$restart:$

$with(plots):$

$with(Statistics):$

$with(StringTools):$

$with(LinearAlgebra):$

$with(ColorTools):$

$$r := (x1, x2, y1, y2) \rightarrow \mathrm{sqrt}\left((x1-x2)^2 + (y1-y2)^2\right):$$

$$Gx := (x1, x2, y1, y2) \rightarrow -\frac{1}{r(x1, x2, y1, y2)^2} \cdot (x1-x2):$$

$$Gy := (x1, x2, y1, y2) \rightarrow -\frac{1}{r(x1, x2, y1, y2)^2} \cdot (y1-y2):$$

$Gx(2, 3, 3, 8)$

$$\frac{1}{26} \tag{1}$$

$$rr := \frac{rand(0..1000)}{40}:$$

$N := 2:$

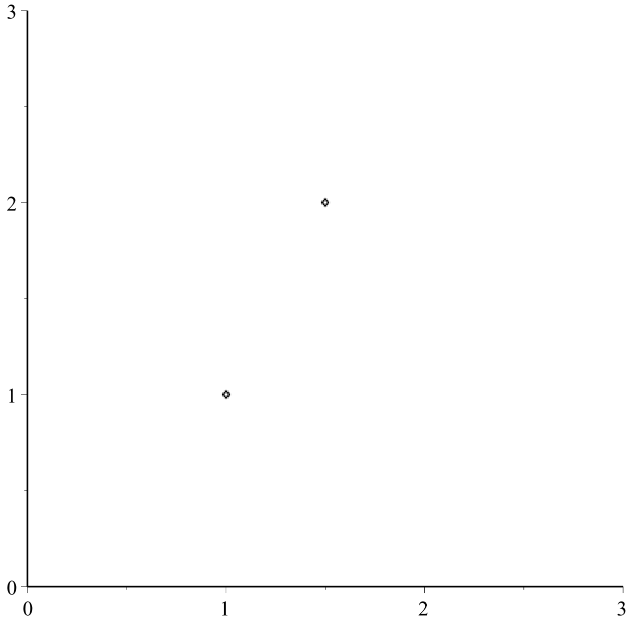
$IR := [[[1, 1], [1.5, 2]]];$

$IV := [[[0, 0.8], [1, 1]]]$

$$[[[1, 1], [1.5, 2]]]$$

$$[[[0, 0.8], [1, 1]]] \tag{2}$$

$pointplot(IR[1], view = [0..3, 0..3])$



$thau := 0.25 :$
$T := 10 :$
$Nt := \frac{T}{thau}$

$$40.00000000 \qquad (3)$$

$XX := Array([IR[1], op([seq([seq([rr(\ ), rr(\ )], k=1..N)], i=1..Nt)])])$

$$\left[\begin{array}{c} \textit{1..41 x 1..2 x 1..2 Array} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{array}\right] \qquad (4)$$

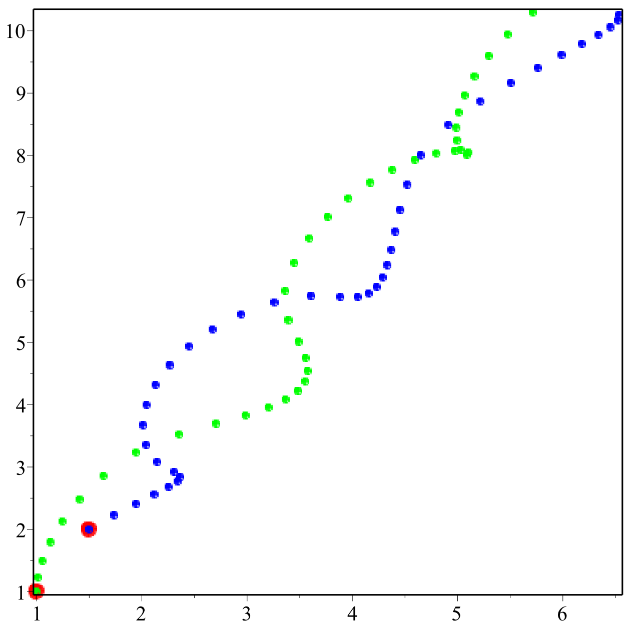
$VV := Array([IV[1], op([seq([seq([1, 2], k=1..N)], i=1..Nt)])])$

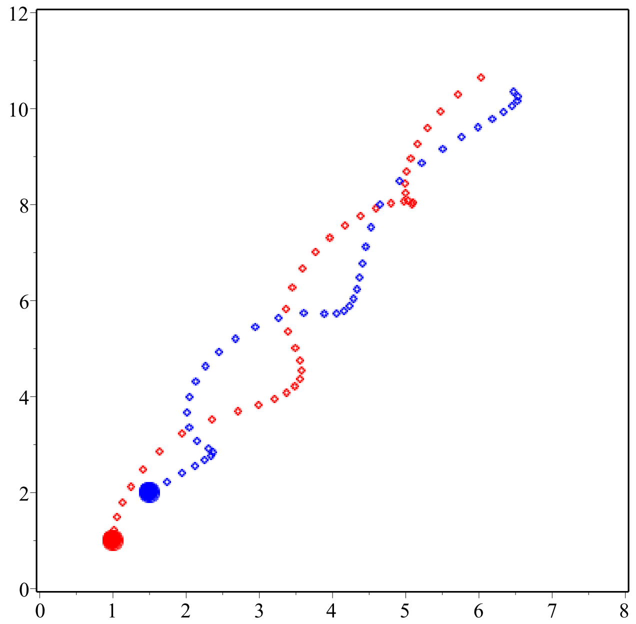
$$\left[\begin{array}{c} \textit{1..41 x 1..2 x 1..2 Array} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{array}\right] \qquad (5)$$

```
t :='t': i :='i': j :='j': k :='k':
for t from 2 to Nt do
   for i from 1 to N do
     SumX := 0;
     SumY := 0;
     for k from 1 to N do
       if i ≠ k then
         SumX := SumX + evalf(Gx(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2]));
         SumY := SumY + evalf(Gy(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2]));
      end if
     end do:
         XX[t, i, 1] := XX[t − 1, i, 1] + thau·VV[t − 1, i, 1] + thau^2/2 ·SumX;
         XX[t, i, 2] := XX[t − 1, i, 2] + thau·VV[t − 1, i, 2] + thau^2/2 ·SumY;
   end do:
   for i from 1 to N do
     SumVY := 0;
     SumVX := 0;
     for k from 1 to N do
       if i ≠ k then
         SumVX := SumVX + evalf(Gx(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k,
     2]) + Gx(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2]));
         SumVY := SumVY + evalf(Gy(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2])
      + Gy(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2]));
      end if
     end do:
         VV[t, i, 1] := VV[t − 1, i, 1] + thau/2 ·SumVX;
         VV[t, i, 2] := VV[t − 1, i, 2] + thau/2 ·SumVY;

   end do

  end do:
XX[5, 2]
                         [ 2.253341827   2.676328124 ]                         (6)
col := (green, blue, yellow )
                               green, blue, yellow                             (7)
IA := display(seq(pointplot(XX[1, j], color = red, symbolsize = 20, symbol = solidcircle), j = 1 ..N))
                                  PLOT(...)
IB := display(seq(seq(pointplot(XX[i, j], color = col[j], symbolsize = 10, symbol = solidcircle), i = 1
       ..Nt), j = 1 ..N))
                                  PLOT(...)                                    (9)
display(IA, IB, axes = boxed)
```


10
9
8
7
6
5
4
3
2
1
1
2
3
4
5
6



```
VFX := (i, k) → piecewise(i ≠ k, Gx(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2]),
     0) :
 VFY := (i, k) → piecewise(i ≠ k, Gy(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k,
     2]), 0) :
 VFVx := (i, k) → piecewise(i ≠ k, Gx(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2])
      + Gx(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2])) :
 VFVy := (i, k) → piecewise(i ≠ k, Gy(XX[t − 1, i, 1], XX[t − 1, k, 1], XX[t − 1, i, 2], XX[t − 1, k, 2])
      + Gy(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2])) :

t :='t': i :='i': j :='j': k :='k':
for t from 2 to Nt do
   for i from 1 to N do
     SumX := evalf(add(VFX(i, k), k = 1 ..N));
     SumY := evalf(add(VFY(i, k), k = 1 ..N));
     XX[t, i, 1] := XX[t − 1, i, 1] + thau·VV[t − 1, i, 1] + thau^2/2 ·SumX;
      XX[t, i, 2] := XX[t − 1, i, 2] + thau·VV[t − 1, i, 2] + thau^2/2 ·SumY;
    end do:
    for i from 1 to N do
      SumVX := evalf(add(VFVx(i, k), k = 1 ..N));
```

```
        $SumVY := evalf(add(VFVy(i, k), k = 1 ..N));$
        $VV[t, i, 1] := VV[t - 1, i, 1] + \frac{thau}{2} \cdot SumVX;$
        $VV[t, i, 2] := VV[t - 1, i, 2] + \frac{thau}{2} \cdot SumVY;$
    end do
end do:
```