ウイルス感染のシミュレーション

事の発端は 2019 年 11 月、中国武漢で原因不明の「ウイルス性肺炎」の患者が見つかったことからでした。その後、このウイルスは世界中に広がり始め、急性呼吸器疾患(COVID-19)を起こすこの新型ウイルスと人類との戦いが始まります。

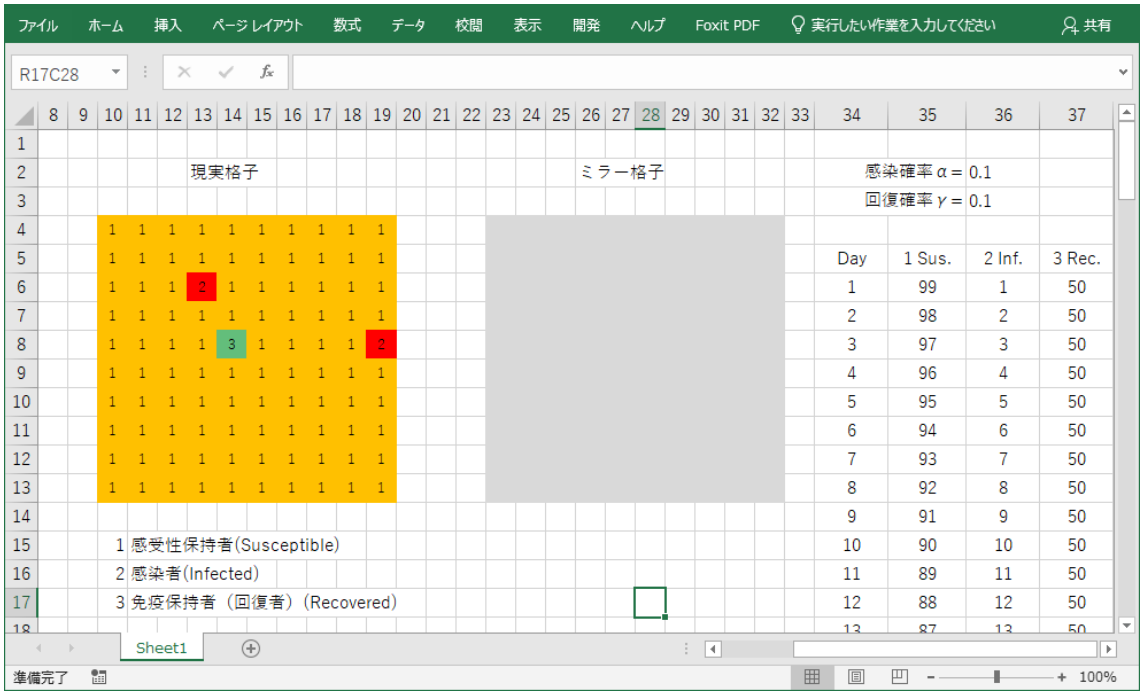
コロナウイルス SARS-CoV-2 と呼ばれるようになった新型ウイルスに対して、世界中でワクチンの開発もが始められました。日本でも始まっていますが,このテキストを書いている時点では、まだ先が見通せないようです。

一方、感染をコンピューターでシミュレーションし、その結果を使いながらの対策を進める自治体が現れました。第一波の時には、感染爆発を防ぐため国は非常事態宣言を発令し、人との接触を 8 割下げることを訴えました。人との接触を避けるこの 8 割という数値は、このコンピューターによるシミュレーション計算から予測されたものでした。

国の内閣官房では COVID-19 AI シミュレーションプロジェクトをスタートさせ、コロナウイルスに対するシミュレーション開発の研究テーマの公募しています。　現在、多くの研究が寄せられ、その中にはこの章で取り上げる「**モンテカルロ法による感染シミュレーション**」もあります。

この章では、ウイルス感染の**格子型モデル**を作り、感染率を変えることでどのような変化が起こるのかという、コンピューターを使ったシミュレーション実験のためのベースを作ってみたいと思います。コンピューターを使って実験する方法を学んでいきましょう。

※　COVID-19 AI シミュレーションプロジェクト　(https://www.covid19-ai.jp/ja-jp/simulation)



まず導入モデルとして 10×10 の 100 人の空間を考える。ここに一人の感染者が現れたらどのようになるのか。それをシミュレーションすることから始めよう。最終的に 100×100 の 10000 人の空間といったより現実に近い形でのシミュレーションに挑んで研究する人が出てくるのを期待している。

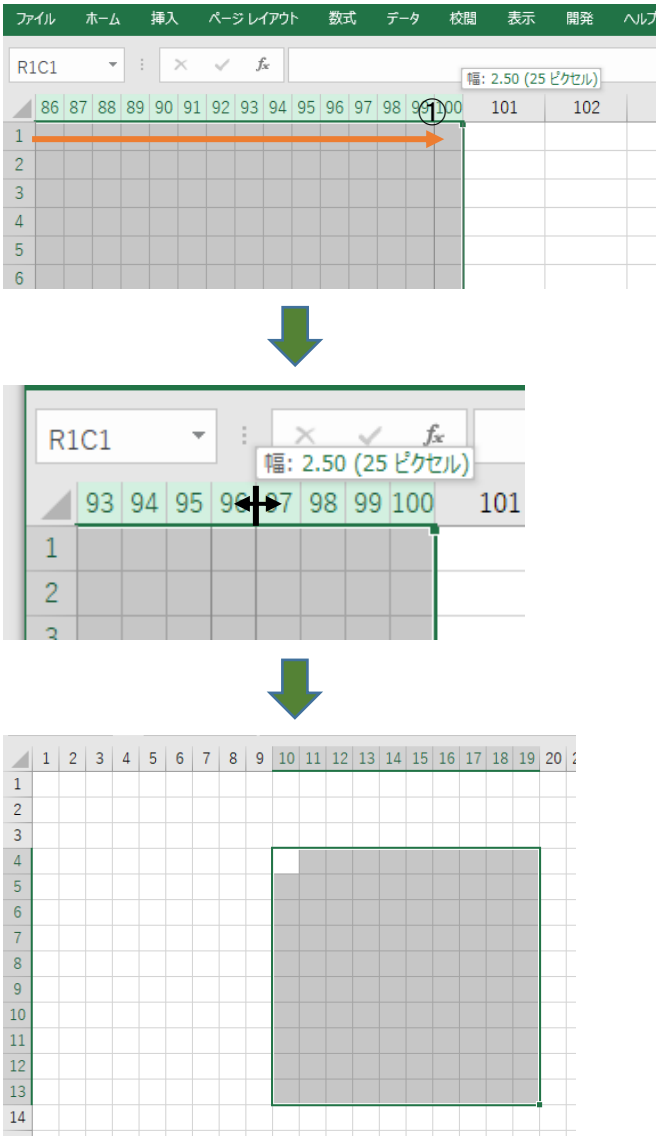


図 4–17

現実空間の設定

新しいエクセルを立ち上げ「**ファイル**」の「**オプション**」→「**数式**」で

□R1C1 参照形式を使用する

にチェックを入れます。

図 4–17 上①のように横の列番号の上にカーソル（矢印）をもっていき左クリックしたまま 100 までドラッグします。次に図 4–1 中②のように適当な列番号の書かれたセルの右辺を左クリックしたまま、縮めて 25p（ピクセル）程度にすると図 4–1 下のようにセルの形が正方形になります。シートに 10×10 の範囲が示されていますね。これに図 4–18 のように 10×10 のエリアの中のセルすべてに数字の 1 を入れます。ついでに中の一つか二つは 2 と 3 にしておきます。

実はこのエリアが、ウイルスが感染する空間になります。10×10 だから 100 人の空間を想定しています。

これを「**現実格子**」と呼ぶことにします。その右側に「**ミラー格子**」と呼ぶ同じ大きさの空間をマウスでドラッグして作ります。ここには何も数字は書かないのですが、図 4–18 のように薄いグレーで色を付けておきましょう。

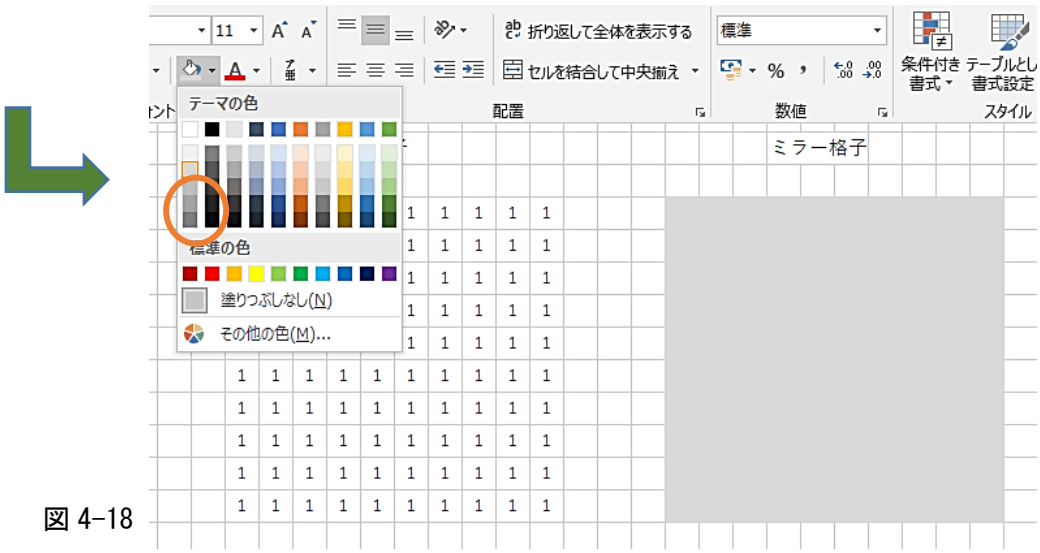


図 4–18

図 4-19

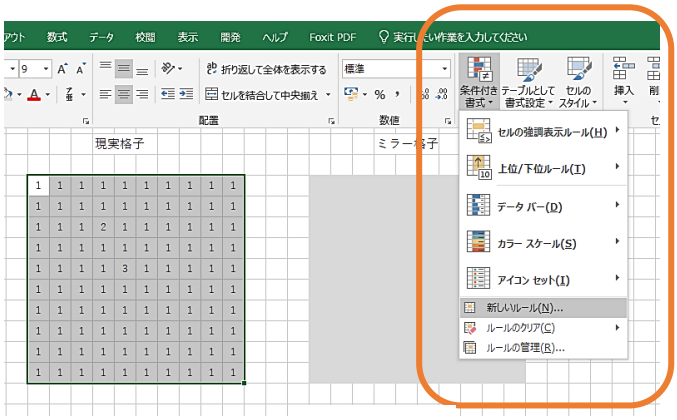


図 4-20

空間の色分け

図 4-19 のようになりましたか？

次に**図 4-20** のように「**現実格子**」全体をドラッグして色を変え、「**ホームタブ**」の中の「**条件付き書式設定**」を選ぶと**図 4-20** のようにメニューが出ます。この中の「**新しいルール**」を選ぶと**図 4-21** のような「**新しい書式ルール**」というダイアログボックスが出てきます。

この中の「**３色スケール**」を選び**図 4-21** の右のように色の設定をします。

実は、この「**現実格子**」の番号は１がまだウイルスに感染していない人、２が感染者、３が感染後回復し抗体を体内に持っている人という設定にしました。

従って、１はちょっと危険信号の**黄色**、感染者は**赤色**、そして回復者は**緑色**といった設定にしてあります。はじめは、このプログラムの作者の意図に沿って言われるとおりにしていってください。

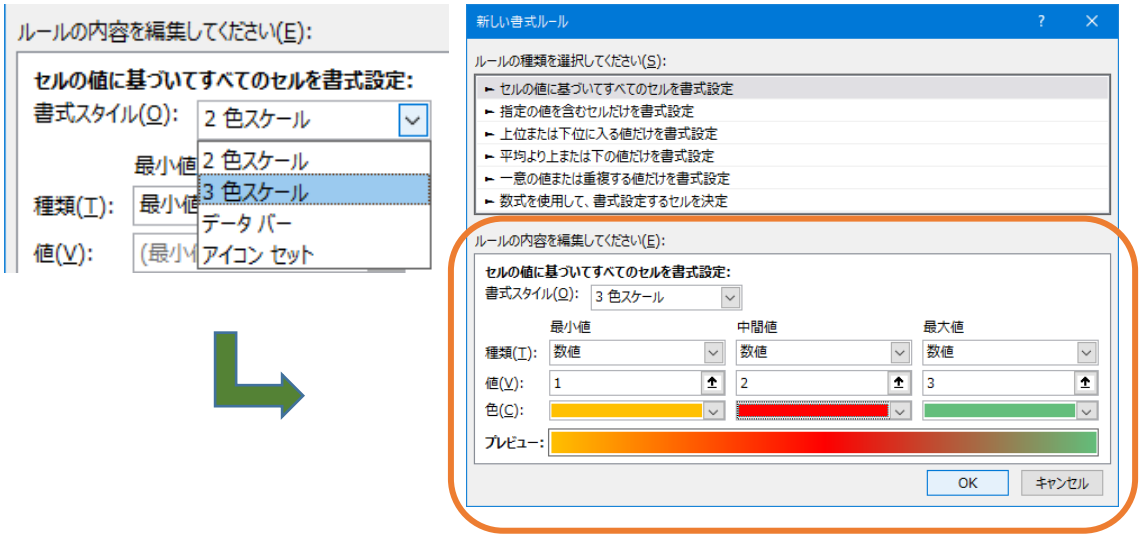


図 4-21

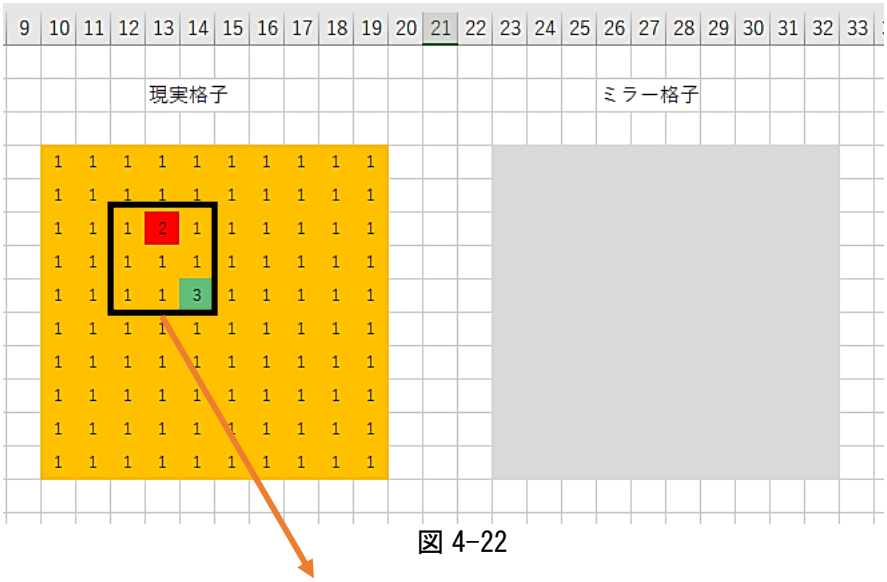


図 4-22

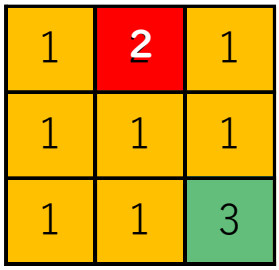


図 4-23

実験ラボの設定

図 4-22 のように「現実格子」の中に色がついて「**感染者**」や「**回復者**」の位置がわかりやすくなりましたね。

ここで少しこの「**現実格子**」の説明をします。**図 4-23** を見てください。**図 4-23** で示された真ん中のセル（7,13）は 1 と書かれていますね。つまり、ウイルス感染の可能性のある人でまだ幸運なことに感染していない人です。この人の名前を A 君としておきます。

この人の周りには、8 人の人がいます。この人たちは、1 日で接触する物理的に距離が近い人だとします。

A 君の上の人が、感染していますね。この人を B 君とします。このとき B 君から A 君がウイルス感染する確率を「**感染確率** α」と呼ぶことにします。

例えば「**感染確率**」が $\alpha=0.1$ の場合、10 日間同じような接触が続くと感染する可能性が 100%あるということにします。つまり、感染確率は 1 日当たり 0.1 の確率になります。

この感染確率での感染可能性を RND()を使って、コンピューターの中でサイコロを振ります。例えば 0～1 までの間の数値を出すサイコロを振って　もし　RND()＜0.1 だったら、セル（13，14）の人の値は 2（**感染**）となることにします。確率 10%（0.1）ですね。

そうじゃなかったらセル（13，14）の A 君の値をそのままとします。

また、一度感染しても回復し抗体を持った人を 3 とします。これを**回復確率** γ〈ガンマ〉と呼ぶことにします。$\gamma=0.1$ というのは、平均 10 日程度で回復するというわけです。回復して 3 になった人は、ウイルスを持たず他の人にウイルスを感染させないとしておきます。

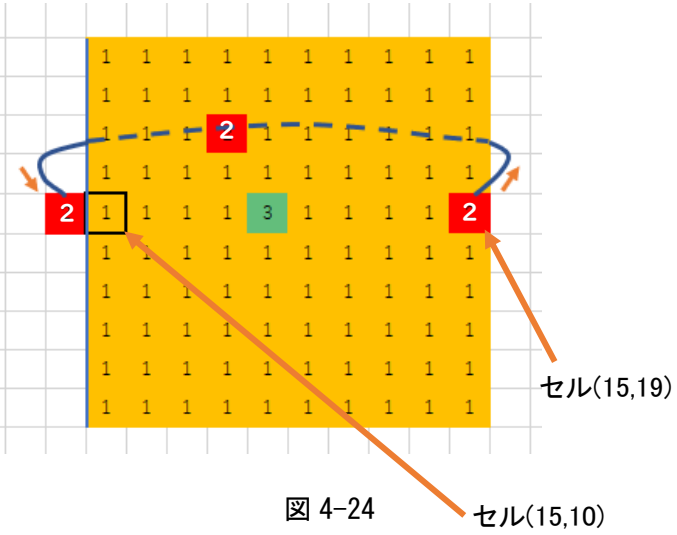


図 4-24

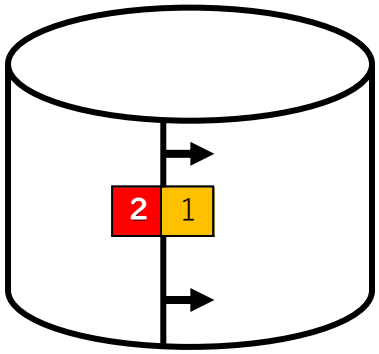


図 4-25

	1	1	1	1	1	1	1	1	1	3	
2	2	1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	3	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	3	1
	2	1	1	1	1	1	1	1	1	2	

図 4-26

端っこの人は？

この**格子モデル**だと、端っこにいる人は周りにいる人数が少ないですね。このような場合、すべて同じ条件のするために、この空間は閉じているように設定します。

どういうことかというと、**図 4-24** の上のように**セル(15,10)**の人の左は誰もいませんが、**セル(15,19)**の人がいるとするわけです。つまり、**図 4-24** の下の円筒の図 4-25 のように、この「**現実格子**」はくるっと回っているという風にしておきます。

同様に上下も、一番上と一番下はつながっているとします。不思議な空間で作図することはできませんが頭の中で想像することはできるかもしれません。「**風呂敷**」で何かを包むように端っこが出会っているわけです。

すると、すべての人が周りに８人いる同じ環境になります。

これはこのような「**仮想空間**」を作るときによくやられる方法です。

この「**風呂敷**」空間で考えると**図 4-26** のようになります。**図 4-26** は「**現実格子**」の外側の所がどこのセルの番号に対応しているか示したものです。これまで説明してきたことが分かっていれば、簡単にわかりますね。確認してください。

ここで問題が出てきます。**図 4-26** の「**現実格子**」の角はどこと対応しているのでしょうか。つまり、**図 4-26** の□の中は何と書けばいいのでしょうか。

少し仲間と確認しあいましょう。

「**風呂敷包み込み空間**」と考えればすぐわかります。答えは、次のページの下に書いてありますので、仲間と答えを見つけたら確認してください。答えを先に見てはいけません。そんなことでは、プログラムは書けませんよ。

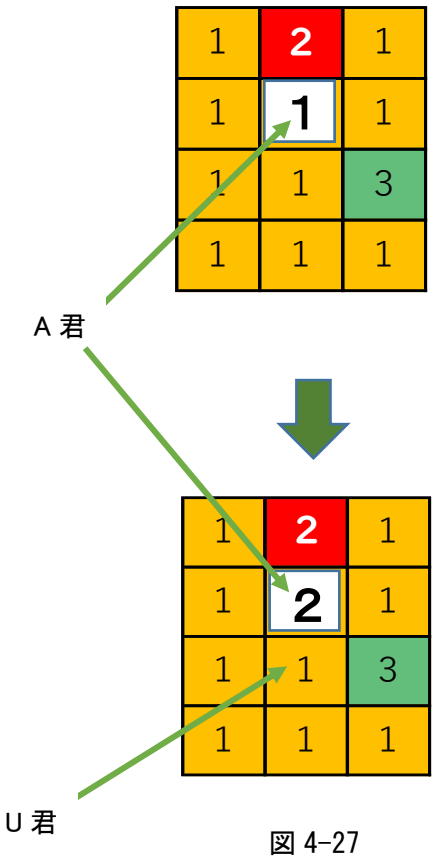


図 4–27

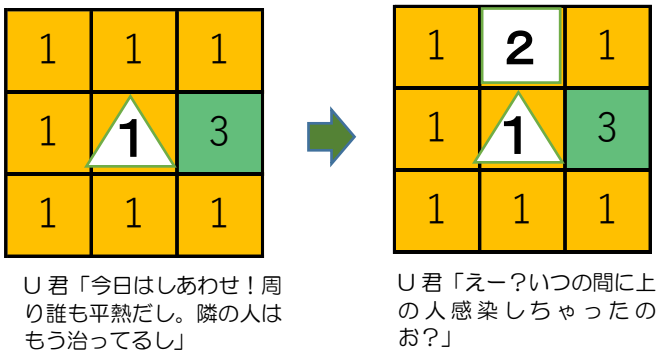


図 4–28

ミラー格子って？

それでは「**ミラー格子**」ってなんでしょう。実は、「**現実格子**」で一人一人の感染や回復を判定していくとき、感染した場合は「2」となるわけですが、それをすぐに書き換えるとおかしなことになります。

ある日の感染状態の環境は、100 人の人すべてに同時に与えられます。ところが、この 100 人の一人一人を、感染したから「2」と書き換えてしまうと、1 日単位ではない感染の波が、判定する順番で起こっていってしまいます。

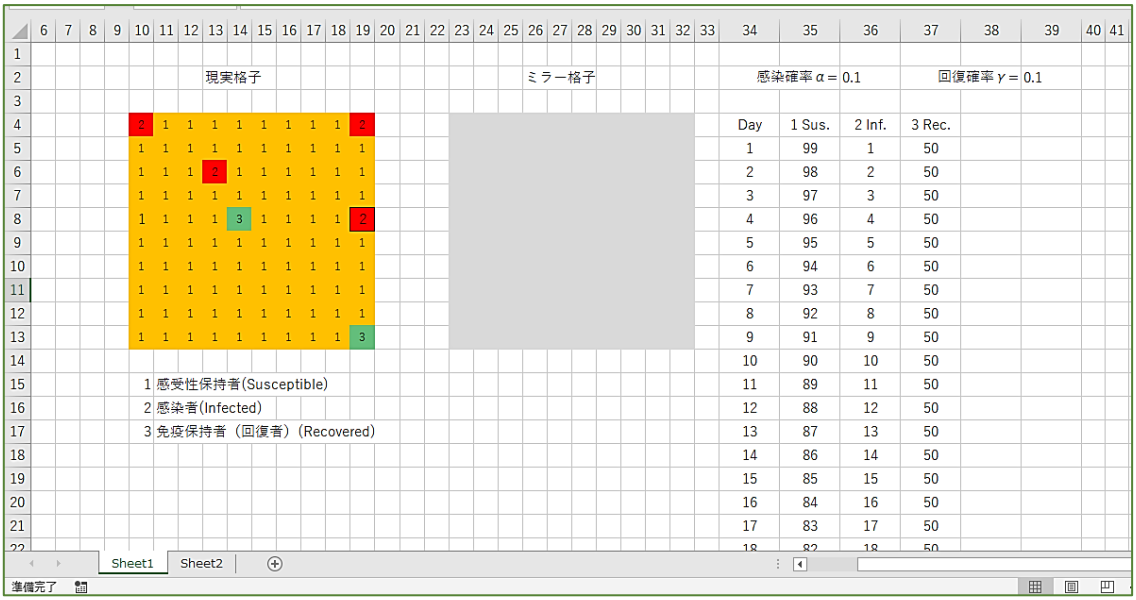
例えば、図 4–27 の上を見てください。真ん中の A 君が感染したかどうかを判定していっている途中で、感染してしまったから「2」と書き換えてたとします。でもそれは本当は明日のことです。

するとそのあとに判定の順番が来た U 君の環境は、はじめと違ってしまいます。なんと突然、上に感染者が現れています。もともと U 君の環境は図 4–28 左側のようになってたはずです。感染の可能性はなかったのです。

このおかしさを取り除くため、まず**ミラー格子**に A 君が感染したことを写し取とり、「**現実格子**」はそのままにしておきます。。

そして、100 人の判定が済んで、すべての人たちの明日の運命が「**ミラー格子**」に記録されたら、それを丸ごと、**現実格子**に張りつけるのです。これで次の日の状況が初めてわかります。そして、また、左上のセルから順番に感染状況を判定して、それをミラー格子に写していきます。

これらの判定作業はすべてプログラム上で行われるので、「**ミラー格子**」にはプログラムの進行とともにその場所の感染状況の判定数字が書かれていきます。



列の 34,35,36,37,38,39 の幅を広げました。「感染確率 α=」の文字はセル(2,34)とセル(2,35)をドラッグしホームタブの「配置」の下図の①→②で右に寄せます。「起伏確率 γ=」も同様の操作です。

図 4-29

3	1	1	1	1	1	1	1	1	1	3	1
2	2	1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	3	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	3	1
2	2	1	1	1	1	1	1	1	1	2	2

前々ページの解答　図の四角の中の数字になります。「風呂敷包み空間」で考えましたか。

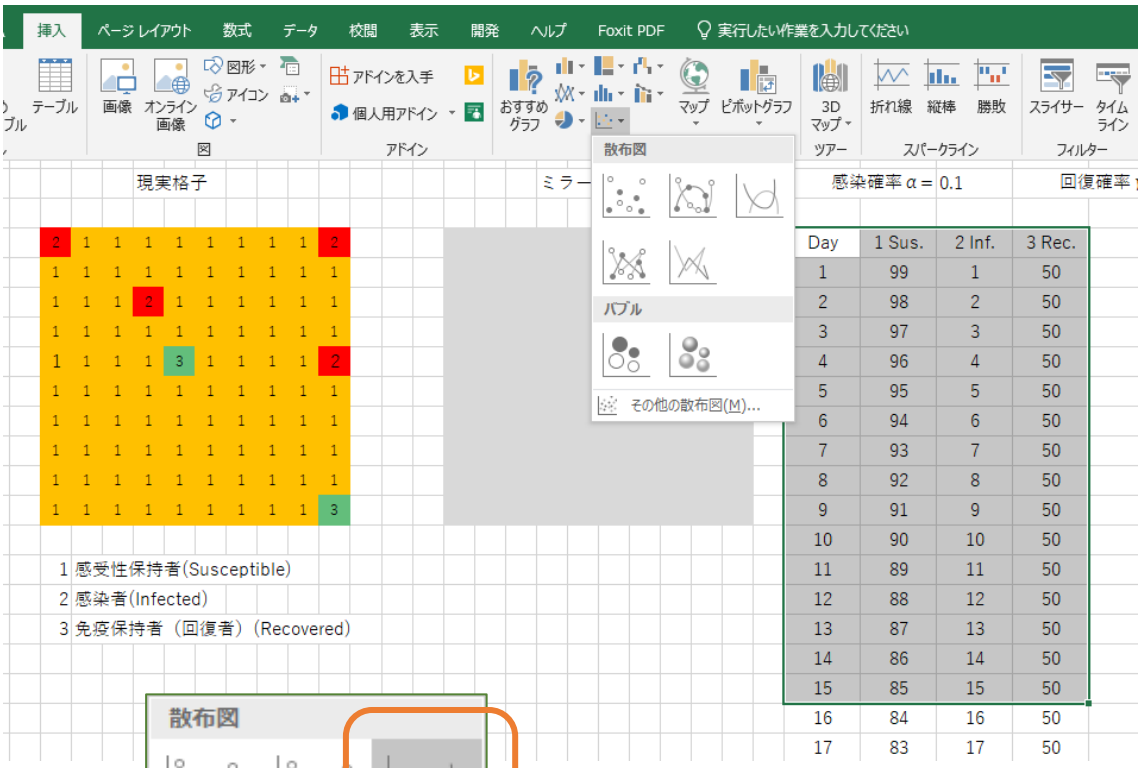
実験ラボ製作 2

再び**エクセルシート 1** の**実験ラボ製作**をやりましょう。

図 4-29 のように「**ミラー格子**」の右側に「**感染者**」が発生してから 1 日ごとの「**感受性保持者**」1 Sus. と「**感染者**」2 Inf.、そして「**免疫保持者**」3 Rec の欄を作り、**日数**(Day)は 100 日程度にして縦に日数以外はダミーの値を入れておきます。これはグラフを作るためです。

また、**図 4-29** をみて「**現実格子**」の下に一行開けて 1，2，3 の数字の意味を書いておきましょう。

できたら、「**ミラー格子**」の横に「**感染確率 α**」と「**回復確率 γ**」の値を**セル**(2,36)と**セル**(2,39)に書いておきます。これはプログラムで使うので位置をしっかり決めておきましょう。値はとりあえずどちらも 0.1 としておきます。



Day	1 Sus.	2 Inf.	3 Rec.
1	99	1	50
2	98	2	50
3	97	3	50
4	96	4	50
5	95	5	50
6	94	6	50
7	93	7	50
8	92	8	50
9	91	9	50
10	90	10	50
11	89	11	50
12	88	12	50
13	87	13	50
14	86	14	50
15	85	15	50
16	84	16	50
17	83	17	50

平滑線

図 4-30

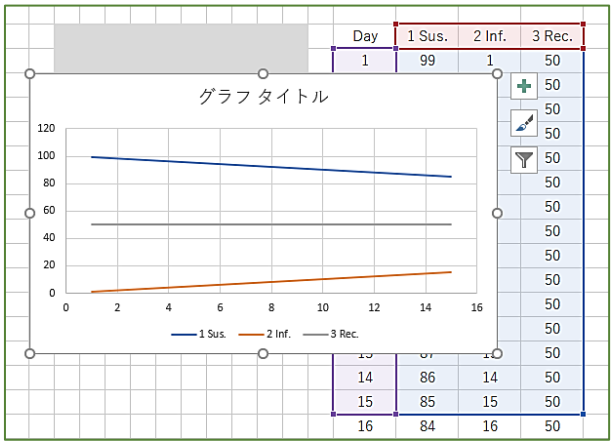


図 4-31

グラフを描く

ダミー（偽物）のデータの入った表を１５日目ぐらいまで**図 4-30** のようにドラッグして色を変えておきます。次に「**挿入**」タブからグラフの「**散布図**」を選び**図 4-31** のように「**平滑線**」をとりあえず選んでおきます。

すると**図 4-31** のような加工前のグラフが現れます。このグラフを加工して次のページにあるようなグラフを作ってください。

横軸は 60 日ほどまで伸びていますが、これは**図 15** の状態で 60 日まで**図 4-31** の下の角をマウスで下に引っ張れば 60 日までグラフが伸びてくれます。

特に「**感受性保持者**」といわれるこれからかかる可能性のある人は「**黄色**」系、感染者は「**赤色**」系の色、免疫保持者（回復者）は「**緑色**」系にしてください。

また、感染者の人数は「**棒グラフ**」で表すことにしましょう。

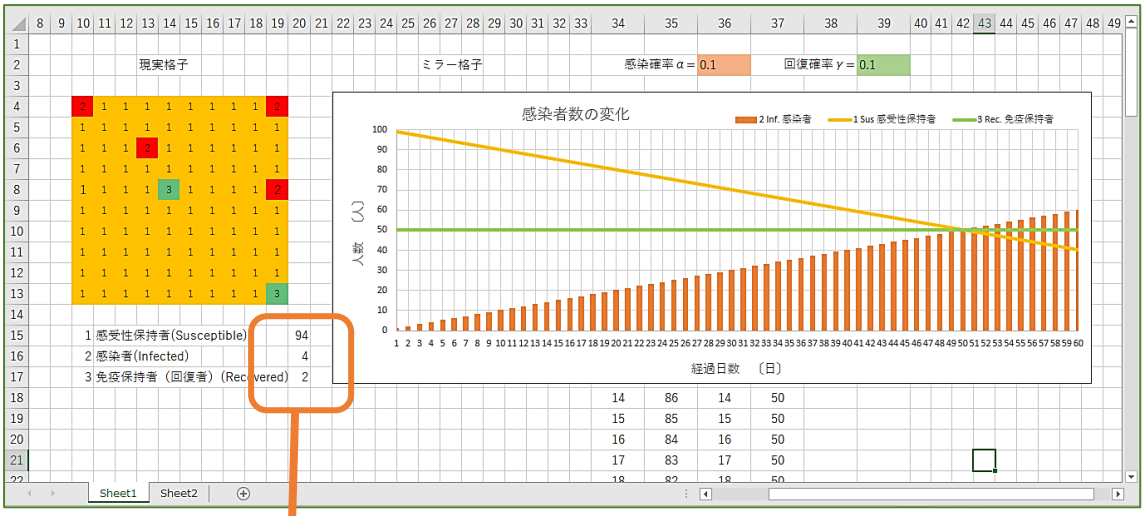
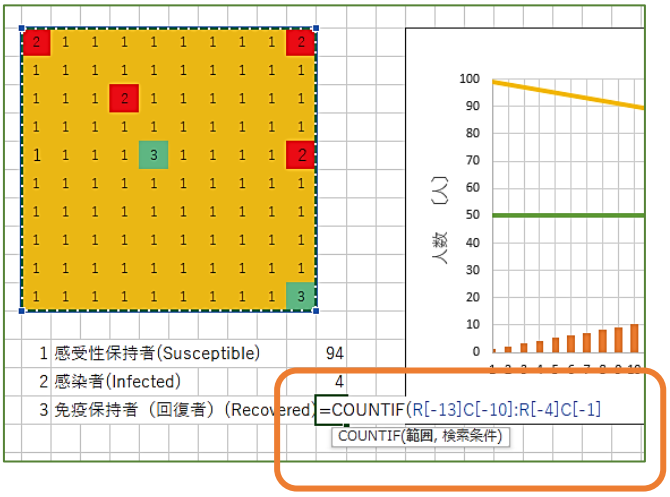
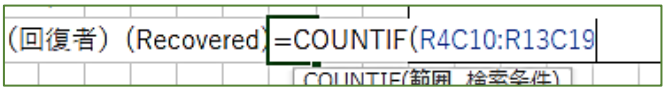


図 4-32

免疫保持者３の人数を
数えさせたいときは…。



ここでファンクションキーF4 を押
して絶対参照にします。



続けて、カウントしたい３を
,3 と打ってかっこ返しします。

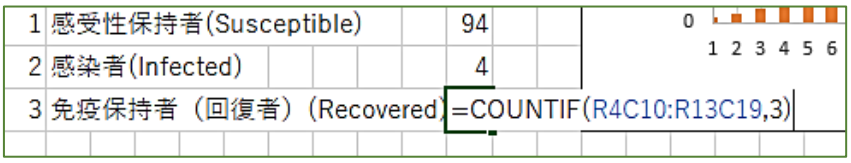


図 4-33

感染者を数える

図 4-32 のようにグラフを変えられましたか？小さな卒業試験ですね。わからない人は、仲間から教えてもらいましょう。

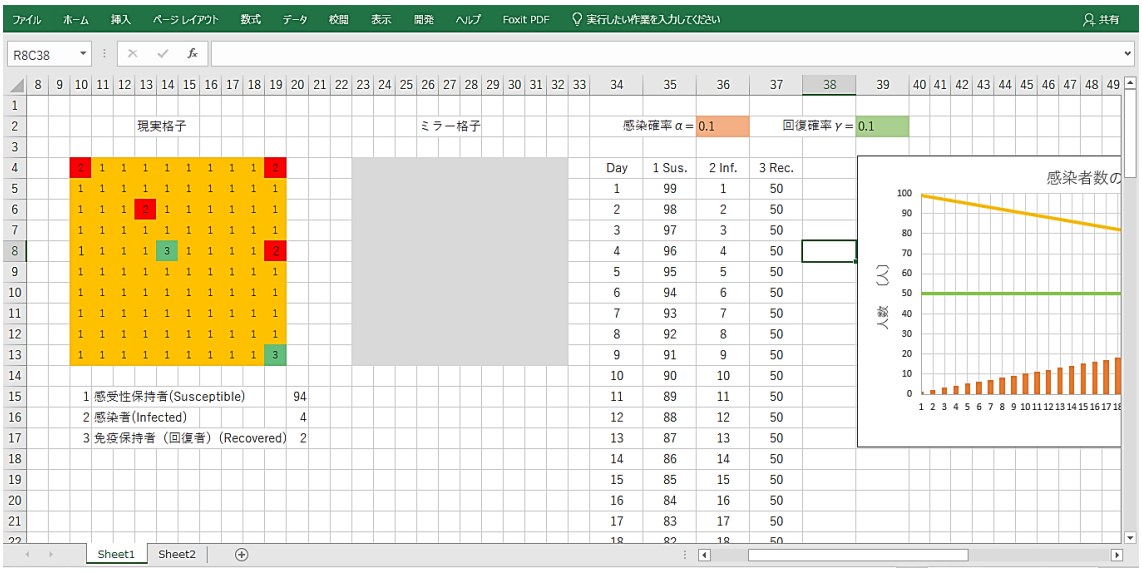
ここで**図 4-32** には「**現実格子**」の中の感染者数などが数字で示されています。この方法は

① 感受性保持者数を出したい**セル(15,20)**をクリック

② =COUNTIF(と書いて現実格子の左上から右下までドラッグ

③ 「F4」を押して「**絶対参照**」にして、

④ 続いて ,1) と書いて Enter すると 94 と「**感受性保持者**」の数が出ます。



プログラム開発中はグラフをちょっと右のほうに
すらしておきます。

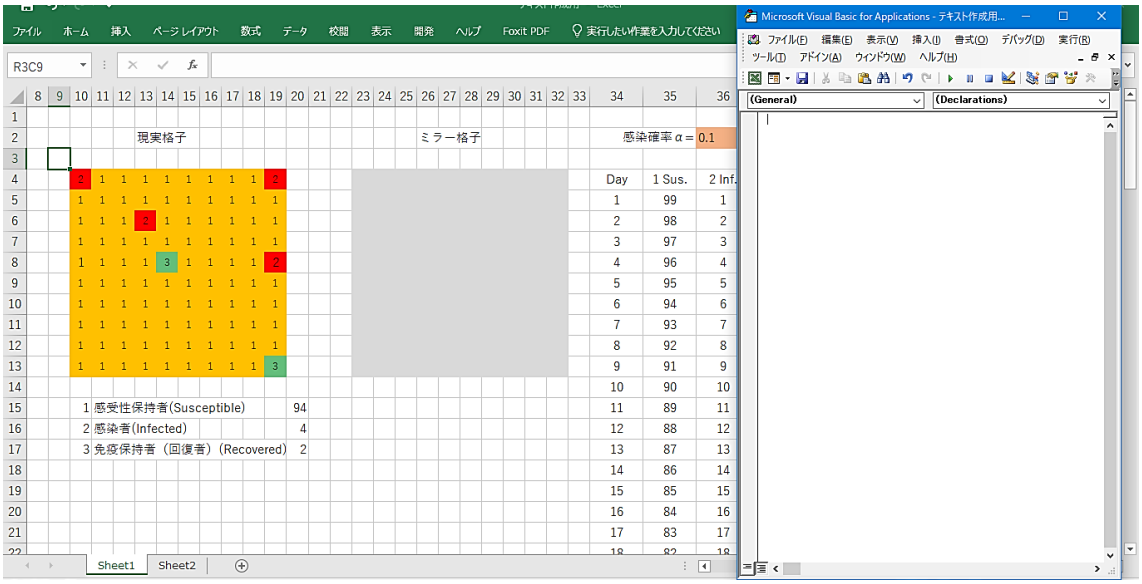


図 4-34

プログラム製作 1

それではプログラムを書いていきましょう。「開発」タブから Visual Basic を選び、プログラムコードを書くためのシートを出します。このとき白いシートが出ていなかったら VBA のウインドウの「**表示**」から「**シート**」を選びます。VBA というのは**マイクロソフト**（ビルゲイツが創設した今のパソコンの基本ソフト『ウインドウズ』や『オフィス（今使ってるやつ）』を開発した巨大ソフトウエア企業）が作った Visual Basic for Applications の略称です。

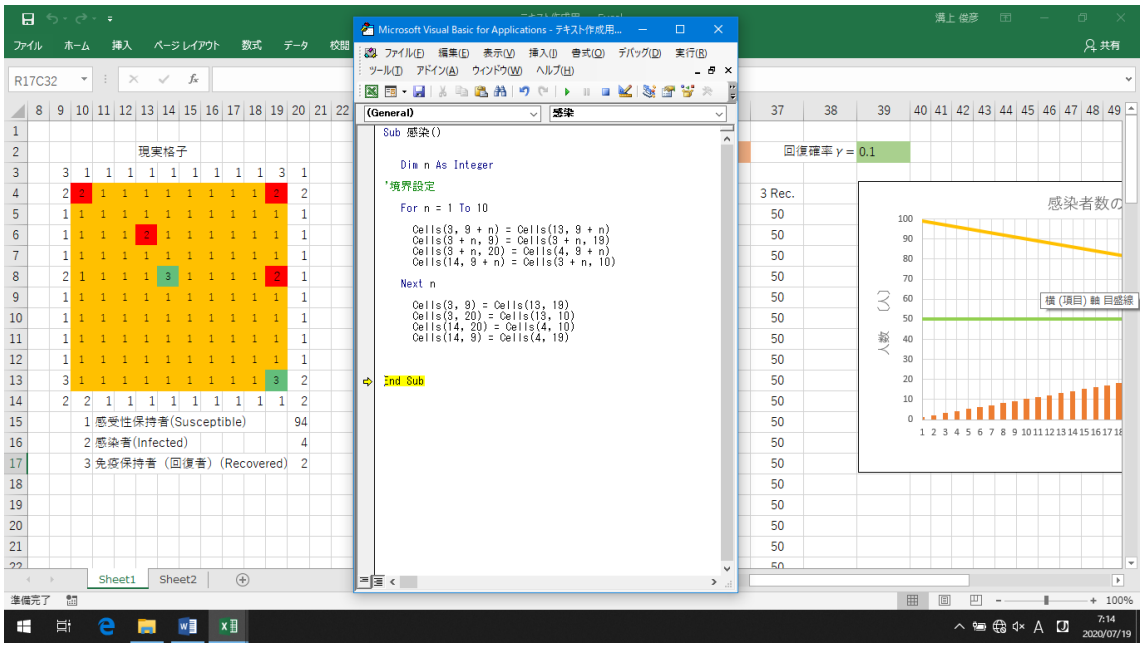


図 4-35

```
Sub 感染()
    Dim n As Integer
' 境界設定
    For n = 1 To 10
        Cells(3, 9 + n) = Cells(13, 9 + n)
        Cells(3 + n, 9) = Cells(3 + n, 19)
        Cells(3 + n, 20) = Cells(4, 9 + n)
        Cells(14, 9 + n) = Cells(4, 9 + n)
    Next n
        Cells(3, 9) = Cells(13, 19)
        Cells(3, 20) = Cells(13, 10)
        Cells(14, 20) = Cells(4, 10)
        Cells(14, 9) = Cells(4, 19)
End Sub
```

まず n=1 としてセル（3,10）にセル（13,10）の値を入れます。次はセル（4,9）にセル(4,19)の値を入れます。・・・という具合にしておくと、次 n=2 のときはその隣のことになり n=3 の時はまたその隣のという具合に次々と値が書かれることになります。

これを描いたら、F8 を押しながら「現実格子」の周りに値が図 19 のように書かれていくか確認してください。

境界設定

「**現実格子**」は風呂敷包み空間のように閉じているのでしたね。

そこで境界にいる人たちの周りにいる人たちの感染状況を、プログラムを使って書けるようにします。

問題 4-2　「現実格子」の境界の周りがどことつなかっているかプログラムを使って図 4-35 のように書きだしなさい。
ただし　For n=1 To 10 ～　Next n を利用すること。

こんな時も「自分が作るならこんな風にするな」という視点を忘れないようにしてください。自分がこんな風にできたらいいなとイメージして「**自分で考える**」のがプログラミングです。自分が解決のための「**問いを作り**」それを「**自分で解く**」のです。

' 感染判定プログラムコードの構造

基本構造１

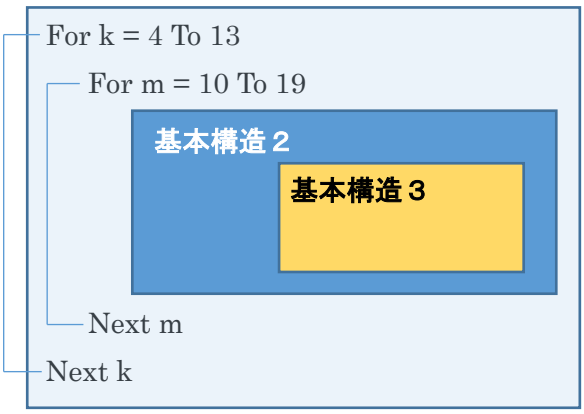


図 4-36

例

```
For k = 4 To 13
    For m = 10 To 19
        Cells(k,m+13) = Cells(k,m).Value
    Next m
Next k
```

図 4-37

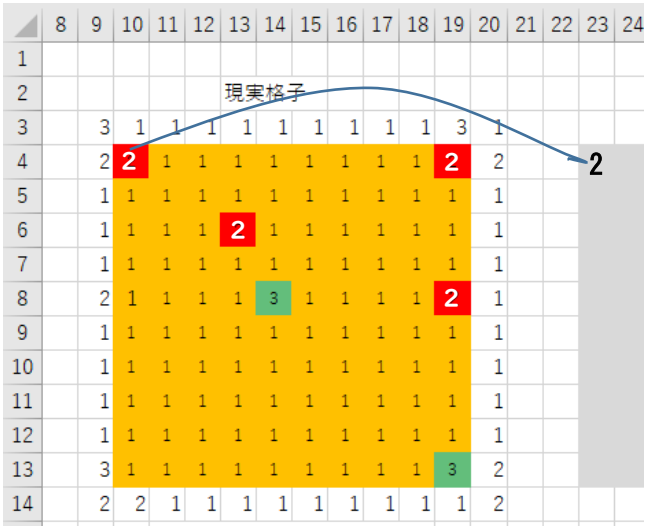


図 4-38

感染判定１

それでは、それぞれ１００人の人たちが感染しているかどうかを判定するプログラムはどのように書けばいいのでしょうか。これが今回のメインのプログラムです。

しかし、これは「難しい」作業です。作者（Mizo-T）は授業の合間と土日を使って一週間かかりました。でも楽しい日々でした。ずっと考えているとあるとき「あっ！こうすればいいや」というアイデアが浮かんでくるのです。

でも君たちに課題として出しても宿題や課題に追われる君たちにとって、できる人でもかなり時間がかかるのではないでしょうか。

今回は、この「**感染判定**」の作者が作ったコードを「**読み解いていく**」ことを課題としましょう。

まず基本的な構造は３つしかありません。

大きな**基本構造１**の中に**基本構造２**があり、その**基本構造２**の中に**基本構造３**があります。

まず**基本構造１**ですが、**図 4-36** のようになっています。k の値が 4 の時 m を 10 として**基本構造２**の仕事をした後、m を一つずつ増やして m=19 まで同じことをします。このときずっと k=4 です。次に k=5 としてまた m を 10 から 19 になるまで**基本構造２**に書かれたことをやるわけです。これは k=13 になるまで続きます。この操作は、Cells(k,m) を考えると**図 4-38** の「**現実格子**」の左上のセルからはじめて横に右端のセルまできたら次、左上の 2 番目のセルから右端までのセルと、すべてのセルに順番に一つずつ同じ作業をやっていることになるのです。

ちなみに**図 4-37** の例だと、現実格子のすべてのセルの値（Value）をミラー格子に鏡のようにすべて映すプログラムになります。

基本構造２

```
Select Case Cells(k, m).Value
    Case Is = 3
        Cells(k, m + 13) = 3
    Case Is = 2
        If Rnd() <= γ Then
            Cells(k, m + 13) = 3
        Else
            Cells(k, m + 13) = 2
        End If
    Case Is = 1
        基本構造３
    Select Case Cells(k, m + 13).Value
            Case Is = 2
            Case Is = 3
        Case Else
            Cells(k, m + 13).Value = 1
        End Select
    Case Else
End Select
```

日本語訳

セル(k,m)のことなんだけど

セル(k,m)の値が３だったら、もう回復して抗体持ってるから、ミラー格子のセル(k,m+13)の値を３と書いといてくれ。

セル(k,m)の値が２だったら、回復する可能性があるかどうかサイコロを振って、0.1 以下の値だったら回復ってことでミラー格子のセル(k,m+13)の値を３にしといてくれ。
サイコロの値がそれ以外だったら、まだ２だからミラー格子のセル(k,m+13)の値を ２にしといてくれ。

セル(k,m)の値が１だったら、ちょっと周りの８人について調べる必要があるから「基本構造３」の検査をする奴に回してくれ。

検査担当

検査から帰ってきた？終わった？
よし、これでミラー格子のセル(k,m+13)には、感染者２か回復者３の数字が入っているはずだ。そいじゃミラー格子セル(k,m+13)の値を調べて、２と３だったらそのままでいいけど、それ以外は全部感受性保持者ということで１のままにしといてくれ。

これでセル(k,m)の検査は終わりだ。
ご苦労様。（えーこれを 100 回やるのお！？）

感染判定２

前頁の**基本構造１**の中に入るのが**基本構造２**です。**基本構造１**は、すべての「**現実格子**」を構成するセルを巡るためのプログラムでしたね。すると**基本構造２**は、「**現実格子**」の一つのセル Cells (k,m) のことだけを考えればいいわけです。

ここでの 基本構文	**日本語訳**
Select Case A	A についてなんだけど
Case Is = 3	A = 3 の場合（Case）は X をしてほしい
X	
Case Is = 2	A = 2 の場合は Y をしてほしい
Y	
Case Is = 1	A = 1 の場合は Z をしてほしい
Z	
Case Else	それら以外の場合は W をしてほしいんだ。
W	
End Select	これだけなんだけど。（あんた、結構要求厳しいぜ）

真ん中のセルが Cells(k,m)とすると

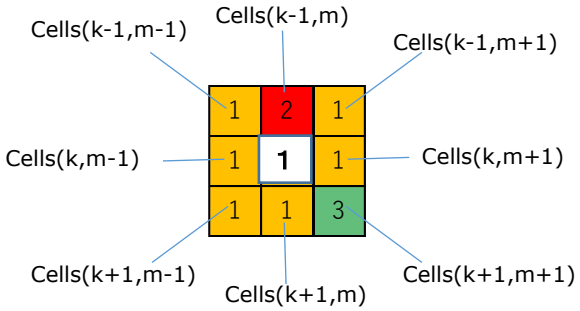


図 4-39

感染判定３

「周りの感染者が多くなると、感染する確率は大きくなるなあ。α=0.1 でも８人全員感染していたら・・・」
「８０％の確率ですか？」
「ああ・・・。」

基本構造３

```
For n = 1 To 3
        Select Case Cells(k - 1, m - 2 + n).Value
          Case Is = 2
           If Rnd() <=  α  Then Cells(k, m + 13) = 2
          Case Else
        End Select
Next n


For n = 1 To 3
     If n = 2 Then
       Else
        Select Case Cells(k, m - 2 + n).Value
          Case Is = 2
           If Rnd() <=  α  Then Cells(k, m + 13) = 2
          Case Else
         End Select
     End If
Next n


For n = 1 To 3
        Select Case Cells(k + 1, m - 2 + n).Value
          Case Is = 2
           If Rnd() <=  α  Then Cells(k, m + 13) = 2
          Case Else
        End Select
Next n
```

日本語訳

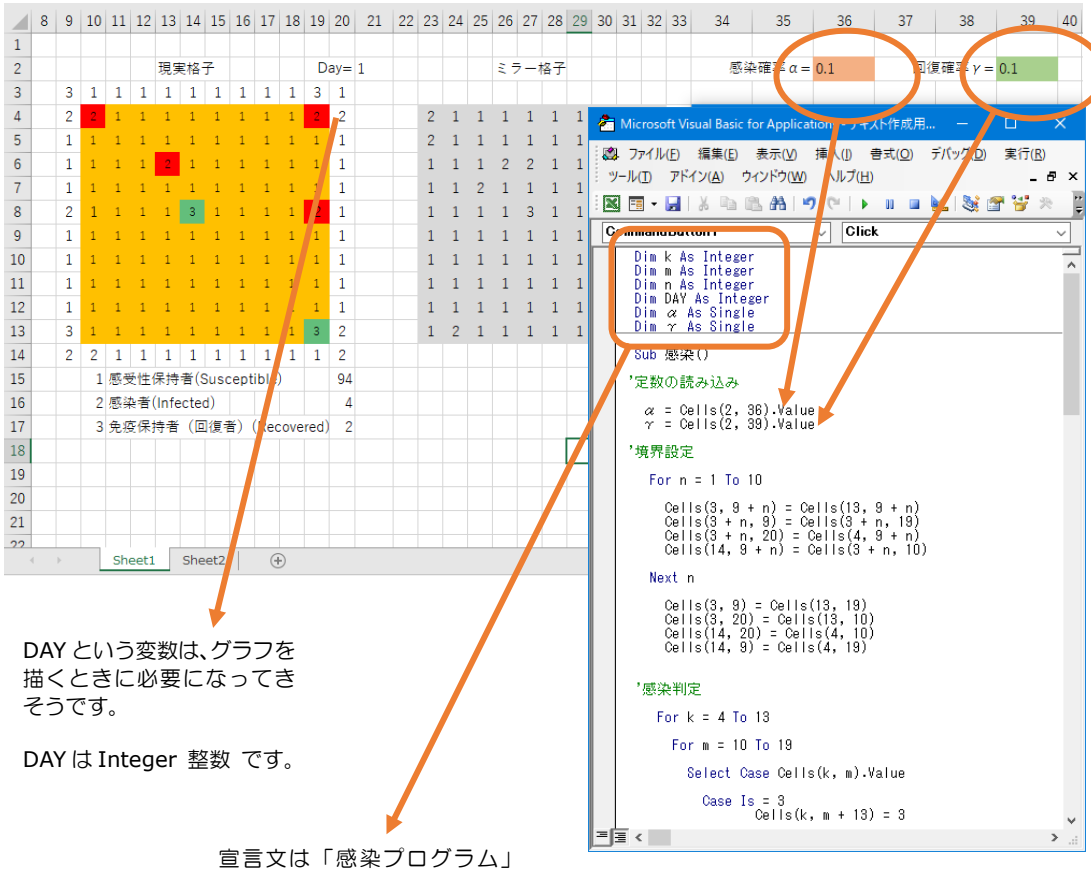
「まだ感染していない感受性保持者が基本構造２より回ってきましたあ。」

「よし調べてみよう。まず周りの 8 人のうち、上の 3 人だ。彼らの値が２だったらサイコロを振って感染率の値より小さかったら本人も感染したと判定、ほかの場合は 1 のままでいい。n=1 の左上から n=2 の真上、そして n=3 の右上の順で調べてくれ。もし本人も感染ならミラー格子の Cells(k,m+13)を 2 にしておいてくれ。他は何もしなくていい。」
「はい！n=1 から n=3 まで上の 3 人からの感染について調べました。」
「速いな」

「次は左右の人を調べよう。」
「はい」
「もちろん n=2 は本人だからなにもしない。n=1 の左隣の人が２だったらその人から感染しているかどうかサイコロを振って調べてくれ。もし感染率 α より小さかったら感染しているから、ミラー格子の Cells(k,m+13)を２にしろ」
「次に右隣も n=3 として調べてくれ。」

「n=1 と n=3 の左右の２人からの感染について調べました。」（いいだろう）

「よし、今度は彼の下側の 3 人だ。」
「同じように n=1 から n=3 まで順番に調べてくれ。やり方は上の 3 人の場合と一緒だ。」
「終わりましたあ！」
「よし、8 人全部調べたな。それじゃ基本構造２の連中に終わったことを連絡知れくれ。」



DAY という変数は、グラフを描くときに必要になってきそうです。

DAY は Integer 整数 です。

宣言文は「感染プログラム」（Sub 感染()）の外に出しておきます。これからボタンを設定し、そのボタンのプログラム(Private Sub プログラム)もこれらの変数を使う可能性があるからです。

α=Cells(2,36).Value
γ=Cells(2,39).Value

アルファは平仮名で入力して変換すると出てきます。平仮名でも OK です。ガンマも同じです。あとの文字は半角入力になります。
DAY という変数もシート上に作りました。注意してください。

Integer 整数
Single 実数（単精度）

図 4-40

プログラム製作２

大まかなプログラムの構造が「**境界設定**」と「**感染判定**」のプログラムでわかりましたか。さっそくそれを入力して、実際に「**ミラー格子**」に感染後の結果が写し取られるのかやってみましょう。

このとき**図 4-40** のように、変数の種類を書くのは **Sub 感染()**の外にしておきます。感染プログラムがメインですが、実際に動かすにはスタートボタンやクリアボタンのための簡単なプログラムを書く必要があるからです。

とはいってもまずこの「**感染プログラム**」が機能しているかどうかを確認しないと前に進みません。まず「**現実格子**」をすべて１にして、その中に１つだけ感染者である２を入れておきます。「**ミラー格子**」には何も書きません。まだプログラムを走らせるボタンなどをつけていませんので、**ファンクションキーF8** を使いながら、プログラムのミスを除いていき、最終的に▶で動き出すか確かめましょう。

確かめるポイントは、「**ミラー格子**」上は感染者の 2 が増加しているか、です。プログラムは 100 人を１回だけ判定しますので、「**ミラー格子**」に変化があるか、ないかだけがチェックポイントになります。

それがうまく言ったら、次の段階にいきましょう。

これまでに完成したプログラム

```
‘プログラムの中で勝手に置いた変数の種類を宣言しておきます
    Dim k As Integer
    Dim m As Integer
    Dim n As Integer
    Dim DAY As Integer
    Dim α As Single
    Dim γ As Single
```

```
Sub 感染()

  ' 定数の読み込み
        α = Cells(2, 36).Value
        γ = Cells(2, 39).Value

  ' 境界設定
      For n = 1 To 10
        Cells(3, 9 + n) = Cells(13, 9 + n)
        Cells(3 + n, 9) = Cells(3 + n, 19)
        Cells(3 + n, 20) = Cells(4, 9 + n)
        Cells(14, 9 + n) = Cells(3 + n, 10)
      Next n
        Cells(3, 9) = Cells(13, 19)
        Cells(3, 20) = Cells(13, 10)
        Cells(14, 20) = Cells(4, 10)
        Cells(14, 9) = Cells(4, 19)
```

基本構造１

```
‘感染判定
    For k = 4 To 13
      For m = 10 To 19
```

基本構造２

```
Select Case Cells(k, m).Value
  Case Is = 3
        Cells(k, m + 13) = 3
  Case Is = 2
      If Rnd() <= γ Then
         Cells(k, m + 13) = 3
       Else
         Cells(k, m + 13) = 2
      End If
  Case Is = 1
```

基本構造３

```
For n = 1 To 3
    Select Case Cells(k - 1, m - 2 + n).Value
          Case Is = 2
        If Rnd() <= α Then Cells(k, m + 13) = 2
      Case Else
    End Select
Next n

For n = 1 To 3
    If n = 2 Then
      Else
       Select Case Cells(k, m - 2 + n).Value
           Case Is = 2
               If Rnd() <= α Then Cells(k, m + 13) = 2
                   Case Else
         End Select
    End If
 Next n

 For n = 1 To 3
     Select Case Cells(k + 1, m - 2 + n).Value
       Case Is = 2
         If Rnd() <= α Then Cells(k, m + 13) = 2
       Case Else
     End Select
 Next n
```

```
        Select Case Cells(k, m + 13).Value
           Case Is = 2
           Case Is = 3
           Case Else
              Cells(k, m + 13).Value = 1
        End Select

    Case Else
  End Select
```

```
      Next m
  Next k
```

```
End Sub
```

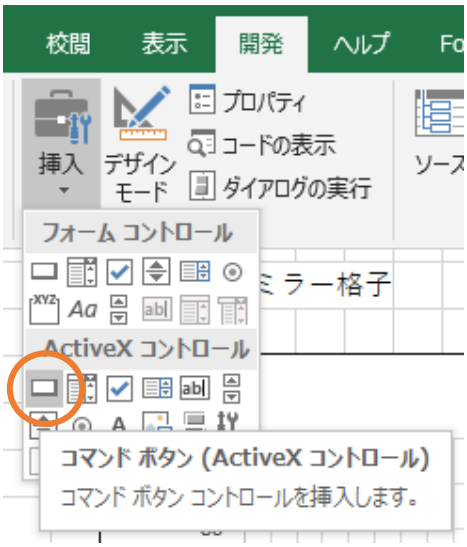


図 4-41

図 4-42

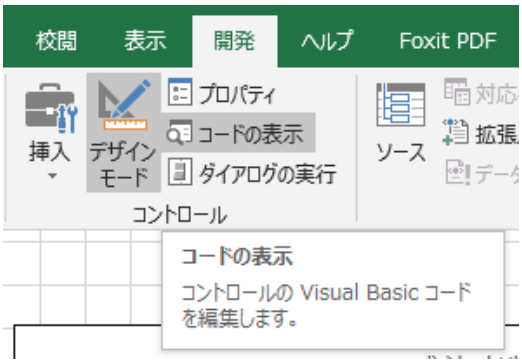


図 4-43

プログラム製作 3

最後の段階として、このプログラムをストレスなく動かして、実験ラボとして色々な実験ができるようにしなければいけません。

ザックリいうと、「**スタート**」と「**クリア**」のスイッチが機能するようにすればいいわけです。まず「**スタート**」のボタンを作りましょう。

図 4-41 のように「**開発**」のタブから「**挿入**」を選び「**ActiveX コントロール**」の四角いボタンを選びます。余談ですがこの「**挿入**」のアイコンは工具カバンになっています。何か実験ラボの「工事」をしている感じですね。

さて、そのボタンの名前を右クリックして『**コマンドボタンオブジェクト**』を選び「**編集**」を使って**図 4-43** のようにボタンの名前を書いておきます。

ここでは「**1 日後の感染状況**」としました。このボタンを押すと、1 日後の感染状況が「**現実格子**」にあらわれるようにします。そして、次にボタンを押すとその上場がグラフ化され、現実格子には 1 日たった後の感染状況が現れるようにします。

さて、このボタンを作っただけではもちろんプログラムは動き出しません。このボタンにはプログラムが書かれていなければいけないのです。それが「ActiveX」という意味です。

それでは**図 4-43** のようにこのボタンをデザインしていきます。**図 27** のように「**開発**」の「**デザインモード**」をクリックして「**コードの表示**」を選びます。

またコードは書いていませんので

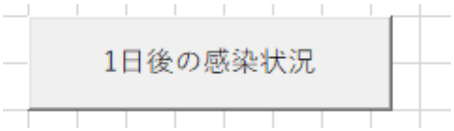
```
Private Sub CommandButton1 Click()

End Sub
```

のように最初と最後しか書かれていません。この中にコードを書いていきます。

ちなみに、このボタンの名前をいくら変えてもこのボタンの意味するコード名は『**コマンドボタン 1**』です。ボタンをたくさん作るときにはコードにはじめ「‘」（アポストロフィ）を書いて例えば「**’一日後の感染状況**」とか書いておくとコードとしては認識されませんので便利です。

「1 日後の感性状況」ボタンのコード




```
CommandButton1                    Click
Private Sub CommandButton1_Click()
'定数の読み込み
   DAY = Cells(2, 21).Value
'シートへの感染数の記入
   Cells(DAY + 4, 34).Value = DAY
   Cells(DAY + 4, 35).Value = Cells(15, 20).Value
   Cells(DAY + 4, 36).Value = Cells(16, 20).Value
   Cells(DAY + 4, 37).Value = Cells(17, 20).Value

   DoEvents

   Calculate
     DAY = DAY + 1
     Cells(2, 21) = DAY
'感染プログラムを実行する
     感染
'境界の消去
    For n = 1 To 12
      Cells(3, 8 + n).Clear
      Cells(2 + n, 9).Clear
      Cells(14, 8 + n).Clear
      Cells(2 + n, 20).Clear
    Next n
'ミラー格子を現実格子にプリント
  For k = 4 To 13
    For m = 10 To 19
      Cells(k, m) = Cells(k, m + 13).Value
    Next m
  Next k
End Sub
```

定数の読み込み：このボタンで一日ごとの感染の変化を追っていきます。感染者が出て何日目なのかが大事ですので、セル（2,21）に書いてある値をまず読み取ります。（単精度）

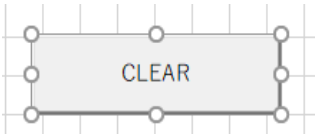
シートへの感染数の記入：感染日数に応じてその日の感染数をシートに感染日数の順に書き込んでいきます。これが直接グラフに反映されていきます。

感染プログラムを実行する：その日の感染分布を元に、次の日の「現実格子」の感染分布を表す感染プログラムを実行させます。

境界の消去：感染プログラムを走らせると「現実格子」の周りに風呂敷包み空間のために数値が書かれますので、次の計算のためにいったんこの周りの数値を消しておきます。

ミラー格子を現実格子にプリント：最後に感染プログラムで出た結果を現実格子に写します。

「CLEAR」ボタンのコード




```
Private Sub CommandButton2_Click()
'現実格子のセルをすべて1にリセット
    For k = 4 To 13
      For m = 10 To 19
         Cells(k, m) = 1
      Next m
    Next k
'ミラー格子の値をすべてクリアし配色を元に戻す
    Range(Cells(4, 23), Cells(13, 32)).Clear
    Range(Cells(4, 23), Cells(13, 32)).Interior.Color = RGB(204, 204, 255)
'グラフのデータをすべてクリアし日付DAYを１にする
    Range(Cells(5, 34), Cells(104, 37)).Clear
    Cells(2, 21) = 1

End Sub
```

現実格子のセルをすべて 1 にリセット：そのままの意味です。ただし次に新しい実験をするときには、この中に感染者 2 を手動で入れる必要があります。免疫保持者 3 でもかまいません。それは実験する人に考えてもらうことにします。

ミラー格子の値をすべてクリアし配色を変える：
そのままの意味です。2 行目の Interior.Color の意味は、このレンジ（範囲）のセルの中（インテリア）の色（カラー）という意味です。RGB(204、204、255)は、パソコンは R（レッド）と G（グリーン）と B（ブルー）の 3 つの色の強度を変えて混合することで色を作り出しています。数字はその強度のことです。
Range（レンジ）とは「範囲」のこと。四角い範囲の左上の角のセルと右下の角のセルの番号を書くだけで、その四角全体の「範囲」を示していることになる。

グラフのデータをすべてクリアし日付 DAY を 1 にする：
Clear は「消去」の意味。Range は右の説明を見てください。

これですべての作業が終わりました。これでうまく動き出すか実験してみましょう。「**クリア**」ボタンを押して「**現実格子**」の任意の場所を 1 か所だけ 2 にして感染者を出現させ、「**1 日後の感染者数**」をクリックしていきましょう。「**現実格子**」と「**グラフ**」が動き出せば成功です。

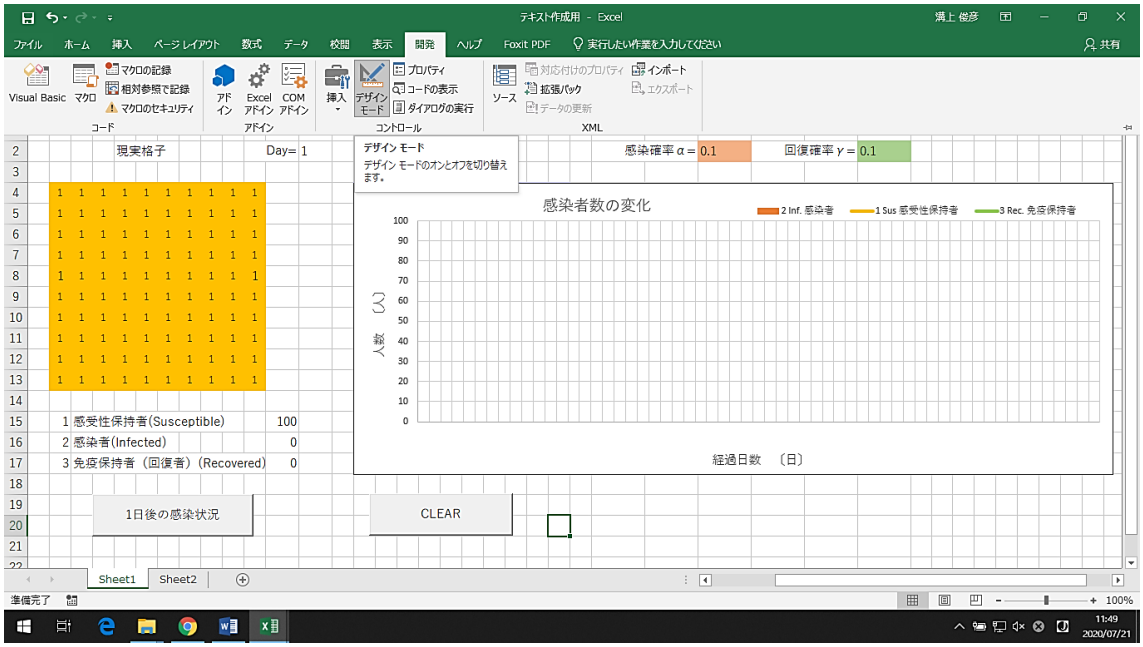


図 4-44

課題Ⅵ　ウイルス感染シミュレーション実験

完成したラボ（図 4-44）を使って、君のテーマでウイルスの感染をシミュレーションし結果を出せ。

この実験ラボだけでもいろいろな実験をすることが可能だ。

1．感染者が一人出た場合、どのようにして広がりどのように収束していくのか。

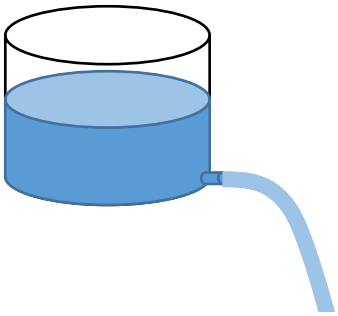
2．感染率を少しずつ変えていくと、どうなるのか。これは生活空間の物理的距離を変えていくことにも相当します。

3．何人かワクチンにより抗体を持った人がいると、どのような感染の形態になるのか。　等々

この実験の規模を大きくして実験することに挑戦する人も出てくるかもしれない。君たちのアイデアを自由に生かしてほしい。

きみろん Comp. 第 5 章

— オイラー法で微分方程式を解く—



現象を微分方程式で表しコンピューターで解く

教えて!goo の中に次のような質問をみつけました。

『ふと思ったのですが、風呂の栓を抜いて風呂の水が空になるまでの時間ってどうやって計算するのでしょうか？よく算数の問題で、「ある水槽で毎時 3L の水が抜けていきます」みたいなのを見るけど、風呂で見ている限り一定量の水が抜けていくわけじゃないですよね。最後のほうなんか勢いがなくなり抜けていきますから。もしかして、微分とか使う難しい計算なんでしょうか ？』

2020 年 7 月 30 日現在

質問が書かれた下側には、何人かの人がこの問題を解こうと試みていました。しかし残念ながら解法にたどり着いた人はいませんでした。質問した人は、「**微分**」という言葉を知っていますので高校生かもしれません。

高校生になって数学や物理を学んでいると、少しずつ自然界の謎が数学的に解けるのではないかという予感がしてきます。皆さんの中にもお風呂に入っているとき、同じ疑問と解けそうな予感を感じた人がいるのではないでしょうか。数学や物理、そしてその道具であるコンピューターは、その予感を現実の問題として設定し直し、その問題を解くために存在しています。

それでは、高校のレベルでこの問題は解けるのでしょうか。仮に解けるとして、この「**お風呂問題**」はどう解けばいいのでしょうか。「きみろん Comp.」は、君が見つけた研究テーマを、コンピューターを使って解けるようにするためにあります。今回はまず例題としてこの問題に取り組んでみることにしましょう。こういった問題に取り組むと、結果が「**微分方程式**」という形になることが結構あります。そんな方程式はこれまで解いたことがない人がほとんどでしょう。また、この「**微分方程式**」はコンピューターで解くことができます。コンピューターで方程式を解くというのはどういう方法なんでしょうか。

「微分方程式」を解き終わったとき、君はもう、あの算数の問題が得意だった小学校の頃に戻れないことを知るはずです。