分布」シート参照）。
- ・2項分布：コインを1回投げて表になる確率を p とします（裏になる確率は $q = 1 - p$ ）。5回コイン投げて「表」が2回でる確率を求めると次のようになります。

● 表
○ 裏

5回コインを投げて表2回、裏3回となる確率は p^2q^3 （確率の積）

→ コイン投げの回数

コインを5回投げて「表」が2回でる場合の数は（左図参照）
 ${}_5C_2 = \frac{5!}{2!(5-2)!} = 10$ 通り

⇒ 「表」が2回でる確率は各場合の確率の和となるので

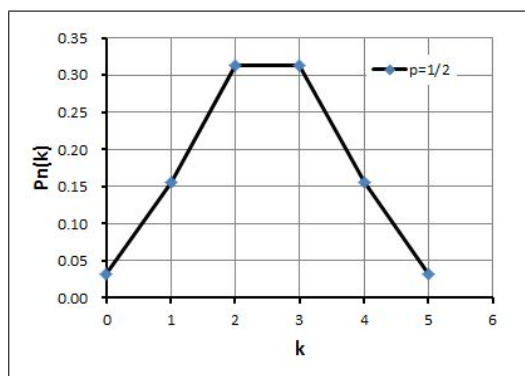
${}_5C_2 p^2 q^3 = 0.31$

2回でる場合: 総計 10 通り

一般に事象 A が起こる確率を p , n 回の試行で事象 A が k 回起こる確率を $P_n(k)$ とすると

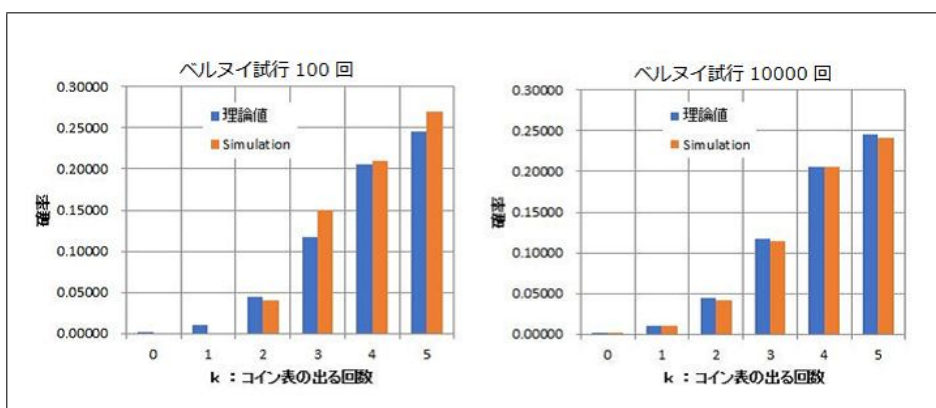
$$P_n(k) = {}_n C_k p^k (1-p)^{n-k} = \binom{n}{k} p^k q^{n-k} \quad (k = 0, 1, 2, \dots, n; p + q = 1) \quad (1)$$

で与えられ、これを2項分布と呼んでいます。ちなみに $\binom{n}{k}$ を2項係数といいます。 $n = 5, p = 0.5$ の2項分布を下図に示します。



(注) 泥臭くコイン投げのベルヌイ試行を多数回やった場合の確率の推移を見てみましょう(表の出る確率 $p = 0.5$ としています)。実際にコイン投げを多数回やるのは大変ですからコンピューターにやらせることにします。

その結果, 左のグラフは 10 回のコイン投げ 1 セットを 100 回繰り返したときの「表」の出る回数の確率分布で, 右のグラフは 10000 回セット繰り返したときのものです。理論値は 2 項分布の計算値で, Simulation はベルヌイ試行を多数回したときの値です。10000 回程度で理論値に近い値となっている様子が分かります。確率というのはその裏に非常に多数回の試行が隠されているということが改めて分かる気がしますね。UP しているエクセルファイルの「ベルヌイ試行.xlsx」を開いている試してみてください。

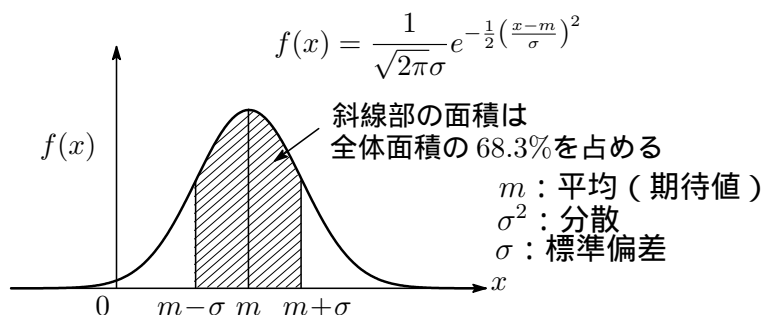


ところで n を増やしていくとどうなるか。当たり前ですが実現回数 k も増加するし, 非実現回数 $n - k$ も増加します。いま $n \rightarrow \infty, k \rightarrow \infty, n - k \rightarrow \infty$ と仮定し, 加えて

$$\frac{(k - p)^2}{n^2} \rightarrow 0 \quad (n \rightarrow \infty)$$

を仮定すると 2 項分布は正規分布で近似できます²。

$$\binom{n}{k} p^k q^{n-k} \simeq \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(k-m)^2}{2\sigma^2}} \quad (\sigma = \sqrt{npq}, m = np) \quad (2)$$



・期待値と分散: 確率変数 X^3 が x_1, x_2, \dots, x_n の値をとり, それぞれの確率が

$$P(X = x_k) = p_k, \quad \sum_{i=k}^n p_k = 1$$

で与えられるとします。これらの試行によって得られる X の期待値 (Expected value) は

$$E(X) = \langle X \rangle = \sum_{k=1}^n x_k p_k \quad (3)$$

²De Moivre・Laplace の定理と呼ばれる。

³例えばコインの表の出る回数など

で与えられますが、これは平均値に他なりません。2項分布の期待値（平均値）は

$$E(X) = \sum_{k=0}^n k p_k = \sum_{k=0}^n k \binom{n}{k} p^k q^{n-k} = np \quad (4)$$

平均値 $E(X)$ と個々のデータ x_k の差の2乗の平均を分散（Variance）といいます。要するに平均値まわりのデータのバラツキの大きさですね。

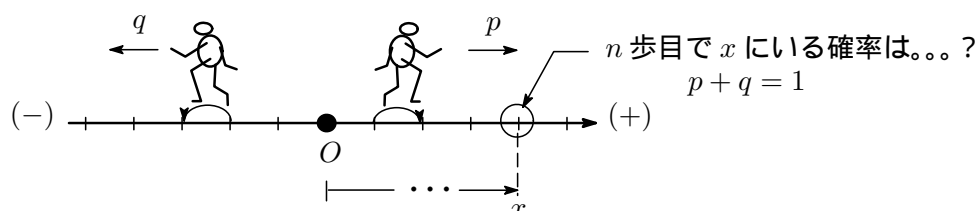
$$V(X) = \sum_{k=0}^n (x_k - E(X))^2 p_k = \sum_{k=1}^n x_k^2 p_k - E^2(X) \quad (5)$$

分散の平方根を標準偏差 σ といい、2項分布の分散と標準偏差は次式で与えられます。

$$V(X) = \sum_{k=0}^n k^2 \binom{n}{k} p^k q^{n-k} - (np)^2 = npq = np(1 - q) \quad (6)$$

$$\sigma = \sqrt{V(x)} = \sqrt{np(1 - q)} \quad (7)$$

- ・ 1次元ランダムウォーク：原点から歩みはじめた酔歩者は右へ左へと行きつ戻りつ1次元数直線上をランダムに歩行しています。 x の正（右）の方向に +1 歩進む確率を p ，負（左）の方向に -1 歩進む確率を $q (= 1 - p)$ としたとき、酔歩者が n 歩目で x の位置にいる確率 $P_n(x)$ を求めます。



さて、 n 歩の歩行のうちで右へ +1 歩移動する回数を k 回とすれば、左へ -1 歩移動する回数は $n - k$ 回。いま、酔歩者は n 歩目で位置 x にいるのだから

$$x = 1 \cdot k + (-1) \cdot (n - k) = 2k - n$$

が成り立ちます。これから k と x の関係は

$$k = \frac{n + x}{2} \quad (8)$$

が成立しますね。つまり、 n 歩の歩行のうち、右へ +1 歩の歩行が k 回あれば酔歩者は x の位置にいるということですね。したがってその確率は (1) より

$$P_n(k) = P_n\left(\frac{n + x}{2}\right) = \binom{n}{\frac{n+x}{2}} p^{(n+x)/2} (1 - p)^{(n-x)/2} \quad (9)$$

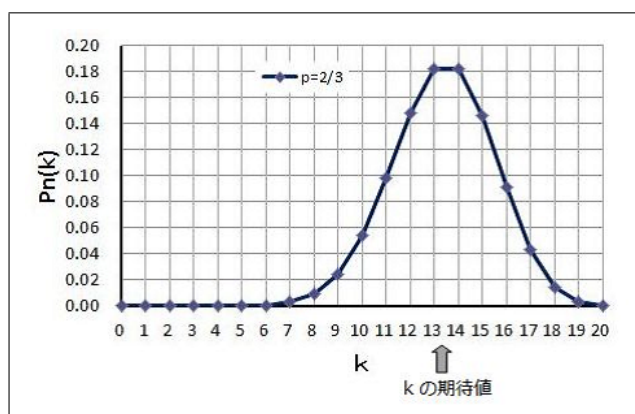
で与えられます。いま、酔歩者は右向きに歩く癖がありその確率を $p = 2/3$ としたとき、 $n = 20$ 歩で x にいる確率を求めてみましょう。 x は (8) より

$$x = 2(k - 10) \quad (10)$$

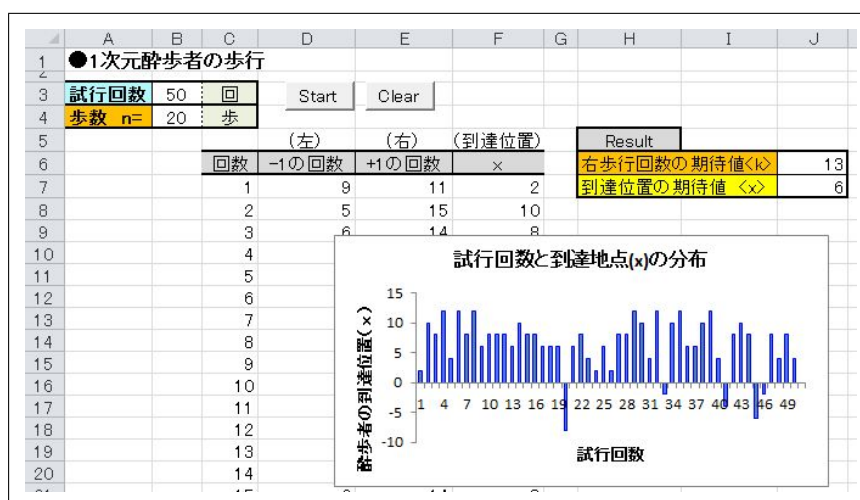
となるので偶数の整数値をとることが分かりますね。求める確率 $P_{20}(k)$ の計算結果は下の通りとなります。

k	x	$P_n(k)$	k	x	$P_n(k)$
0	-20	0.000	11	2	0.099
1	-18	0.000	12	4	0.148
2	-16	0.000	13	6	0.182
3	-14	0.000	14	8	0.182
4	-12	0.000	15	10	0.146
5	-10	0.000	16	12	0.091
6	-8	0.001	17	14	0.043
7	-6	0.003	18	16	0.014
8	-4	0.009	19	18	0.003
9	-2	0.025	20	20	0.000
10	0	0.054			

期待値 $E(X = k)$ は (9) より $E(X = k) = np = 20 \times \frac{2}{3} \doteq 13$ 。(10) より酔歩者は $x = 6$ の位置にいると予想でき、その確率は $P_{20}(13) = 0.182$ となります。仮に 50 歩目での期待値は $np = 50 \times 2/3 \doteq 33$, $x = 2 \times 33 - 50 = 16$ ということになりね。



ところで、ベルヌイ試行の(注)のところで述べたように、上の期待値は大勢の互いに見知らぬ酔歩者がそれぞれ単独に数直線上を 20 歩ランダムウォークした多数回の試行結果の平均値であって、個々の酔歩者がランダムウォークして $x = 6$ の位置に達したというのではないですね。標準偏差 $\sigma = \sqrt{np(1-q)} = \sqrt{20 \times \frac{2}{3} \times \frac{1}{3}} \simeq 2.1$ のバラツキがあります。このあたりのことを VBA でシミュレーションしてみました(Excel シートの「期待値」参照)。試行回数や歩数に適当な値を入力してその様子を確認してください。



<ソースコード>

```

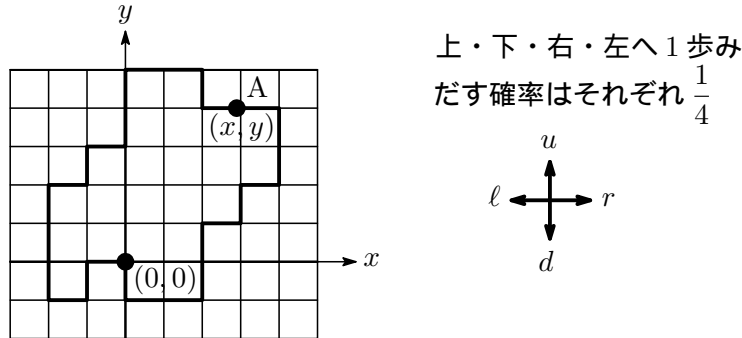
Option Explicit
Public cnt As Integer
Sub RandomWalk_1d()
    Dim i As Integer, j As Integer, rd As Integer
    Dim k As Integer, n As Integer, x As Integer
    Dim S As Single, Ave_k As Single, Ave_x As Single
    Dim V As Single, Vx As Single
    cnt = Range("B3").Value          ' 試行回数 (酔歩者の人数)
    n = Range("B4").Value            ' 既定歩数
    S = 0
    For j = 1 To cnt
        k = 0: x = 0
        Cells(6 + j, 3).Value = j
        For i = 1 To n
            rd = Application.WorksheetFunction.RandBetween(1, 3) ' 1,2,3の整数生成乱数
            If rd >= 2 Then k = k + 1 ' 右に一步の確率は2/3
            Cells(6 + j, 5).Value = k ' 右に一步の回数
            Cells(6 + j, 4).Value = n - k ' 左に一步の回数
        Next i
        x = 2 * k - n
        Cells(6 + j, 6).Value = x    ' 数直線上の到達位置
    Next j
    Ave_k = Round(WorksheetFunction.Sum(Range(Cells(7, 5), Cells(6 + cnt, 5))) / cnt)
    Range("J6").Value = Ave_k       ' 期待値 (平均値)
    Ave_x = Round(WorksheetFunction.Sum(Range(Cells(7, 6), Cells(6 + cnt, 6))) / cnt)
    Range("J7").Value = Ave_x       ' 到達位置の期待値 (平均値)
' グラフ表示-----
With ActiveSheet.Shapes.AddChart.Chart
    .ChartType = xlColumnClustered ' 集合縦棒
    .SetSourceData Range(Cells(7, 6), Cells(7 + cnt, 6)) ' 作図範囲
    .Axes(xlValue).HasMajorGridlines = False ' Falseで補助線消去
    .Axes(xlCategory).HasMajorGridlines = False
    .HasLegend = False ' 凡例なし
    .SeriesCollection(1).Format.Line.ForeColor.RGB = RGB(0, 0, 255)
    .HasTitle = True
    .ChartTitle.Text = "試行回数と到達地点 (x) の分布" ' グラフのタイトル
    With .ChartTitle.Format.TextFrame2.TextRange.Font
        .Size = 12
        .Fill.ForeColor.ObjectThemeColor = msoThemeColorDark1
    End With
    With .Axes(xlValue) ' 横軸のラベル
        .HasTitle = True
        .AxisTitle.Text = "酔歩者の到達位置 ( x )"
    End With
    With .Axes(xlCategory) ' 縦軸のラベル
        .HasTitle = True
        .AxisTitle.Text = "試行回数"
    End With
End With
End Sub
' 計算結果とグラフの消去-----
Sub Erase_Click()
    Dim i As Long
    Range(Cells(7, 3), Cells(7 + cnt, 6)).ClearContents ' セルに格納したデータを消去
    Range("J6:J7").ClearContents
    With ActiveSheet ' 複数の埋め込みグラフを消去
        For i = .ChartObjects.Count To 1 Step -1
            .ChartObjects(i).Delete
        Next i
    End With
End Sub

```

B) 2次元ランダムウォーク

最後に2次元ランダムウォークについて簡単に触れておきます。

- 格子点 A に達する確率：原点からスタートして n 歩目で平面格子上の点 $A(x, y)$ にいる確率を $P_n(x, y)$ とし、この確率を求めます。



酔歩者は右に n_r 歩，左に n_l 歩，上に n_u 歩，下に n_d 歩だけ動いて A に達したとすると

$$\begin{cases} x = n_r - n_l \\ y = n_u - n_d \\ n = n_r + n_l + n_u + n_d \end{cases} \quad (11)$$

が成り立ちます。これを満たす n_r, n_l, n_u, n_d であればどんな正の整数でもいいわけで，図に示すような様々なルートが考えられます。このようなルートの総数は，最初の「最短経路の数」のところで示したように

$$\frac{n_r + n_l + n_u + n_d}{n_r! n_l! n_u! n_d!} = \frac{n!}{n_r! n_l! n_u! n_d!} = \binom{n}{n_r n_l n_u n_d} \text{通り} \quad (12)$$

あります⁴。上下左右いずれの方向に一步進むかは $1/4$ の確率なので，原点から n ステップで A に達する各ルートの確率はいずれも

$$\frac{1}{4} \times \frac{1}{4} \times \cdots \times \frac{1}{4} = \left(\frac{1}{4}\right)^n \quad (13)$$

$P_n(x, y)$ は (12) で得られた各ルートに (13) を掛けた確率の和で表されるので

$$P_n(x, y) = \sum_{n_r, n_l, n_u, n_d} \left(\frac{1}{4}\right)^n \binom{n}{n_r n_l n_u n_d} \quad (14)$$

となります。和は (11) を満たすような n_r, n_l, n_u, n_d についてとっています。ちょっと話が抽象的なので，具体的に6歩ランダムウォークして元の原点 $(0, 0)$ に戻ってくる確率 $P_6(0, 0)$ を求めてみましょう。(11) で $x=y=0$ とおいて

$$\begin{cases} 0 = n_r - n_l & \rightarrow n_r = n_l \\ 0 = n_u - n_d & \rightarrow n_u = n_d \\ 6 = n_r + n_l + n_u + n_d & \rightarrow n_r + n_u = 3 \end{cases}$$

⁴右辺の括弧は多項係数と呼ばれる。

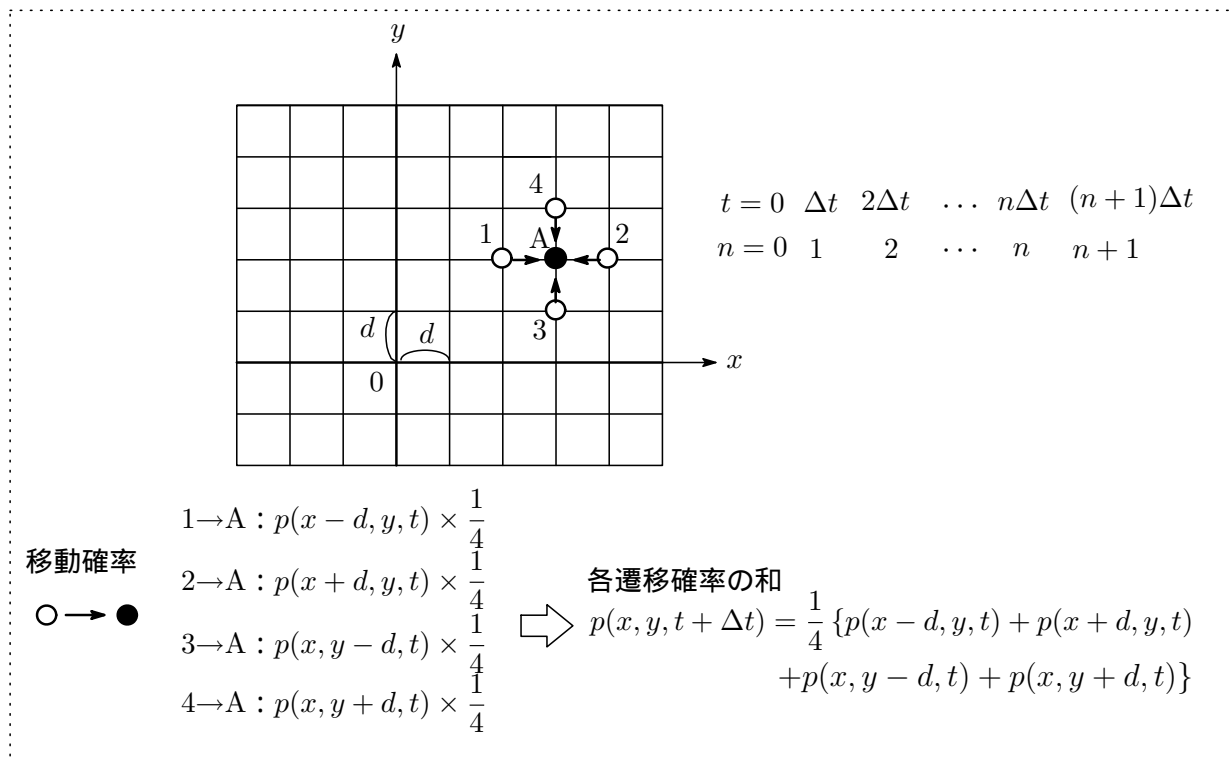
が成り立つ n_r, n_ℓ, n_u, n_d を求めると次の4つのケースしかないことがわかります。

n_r	n_ℓ	n_u	n_d	n
0	0	3	3	6
1	1	2	2	6
2	2	1	1	6
3	3	0	0	6

したがって求める確率は (14) より

$$P_6(0,0) = \left(\frac{1}{4}\right)^6 \left\{ \binom{6}{0033} + \binom{6}{1122} + \binom{6}{2211} + \binom{6}{3300} \right\} = \frac{25}{256} \doteq 0.098$$

- ・ 拡散方程式：2次元ランダムウォークは2次元拡散現象と物理的に等価な現象⁵であることを見ていきます。さて、マス目サイズが d の正方形格子を粒子が上下左右ランダムに運動しています。 $t=0$ で原点にいた粒子が時刻 $t_{n+1} (= (n+1)\Delta t)$ で (x, y) にいたとしましょう。粒子が $n+1$ ステップ目 (時刻 $t + \Delta t$) で格子座標 $A(x, y)$ にいるためには、1



ステップ前の $n(t)$ では隣接する4点の座標 $(x-d, y)$, $(x+d, y)$, $(x, y-d)$, $(x, y+d)$ のいずれかにいなければなりませんね。粒子がこのいずれかの座標から座標 $A(x, y)$ に移動する確率はそれぞれ $1/4$ の等確率なので粒子が A 点にいる確率 $p(x, y, t + \Delta t)$ は

$$p(x, y, t + \Delta t) = \frac{1}{4} \{p(x-d, y, t) + p(x+d, y, t) + p(x, y-d, t) + p(x, y+d, t)\} \quad (15)$$

となります。ここで両辺から $p(x, y, t)$ を差し引いて並べ替えると

$$p(x, y, t + \Delta t) - p(x, y, t) = \frac{1}{4} \{p(x+d, y, t) - 2p(x, y, t) + p(x-d, y, t) + p(x, y+d, t) - 2p(x, y, t) + p(x, y-d, t)\} \quad (16)$$

⁵なに□も2次元ランダムウォークだけに限りません(念の為)。

となります。格子のマス目サイズ d を微小量とすれば，上式に次の差分近似式

$$p(x, y, t + \Delta t) - p(x, y, t) \simeq \frac{\partial p}{\partial t} \Delta t$$

$$p(x + d, y, t) - 2p(x, y, t) + p(x - d, y, t) \simeq \frac{\partial^2 p}{\partial x^2} d^2$$

$$p(x, y + d, t) - 2p(x, y, t) + p(x, y - d, t) \simeq \frac{\partial^2 p}{\partial y^2} d^2$$

が適用でき，それを実行すると次の2次元拡散方程式⁶が得られます。

$$\frac{\partial p}{\partial t} = D \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right), \quad D = \frac{d^2}{4\Delta t} \quad (17)$$

D は拡散係数と呼ばれ，拡散の速さを表す比例定数で $[L^2T^{-1}]$ の次元を持ちます。

2020.8.23 猛暑 by KENZOU

⁶1次元拡散方程式の数値解は HP 数学のコーナーの「Excel VBA で数値解析」を参照してください。