

テーマ	追跡曲線について考えてみよう				
概要	追われる動物の動きを予測して追跡した場合を考える。				
ねらい	シミュレーション結果から工夫させる能力を付けさせる。				
関連する主な科目・項目	モデル化とシミュレーション	難易度	易・普・難	想定時間数	約1時間
準備するもの	表計算ソフト(Excel)	作成者	笠原 常豊		

1 学習の展開

課題1 キツネが 50m離れたところをウサギが一直線に走っていくのを見つけました。これをキツネは追いかけて走り出しました。時刻0にウサギは原点(0,0)、キツネは(0,50)にいるものとします。ウサギはキツネには気づかないでx軸の正の方向に $u = 6 \text{ m / 秒}$ で走ります。キツネはウサギの方向に向かって $v = 10 \text{ m / 秒}$ で追いかけるときのどうなるでしょうか？

時刻 t におけるウサギの位置 (x_t, y_t) キツネの位置 (X_t, Y_t) とすると、

時刻 $t + h$ におけるウサギの位置は、 $(x_t + u h, y_t)$ となり、

キツネの位置 $(X_t + (X_t - x_t) v h / \sqrt{(X_t - x_t)^2 + (Y_t - y_t)^2},$

$Y_t + (Y_t - y_t) v h / \sqrt{(X_t - x_t)^2 + (Y_t - y_t)^2})$ となります。

初期設定

Cells(1, 1) = "連番"	}	項目名
Cells(1, 2) = "時間"		
Cells(1, 3) = "x"		
Cells(1, 4) = "y"		
Cells(1, 5) = "X"		
Cells(1, 6) = "Y"		
Cells(1, 7) = "時間幅"		
Cells(1, 8) = "u"		
Cells(1, 9) = "v"		
Cells(2, 2) = 0		(時間初期値)
Cells(2, 3) = 0		(追われる初期 x 座標)
Cells(2, 4) = 0		(追われる初期 y 座標)
Cells(2, 5) = 0		(追う初期 x 座標)
Cells(2, 6) = 50		(追う初期 y 座標)
Cells(2, 7) = 0.1		(単位時間)
Cells(2, 8) = 6		(追われる速さ)
Cells(2, 9) = 10		(追う速さ)

結果の画面

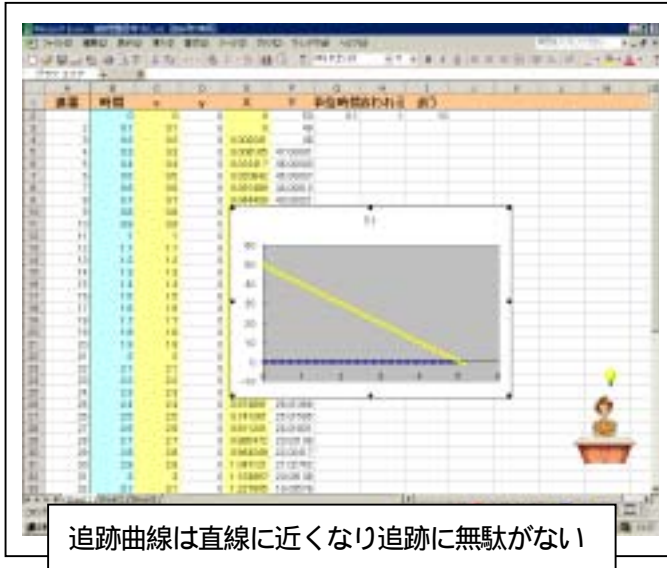


課題2 課題1の問題に対して、追う動物の速さと追われる動物の速さを変えてみよう。

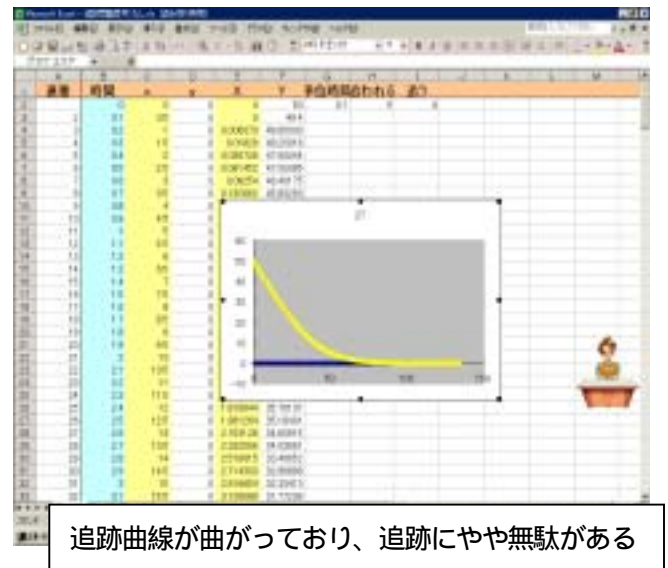
例題のように二つの動物の速さの比が約1:2の時の場合以外に下記の2通りで示そう。

1. 速さの差が大きい場合 (追う速さ 10 追われる速さ 1)
2. 速さの差が小さい場合 (追う速さ 6 追われる速さ 5)

1. (追う速さ 10 追われる速さ 1)



2. (追う速さ 6 追われる速さ 5)



課題3 上記のモデルを少し現実に近づける工夫をしてみましょう。

効率よく獲物を捕らえるには、できる限る最短時間で獲得するという考え方があります。そのためには追跡曲線を直線に近づけていったほうがよいと思われます。実際に動物が獲物を追う時には相手の行動を予測して向かっていくことがあるのではないのでしょうか。即ち、知能を持った動物ほど獲物の過去の行動を分析してその後を予測しながら行動することができるはずであります。よって、獲物に向かって追いかけるだけでなく、追いかける動物の数秒先を予測してそこに向かって行ったら少しでも効率が良く追いかけることができます。そこで、下記の(1)~(3)の考え方にそってアルゴリズムを再考してみよう。

- (1) 0秒後から数秒までは相手の行動が予測できないであろう。この時間を「初期」とします。
この初期の時間は、追う動物は獲物に向かって進むことにします。
- (2) 数秒前からの過去のデータを利用して未来を予測した場所に向かって追跡する時間を「中期」とします。すなわち、この時は獲物よりも先を目指して進むことになります。
追われる動物の位置を(過去係数×単位時間×2)秒前の位置をSとして、
(過去係数×単位時間×1)秒前の位置をTとします。
追われる動物の今の位置をUとすると、
追われる動物が $((U - T) / (T - S)) \times (U - T) + U$ の位置に向かうと予想して、
追う動物がその方向に向かうと決めます。
- (3) 最後の数秒間は、獲物との距離が近づくためひたすら獲物に向かって突き進むものとします。
この追跡する時間を「終期」と呼びます。

初期設定 ()

```

Cells(1, 2) = "時間"
Cells(1, 3) = "x"
Cells(1, 4) = "y"
Cells(1, 5) = "X"
Cells(1, 6) = "Y"
Cells(1, 7) = "時間幅"
Cells(1, 8) = "u"
Cells(1, 9) = "v"
Cells(2, 2) = 0 (時間初期値)
Cells(2, 3) = 0 (追われる初期x座標)
Cells(2, 4) = 0 (追われる初期y座標)
Cells(2, 5) = 0 (追う初期x座標)
Cells(2, 6) = 50 (追う初期y座標)
Cells(2, 7) = 0.1 (単位時間)
Cells(2, 8) = 4 (追われる速さ)
Cells(2, 9) = 5 (追う速さ)
    
```

初期設定 ()

```

Cells(4, 8) = "過去係数"
Cells(4, 9) = "前々"
Cells(4, 10) = "前"
Cells(5, 8) = 10 (過去係数)
Cells(5, 9) = Cells(5, 8) * Cells(2, 7) * 2
Cells(5, 10) = Cells(5, 8) * Cells(2, 7)

Cells(7, 8) = "初期割合"
Cells(7, 9) = "中期割合"
Cells(7, 10) = "終期割合"
Cells(8, 8) = 0.2 (初期割合)
Cells(8, 9) = 0.7 (中期割合)
Cells(8, 10) = 1 - Cells(8, 8) - Cells(8, 9)
    
```

ループによる計算

```

Do
初期
n = n + 1
GoSub Emono
If (Cells(n + 1, 6) <= (1 - Cells(8, 8)) *
Cells(2, 6)) Then
f = 1
End If
Loop While f = 0

中期
f = 0
Do
n = n + 1
GoSub Emono_yosou
If (Cells(n + 1, 6) <= (1 - Cells(8, 8) -
Cells(8, 9)) * Cells(2, 6)) Then
f = 1
End If
Loop While f = 0

終期
f = 0
Do
n = n + 1
GoSub Emono
dx = Cells(n + 1, 3) - Cells(n + 1, 5)
dx = Cells(n + 1, 6) - Cells(n + 1, 4)
If (dx <= 0 And dy <= 0) Then
f = 1
End If
Loop While f = 0
    
```

サブルーチンによる計算

```

Emono: (獲物に向かっていく計算)
Cells(n + 1, 1) = n
Cells(n + 1, 2) = Cells(n, 2) + Cells(2, 7)
Cells(n + 1, 3) = Cells(n, 3) + Cells(2, 7) * Cells(2, 8)
Cells(n + 1, 4) = Cells(n, 4)
a = (Cells(n, 3) - Cells(n, 5)) ^ 2
b = (Cells(n, 4) - Cells(n, 6)) ^ 2
c = a + b

Cells(n + 1, 5) = Cells(n, 5) + ((Cells(n, 3) - Cells(n, 5)) *
Cells(2, 7) * Cells(2, 9)) / Sqr(c)
Cells(n + 1, 6) = Cells(n, 6) + ((Cells(n, 4) - Cells(n, 6)) *
Cells(2, 7) * Cells(2, 9)) / Sqr(c)

Return

Emono_yosou: (獲物の予想場所に向かっていく計算)
Cells(n + 1, 1) = n
Cells(n + 1, 2) = Cells(n, 2) + Cells(2, 7)
Cells(n + 1, 3) = Cells(n, 3) + Cells(2, 7) * Cells(2, 8)
Cells(n + 1, 4) = Cells(n, 4)

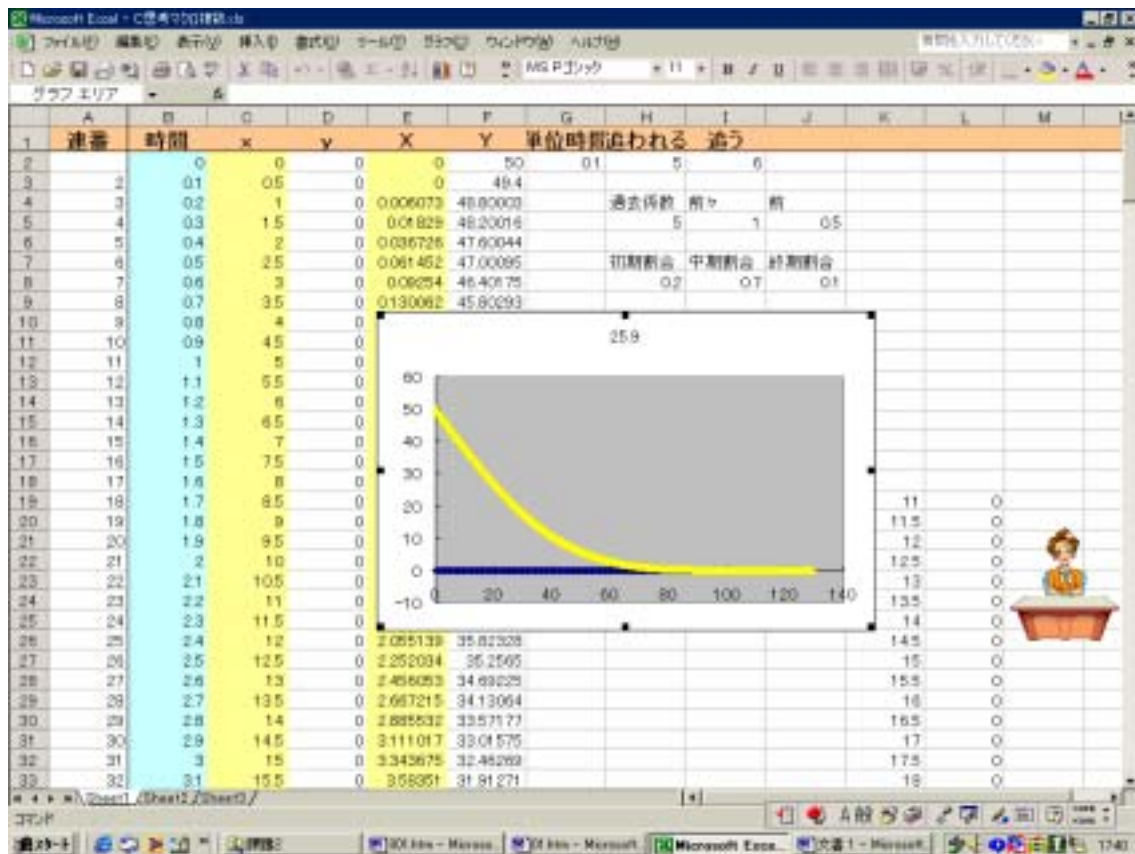
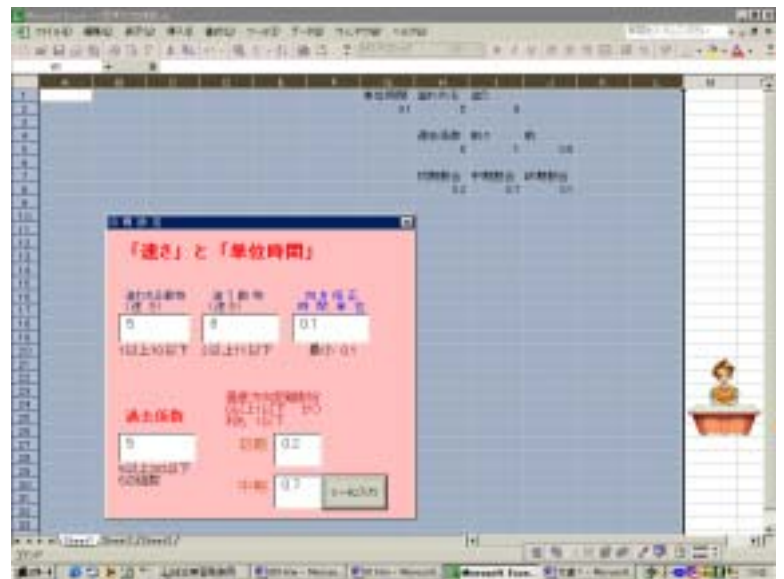
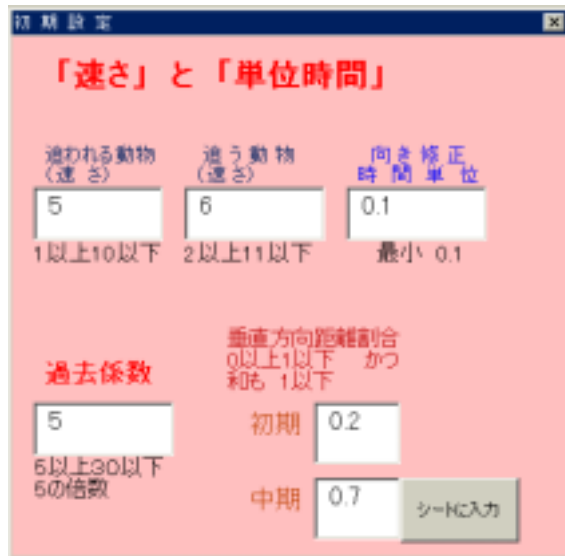
k = Cells(5, 8)
If (n - 2 * k < 3) Then
Cells(n, 11) = Cells(2, 3)
Else
Cells(n, 11) = Cells(n, 3) + ((Cells(n, 3) - Cells(n - k, 3)) ^
2) / (Cells(n - k, 3) - Cells(n - 2 * k, 3))
End If

Cells(n, 12) = Cells(n + 1, 4)

a = (Cells(n, 11) - Cells(n, 5)) ^ 2
b = (Cells(n, 12) - Cells(n, 6)) ^ 2
c = a + b

Cells(n + 1, 5) = Cells(n, 5) + ((Cells(n, 11) - Cells(n, 5)) *
Cells(2, 7) * Cells(2, 9)) / Sqr(c)
Cells(n + 1, 6) = Cells(n, 6) + ((Cells(n, 12) - Cells(n, 6)) *
Cells(2, 7) * Cells(2, 9)) / Sqr(c)

Return
    
```



やや、追跡時間が短縮されたことがわかります。

2 発展学習

- (1) マクロを利用して、「初期値を変化させたときどのように到達時間が変化するか」を求めよう。
- (2) 追う動物・追われる動物が等直線運動以外の動きをするとき、そのシミュレーションをしてみよう。

