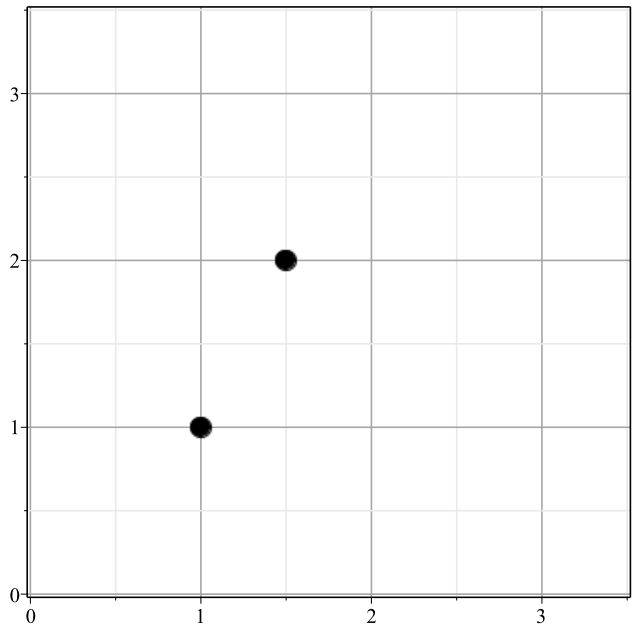
```
restart
with(stats) :
with(plots) :
with(Statistics) :
with(StringTools) :
with(LinearAlgebra) :
xA0 := 1 :
yA0 := 1 :
xB0 := 1.5 :
yB0 := 2 :
 VAx0 := 0 :
 VAy0 := 0.8 :
 VBx0 := 1 :
 VBy0 := 1 :
r := (x1, x2, y1, y2) → sqrt( (x1 − x2)^2 + (y1 − y2)^2 ) :
Gx := (x1, x2, y1, y2) → - 1/r(x1, x2, y1, y2)^2 · (x1 − x2) :
Gy := (x1, x2, y1, y2) → - 1/r(x1, x2, y1, y2)^2 · (y1 − y2) :
Gx(1, 2, 3, 4)
```

$$\frac{1}{2} \tag{1}$$

```
OR := [ [ [xA0, yA0], [xB0, yB0] ] ] :
OV := [ [ [VAx0, VAy0], [VBx0, VBy0] ] ] :
pointplot(OR[1], symbol = solidcircle, symbolsize = 25, gridlines = true, axes = boxed, view = [0 ..3.5, 0
      ..3.5 ])
```



$t := 0.25$:
$TT := 10$:
$N := \frac{TT}{t}$

40.00000000 **(2)**

$XX := [[[xA0, yA0], [xB0, yB0]]]$:
$VV := [[[VAx0, VAy0], [VBx0, VBy0]]]$:

for i **from** 1 **to** N **do**:

$xA1 := xA0 + t \cdot VAx0 + \frac{t^2}{2} \cdot Gx(xA0, xB0, yA0, yB0)$;

$yA1 := yA0 + t \cdot VAy0 + \frac{t^2}{2} \cdot Gy(xA0, xB0, yA0, yB0)$;

$xB1 := xB0 + t \cdot VBx0 + \frac{t^2}{2} \cdot Gx(xB0, xA0, yB0, yA0)$;

$yB1 := yB0 + t \cdot VBy0 + \frac{t^2}{2} \cdot Gy(xB0, xA0, yB0, yA0)$;

$VAx1 := VAx0 + \frac{t}{2} \cdot (Gx(xA0, xB0, yA0, yB0) + Gx(xA1, xB1, yA1, yB1))$;

$$VAy1 := VAy0 + \frac{t}{2} \cdot (Gy(xA0, xB0, yA0, yB0) + Gy(xA1, xB1, yA1, yB1));$$

$$VBx1 := VBx0 + \frac{t}{2} \cdot (Gx(xB0, xA0, yB0, yA0) + Gx(xB1, xA1, yB1, yA1));$$

$$VBy1 := VBy0 + \frac{t}{2} \cdot (Gy(xB0, xA0, yB0, yA0) + Gy(xB1, xA1, yB1, yA1));$$

$XX := [op(XX), [[xA1, yA1], [xB1, yB1]]];$
$VV := [op(VV), [[VAx1, VAy1], [VBx1, VBy1]]];$
$xA0 := xA1;$
$yA0 := yA1;$
$xB0 := xB1;$
$yB0 := yB1;$
$VAx0 := VAx1;$
$VAy0 := VAy1;$
$VBx0 := VBx1;$
$VBy0 := VBy1;$
end do:

$XX[3]$

$$[[1.054700942, 1.490966816], [1.945299058, 2.409033184]] \quad \textbf{(3)}$$

$IA := pointplot(XX[1, 1], color = red, symbolsize = 25, symbol = solidcircle)$

$$PLOT(\ldots) \quad \textbf{(4)}$$

$IB := pointplot(XX[1, 2], color = blue, symbolsize = 25, symbol = solidcircle)$

$$PLOT(\ldots) \quad \textbf{(5)}$$

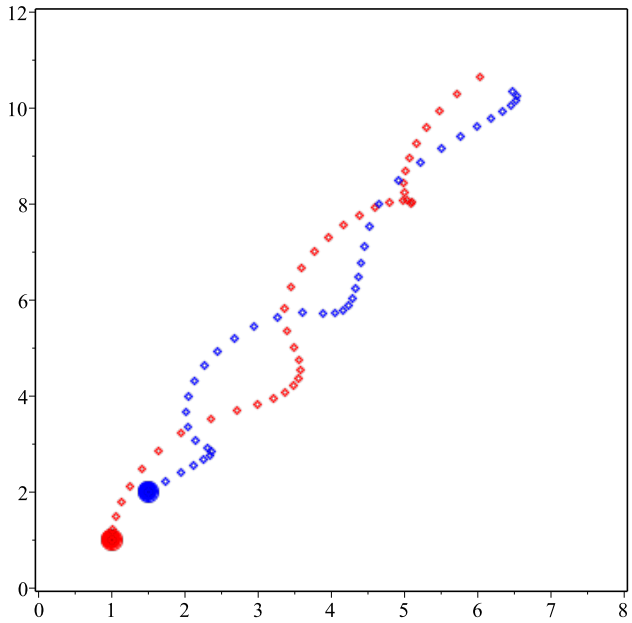
$A := display(seq(pointplot(XX[i, 1], color = red), i = 1 ..nops(XX)))$

$$PLOT(\ldots) \quad \textbf{(6)}$$

$B := display(seq(pointplot(XX[i, 2], color = blue), i = 1 ..nops(XX)))$

$$PLOT(\ldots) \quad \textbf{(7)}$$

$display(IA, IB, A, B, view = [0 ..8, 0 ..12], axes = boxed)$




```
restart :
with(plots) :
 with(Statistics) :
 with(StringTools) :
 with(LinearAlgebra) :
 with(ColorTools) :
r := (x1, x2, y1, y2) → sqrt((x1 − x2)^2 + (y1 − y2)^2) :
Gx := (x1, x2, y1, y2) → -1/r(x1, x2, y1, y2)^2 · (x1 − x2) :
Gy := (x1, x2, y1, y2) → -1/r(x1, x2, y1, y2)^2 · (y1 − y2) :

Gx(2, 3, 3, 8)
```

$$\frac{1}{26} \qquad \textbf{(8)}$$

```
rr := rand(0..1000)/40 :
 rv := rand(-500..500)/1000 :
```

$N := 8$:

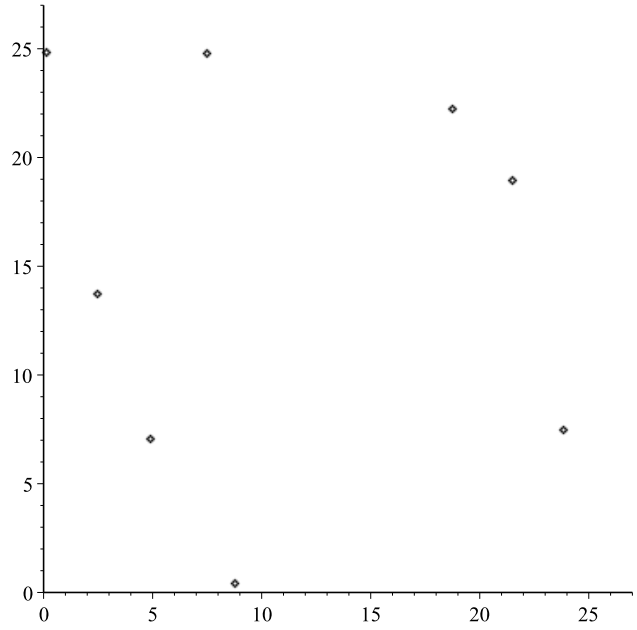
$IR := [[seq([rr(\,), rr(\,)], i = 1..N)]]$;

$IV := [[seq([rv(\,), rv(\,)], i = 1..N)]]$

$$\left[\left[\left[\frac{43}{2}, \frac{379}{20}\right], \left[\frac{75}{4}, \frac{889}{40}\right], \left[\frac{15}{2}, \frac{991}{40}\right], \left[\frac{1}{8}, \frac{993}{40}\right], \left[\frac{477}{20}, \frac{299}{40}\right], \left[\frac{99}{40}, \frac{549}{40}\right], \left[\frac{49}{10}, \frac{141}{20}\right], \left[\frac{351}{40}, \frac{2}{5}\right]\right]\right]$$

$$\left[\left[\left[\frac{243}{1000}, \frac{6}{25}\right], \left[\frac{357}{1000}, \frac{131}{500}\right], \left[-\frac{339}{1000}, -\frac{227}{1000}\right], \left[-\frac{321}{1000}, -\frac{189}{1000}\right], \left[-\frac{33}{100}, \frac{47}{500}\right], \left[-\frac{119}{250}, \frac{101}{1000}\right], \left[\frac{13}{125}, \frac{327}{1000}\right], \left[-\frac{3}{125}, -\frac{103}{1000}\right]\right]\right] \quad \textbf{(9)}$$

$pointplot(IR[1], view = [0..27, 0..27])$



$thau := 0.05$:

$T := 30$:

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

600.0000000 **(10)**

Try to separate in multiple steps... And explain each step
Creates N entries of pairs of values (X,Y) pairs. This is our row

$ss := seq([1, 2], p = 1..N)$

$$[1, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2] \quad \textbf{(11)}$$

Here we create i many rows of ss. So array is like Nxi but eat entry is 1x2
each row is a new position (X,Y) for each of 8 particles.

$Array([seq([ss], i = 1..Nt)])$

$$\begin{bmatrix} 1..600\ x\ 1..8\ x\ 1..2\ Array \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \quad \textbf{(12)}$$

everyrhing in 1 step as kyrkos did
Maybe make two separate arrays for YY and XX

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

maybe different approach for array?

$FF := Array([IR[1], seq([[op([1, 2])], [op([1, 2])]], o = 1..2)])$

$$\begin{bmatrix} 1..3\ x\ 1..5\ x\ 1..2\ Array \\ Data\ Type: anything \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix} \quad \textbf{(13)}$$

we cannot have particle act on itself becasue it does not, byt we have all the other actions on it, so we have N-1 forces on the particle acting. In the piecewise we have tt is the time (row number) and k is the particle number and i is the particle number. Because we need each i with every k summed up.
Idea - do piecevise and sum in the loop with the if condition, it k!=i then sum stuff up

$VFX := (i, k) \rightarrow piecewise(i \neq k, Gx(XX[t-1, i, 1], XX[t-1, k, 1], XX[t-1, i, 2], XX[t-1, k, 2]), 0):$
$VFY := (i, k) \rightarrow piecewise(i \neq k, Gy(XX[t-1, i, 1], XX[t-1, k, 1], XX[t-1, i, 2], XX[t-1, k, 2]), 0):$
$VFVx := (i, k) \rightarrow piecewise(i \neq k, Gx(XX[t-1, i, 1], XX[t-1, k, 1], XX[t-1, i, 2], XX[t-1, k, 2]) + Gx(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2])):$
$VFVy := (i, k) \rightarrow piecewise(i \neq k, Gy(XX[t-1, i, 1], XX[t-1, k, 1], XX[t-1, i, 2], XX[t-1, k, 2]) + Gy(XX[t, i, 1], XX[t, k, 1], XX[t, i, 2], XX[t, k, 2])):$

$t := 't': i := 'i': j := 'j': k := 'k':$
for t **from** 2 **to** Nt **do**
  for i **from** 1 **to** N **do**
    $SumX := evalf(add(VFX(i, k), k = 1..N));$
    $SumY := evalf(add(VFY(i, k), k = 1..N));$

$$XX[t, i, 1] := XX[t-1, i, 1] + thau \cdot VV[t-1, i, 1] + \frac{thau^2}{2} \cdot SumX;$$

$$XX[t, i, 2] := XX[t-1, i, 2] + thau \cdot VV[t-1, i, 2] + \frac{thau^2}{2} \cdot SumY;$$

  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:

$SumFX$

$$-0.5725664487 \quad \textbf{(14)}$$

$SumFY$

$$1.554047900 \quad \textbf{(15)}$$

$$XX[t,i,1] := XX[t-1,i,1] + thau\cdot VV\cdot[t-1,t,1] + \frac{thau^2}{2}\cdot SumFX;$$

Error, Array index out of range

Idea - do piecevise and sum in the loop with the if condition, it k!=i then sum stuff up

for t **from** 2 **to** Nt **do**
for i **from** 1 **to** N **do**
$SumX := 0;$
$SumY := 0;$
$SumVY := 0;$
$SumVX := 0;$
for k **from** 1 **to** N **do**
if $i \neq k$ **then**
$SumX := SumX + evalf(Gx(XX[t-1,i,1], XX[t-1,k,1], XX[t-1,i,2], XX[t-1,k,2]));$
$SumY := SumY + evalf(Gy(XX[t-1,i,1], XX[t-1,k,1], XX[t-1,i,2], XX[t-1,k,2]));$
$XX[t,i,1] := XX[t-1,i,1] + thau\cdot VV[t-1,i,1] + \frac{thau^2}{2}\cdot SumX;$
$XX[t,i,2] := XX[t-1,i,2] + thau\cdot VV[t-1,i,2] + \frac{thau^2}{2}\cdot SumY;$
$SumVX := SumVX + evalf(Gx(XX[t-1,i,1], XX[t-1,k,1], XX[t-1,i,2], XX[t-1,k,2]) + Gx(XX[t,i,1], XX[t,k,1], XX[t,i,2], XX[t,k,2]));$
$SumVY := SumVY + evalf(Gy(XX[t-1,i,1], XX[t-1,k,1], XX[t-1,i,2], XX[t-1,k,2]) + Gy(XX[t,i,1], XX[t,k,1], XX[t,i,2], XX[t,k,2]));$
$VV[t,i,1] := VV[t-1,i,1] + \frac{thau}{2}\cdot SumVX;$
$VV[t,i,2] := VV[t-1,i,2] + \frac{thau}{2}\cdot SumVY;$
end if
end do
end do
end do:

Warning, computation interrupted

$XX[7,3]$

$$\begin{bmatrix} 7.398219526 & 24.69589048 \end{bmatrix}$$

$col := (green, blue, \text{"Peru"}, yellow, \text{"Niagara DarkOrchid"}, orange, \text{"Tan"}, \text{"Pink"})$

$green, blue, \text{"Peru"}, yellow, \text{"Niagara DarkOrchid"}, orange, \text{"Tan"}, \text{"Pink"}$ **(17)**

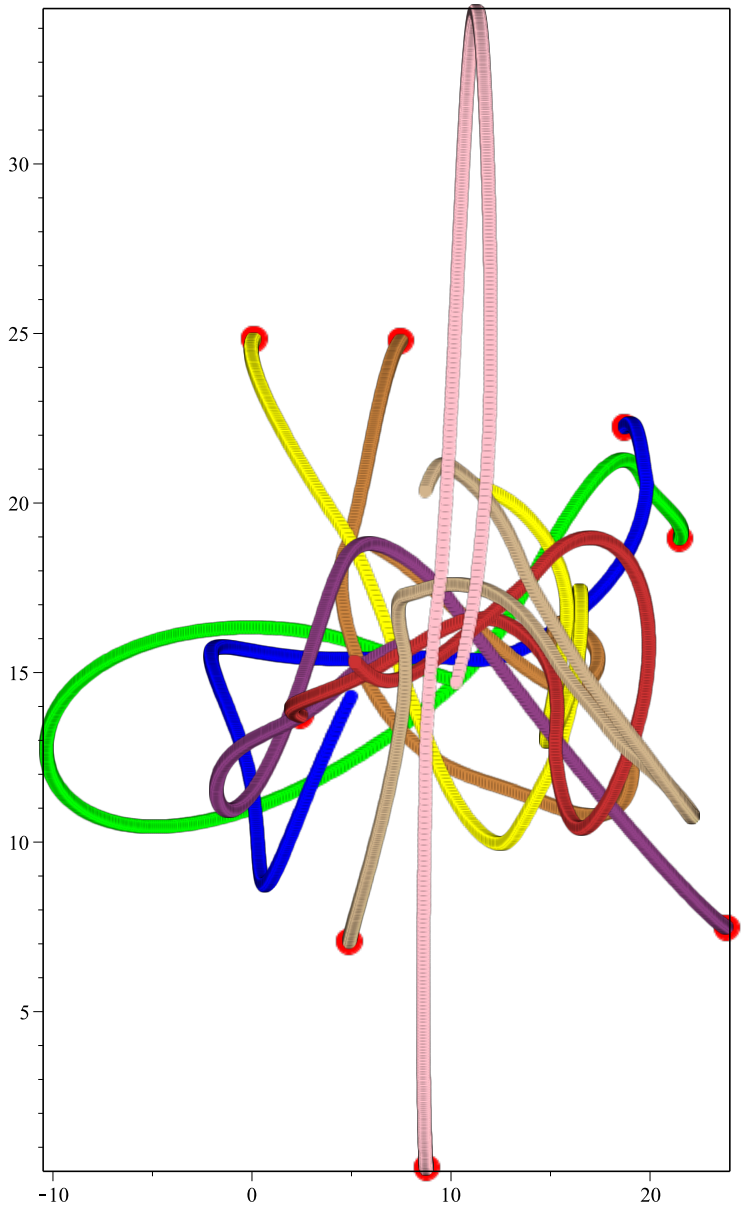
$IA := display(seq(pointplot(XX[1, j], color = red, symbolsize = 20, symbol = solidcircle), j = 1..N))$

$PLOT(...)$

$IB := display(seq(seq(pointplot(XX[i, j], color = col[j], symbolsize = 10, symbol = solidcircle), i = 1..Nt), j = 1..N))$

$PLOT(...)$ **(19)**

$display(IA, IB, axes = boxed)$


30
25
20
15
10
5
-10
0
10
20

$t := 't'$

t **(20)**

$animate((pointplot, [seq(XX[t, i], i = 1 ..N)], t = 1 ..Nt, axes = boxed, symbol = solidcircle, symbolsize = 10))$

Error, bad index into Array

```
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
  end do
end do:
```