Project No. 06:
Particle Swarm Optimization

By: Adam Rosenbaum

Instructor: Bruce MacLennan

UT-Knoxville CS420

8 May 2017

I. INTRODUCTION

This project *Particle Swarm Optimization* is an exploration into the types of behavior exemplified in bird flocking or fish schooling. Specifically, outcomes were explored between a range of inertias across various sized populations with sociability and cognition. In this project, each randomly initialized particle used a fitness function in hopes of achieving a maximum fitness dependent on the globally most fit particle within the smallest proximity to a maxima. While the population converges to a optimal fitness, calculations of average locailty error and the number of particles within a radius of the maxima are observed. As well, qualitative analysis of the population behavior is given in an attempt to identify parameters that provide a more optimal solution.

II. METHODS

To perform the experiment a particle swarm algorithm has been programmed in C++ specifically for this purpose. The program implements particle steering utilizing inertia, cognition, and sociability; and performs all calculations necessary to update fitness accross the population. Additional scripting languages are utilized to generate plots and perform qualitative analysis as well. Parameters supplied to the program include terrain maxima; the population size; particle inertia; ranges of sociability and cognition; and finally, the number of epochs to simulate.

A. The Particle Swarm Algorithm

```
for p in Swarm:
    position = randomPosition()
    p.initialize(position, fitness(position))
    if p.fitness > Swarm.best.fitness:
        Swarm.best = p
for e in epoch:
    error = AvgError(Swarm)
    if error <= threshold: break
    else:
        for p in Swarm:
            p.updatePosition(Swarm.best)
            if p.fitness > Swarm.best.fitness:
                Swarm.best = p
```

The Particle Swarm Algorithm states for each particle (p) in the swarm, initialize the particle to a random position and calculate the fitness of this position. The particle having the best fitness in the swarm is recorded. For each epoch of the simulation, if the error is below a predefined threshold - end the simulation. Otherwise, update each particle position in accordance to the most fit particle in the swarm. When a particle becomes more fit than the most fit in the swarm, record this particle as the new fittest particle. Albeit this algorithm suffices as an abstract view, it lacks granularity concerning the determination of fitness, dynamics (movement) and furthermore, how do we define an optimum parameter set?

B. Fitness

As aforementioned, our fitness scheme describes the most "fit" gene as having closest proximity to the specified maxima. The *global* maxima has been predetermined to be the x and y coodinates (20, 7); and for later experimentation a supplimental *local* maxima of (-20, -7) was included. These coordinates are utilized in unison with maximum world dimensions (x_{max} and y_{max}) of 100 to facilitate distance equations for calculating fitness. Accordingly, the distance equations materialize as follows:

$$d_m = \frac{\sqrt{x_{max}^2 + y_{max}^2}}{2}$$

$$d_n = \sqrt{(p_x + 20)^2 + (p_y + 7)^2}$$

$$d_p = \sqrt{(p_x - 20)^2 + (p_y - 7)^2}$$

where p_x, p_y are the particles x and y coordinates respectively. In the absence of a *local* maxima, the fitness equation can now be evaluated as equation (1). Converseley, equation (2) evaluates the fitness in the presence of both *local* and *global* maxima.

$$Q(p_x, p_y) = 100 \cdot \left(1 - \frac{d_p}{d_m}\right) \quad (1)$$

$$Q(p_x, p_y) = 9 \uparrow \{0, (10 - d_p^2)\} + 10\left(1 - \frac{d_p}{d_m}\right) + 70\left(1 - \frac{d_n}{d_m}\right) \quad (2)$$

C. Dynamics

Dyanmics is described as the motion of bodies under the action of forces. Intuitively, our experiment utilizes common dynamic equations to update the position of the particles in the swarm - namely velocity held to a constant acceleration. Straightforward we may analyze position update as the sum of the previous position and the velocity in each direction (x and y).

$$p'_{x,y} = p_{x,y} + v_{x,y}$$

Here velocity is influenced by inertia (I); sociability (s) and cognition (c); the personal and global best positions ($f(x, y)$, $g(x, y)$ respectively); the position of the particle ($p_{x,y}$); and finally random variation (r_1, r_2) in the range of $(0, 1)$.

$$v'_{x,y} = I \cdot v_{x,y} + c\left[r_1 \cdot (f(p_x, p_y) - g(x, y))\right] + s\left[r_2 \cdot (g(x, y) - p_{x,y})\right]$$

The deviation from classic Dynamics is also observable such that each particle is influenced by personal and global best parameters. In effect, this provides guidance for each particle towards the best known position in the search space. Cognition and social coefficients respectively scale a particles response to seek a more optimal locality determinant of *their* best position, or the best position in the swarm.

D. Measurements

Measuring error is requisite for determining convergence to a solution; alloted to 1% in these experiments. Error calculations consists of determining the average error in each particles position in contrast to the global best. Additionally, the percentage of particles within a radius was measured. In these simulations, the radius was set to 5 units of distance. These equations are presented below:

$$
\begin{aligned}
P &= \{p_0, p_1, \cdots, p_k\} \\
\Omega(P) &= \{p_\omega | \exists p_\omega, p_k \in P | Q(p_\omega) > Q(p_k)\} \\
error(P) &= \sum_{k \in x,y}^{|P|} \sqrt{\frac{1}{2 \cdot |P|} \cdot (p_k - \Omega(P))^2} \\
radius(P) &= \frac{|\sum_{k \in x,y}^{|P|}[(25 > k^2_{x-maxima} + k^2_{y-maxima}) \cdot 0 + (25 \leq k^2_{x-maxima} + k^2_{y-maxima}) \cdot 1]|}{|P|}
\end{aligned}
$$

It is notable that as the size of the population increases, the validity of measurements becomes less applicable for radii measurements as no normalization is applied. Accordingly, the error measurment is in respective to the swarm as a whole without regards to maxima. However, these measurements need not be normalized as they are used qualitativley. In context, performance optimization in particle swarms is ill defined amongst researchers, thus this loose definition may suffice.

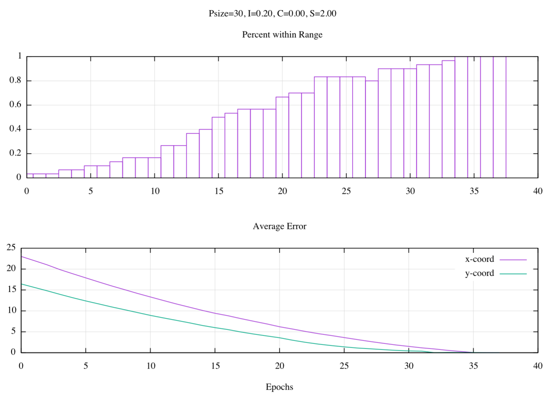
III. RESULTS & ANALYSIS

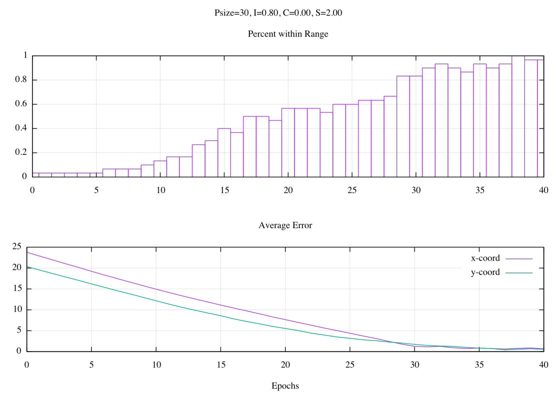
A. Filtering Optimal Parameters

Initially, enumeration of parameters with a medium to coarse granularity was used to prune simulations with unfavorable outcomes. Enumerating parameters yielded 1000 simulation result that where then filtered to approximately 20 favorable results. Oncemore, these simulations were then reevaluated and further filtered to aquire a handful of optimal parameter sets. After this aquisition process, each simulation was rerun with finer variation in parameters to grasp an understanding of how each parameter affected the outcome of the simulation. This approach is analogous to a top-down refinement process.

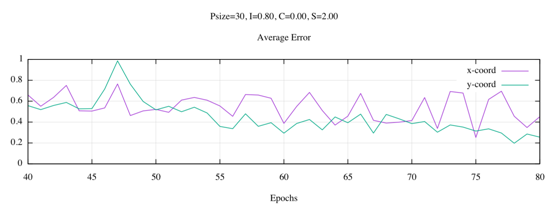
B. Inertia

Inertia is the resistance of any physical object to change its speed, direction, or state of rest. Furthermore it describes the tendency of objects to keep moving in a straight line at constant velocity. In the experiment, higher inertias were associated with more epochs until convergence - given otherwise identical parameters.

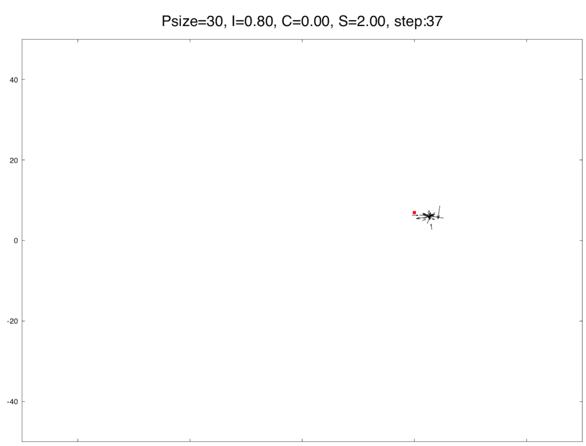


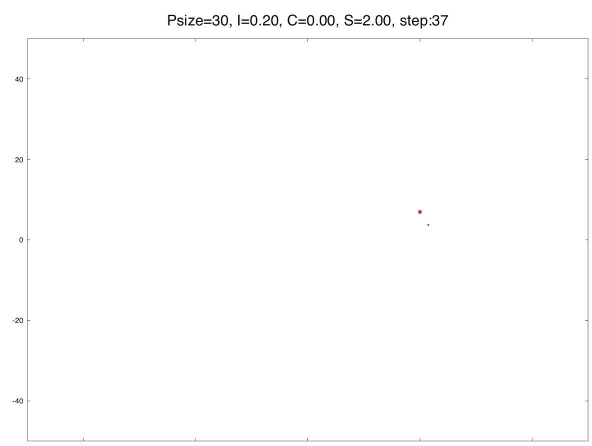


Oscillations are also prevalant in the average error of the particles. Pressumably this behavior is resultant of a particle's resistance to change direction upon encountering the global best coordinates. Below shows a continuation of the above graphs for inertia equal to 0.8 drawing light to these oscillations.



Observing both swarms plotted in a two dimensional space at identical steps we can identify below that 0.8 exhibits a more irratic behavior as it attempts to fight the larger inertia. Furthermore 0.2 has a significantly tighter grouping at identical epochs.

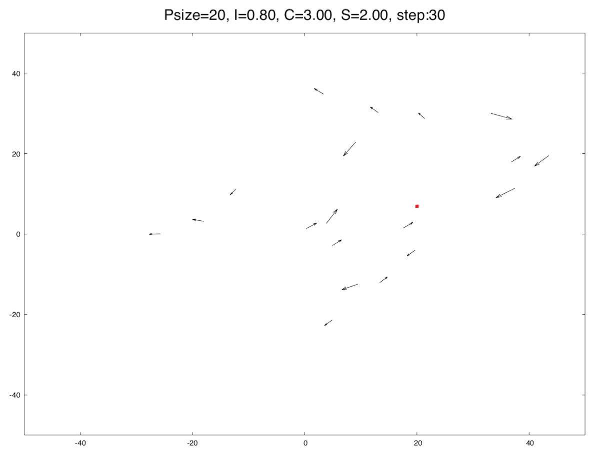


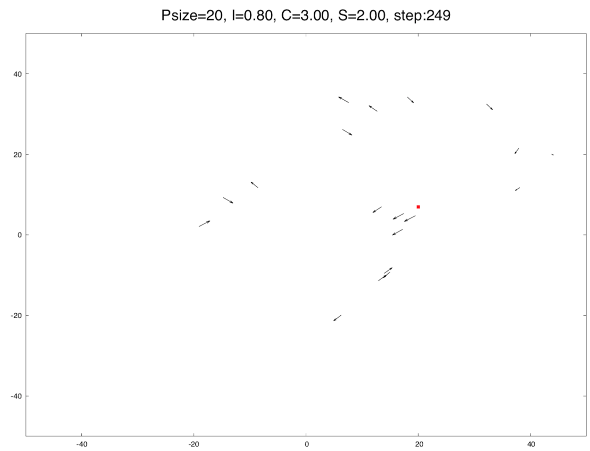


Visibility of osillatory behavior is much more prevalent while observing the swarms in motion. I invite you to view the animations in `overview.html` attached with this report to gain a more insightful understanding of inertial affects on otherwise identical swarms. This behavior is also exemplary of inertial affects in the presence of both local and global maxima. Overall, lower inertial rates were consistently associated with faster rates of convergence.

C. Cognition & Sociability

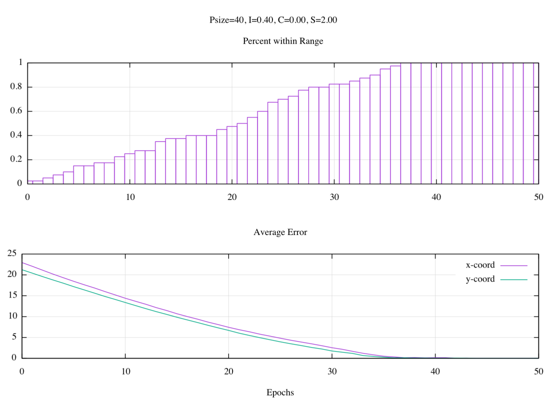
In every filtered experiment, these parameters were modified in an attempt to jitter more optimal behavior. Without fail, an increase in cognition resulted in irratic behavior for each particle. Given the velocity equation presented in subsection **Dynamics** one may conjecture that raising the cognition effectively generates a distrust amongst the particles concerning an optimum. This "distrust" manifests itself as each particle continually oscillates between what it percieves to be the optimum and that of the group perception. For instance, below we see a simulation with high cognition where the *30th* epoch of the two dimensional space is almost indistinguishable from the expired simulation (*249th* epoch).

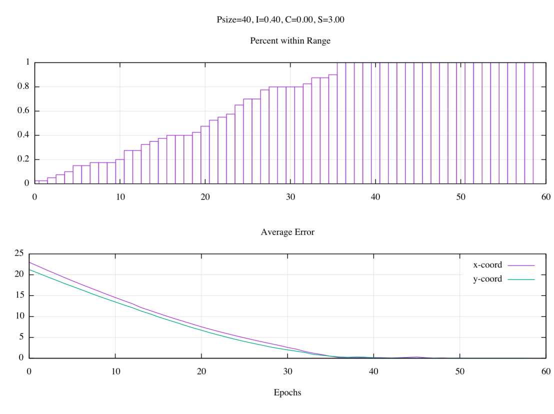




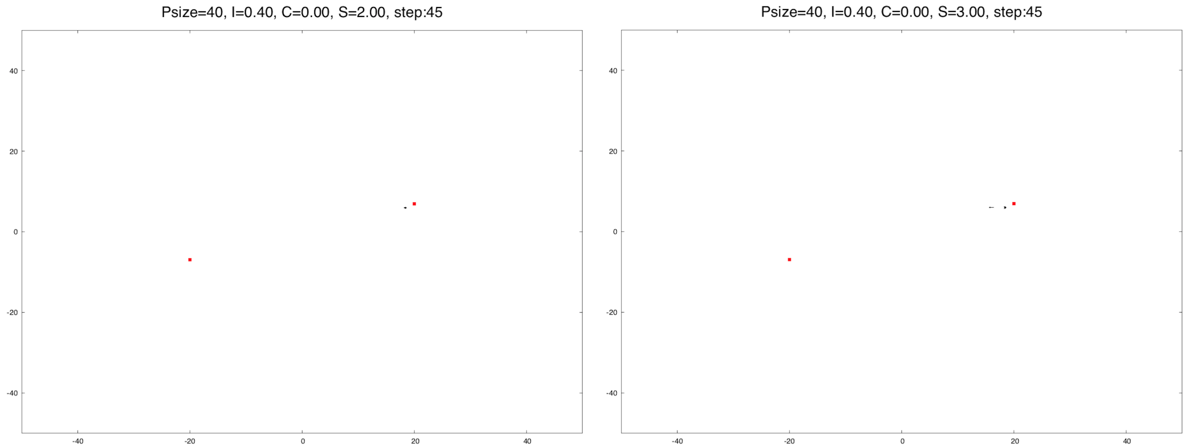
This "double minded" behavior is significantly more prevalent for increased cognition. Hence, one may confidently postulate by zeroing the cognition, we may effectively eliminate the effects of *local* maxima detering from obtaining a *global* maxima. Again, `overview.html` displays multiple example animiations asserting this claim.

The social coefficient regularly resulted in behavior similar to high inertial values in the absence of a cognition parameter. That is, higher social parameters were associated with more epochs until convergence - given otherwise identical parameters. Similarly, oscillatory behavior presented itself with greater sociability. These effects were significantly less drastic however, as they typically would not inhibit the convergence of the population. For example, below it is observable that the only differing parameter is sociability.





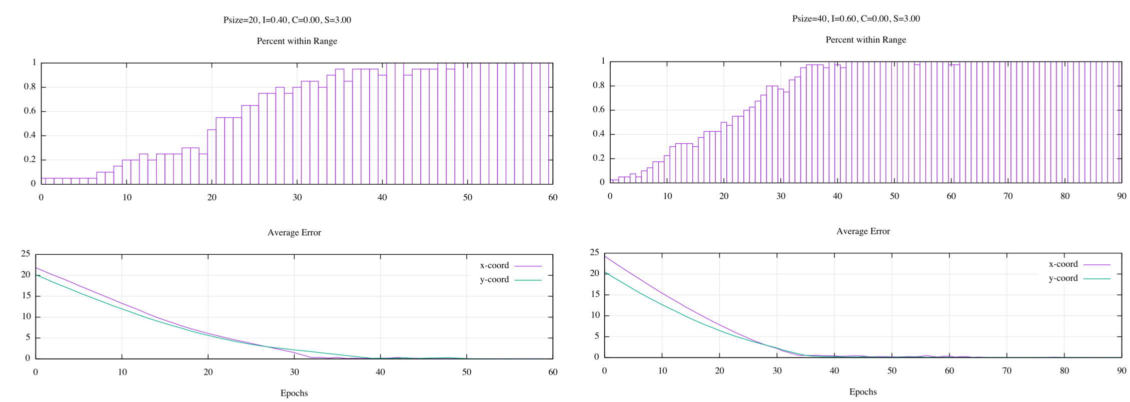
The *45th* epoch of the 3.0 simulation exhibits the oscillatory behavior specified. Additionally upon evaluating the two dimensional space for both swarms below we may observe a tighter grouping at identical epochs for the lower social parameter.



Although it may be unclear here, the leftmost particle in the higher social experiment orbits the swarm multiple times before converging. To observe this phenomenon more clearly, see `overview.html`. Additionally, the lower cognition parameter prevents the swarm from grouping around the *local* maxima as aforementioned.

D. Population Size

Of the initial 1000 enumerated parameters, no populations less than twenty exhibited behavior acceptable for continuation in the evaluation process. In this context, acceptable behavior is defined as either converging to 1% error, or spacial locality within five units of distance from the global maxima. Often, simulations that did converge would be far too distant from the maxima to be considered a feasable solution. In the presence of both local and global maxima, populations below forty incurred identical problems. One postulate for this behavior is that more particles are required to offset being drawn to the local maxima. However, simulations with similar parameters asside from population size merit the proposition that larger populations converge less rapidly than smaller ones. This proposal intuitively draws from the fact that fewer particles need be within a radius for convergence. The instance below is a possible evidence for such an assertion.



IV. CONCLUSION

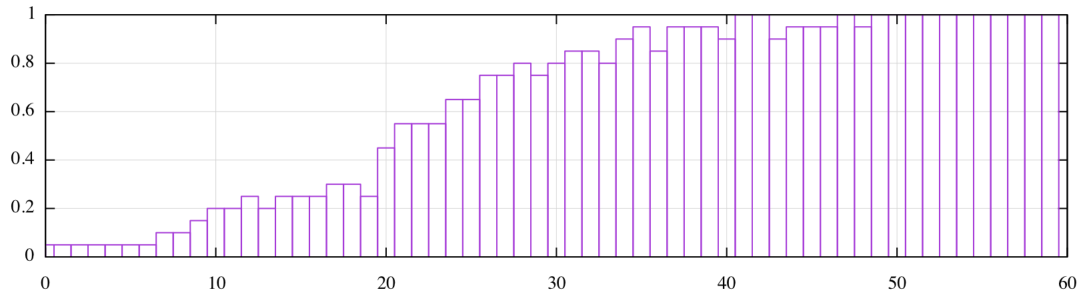
It is evident when lower inertia is present in the population with low cognition, overall more optimal solutions materialize. More specifically, cognition seems to be detrimental to the population as a whole. As well, the social parameter is typically associated with more stability in smaller populations with the caveat of a few additional epochs during oscillation. The increase in population seems to cause greater likelihood the swarm will converge in the presence of local maxima especially as inertia and cognition are held at lower values.

V. ATTACHMENTS

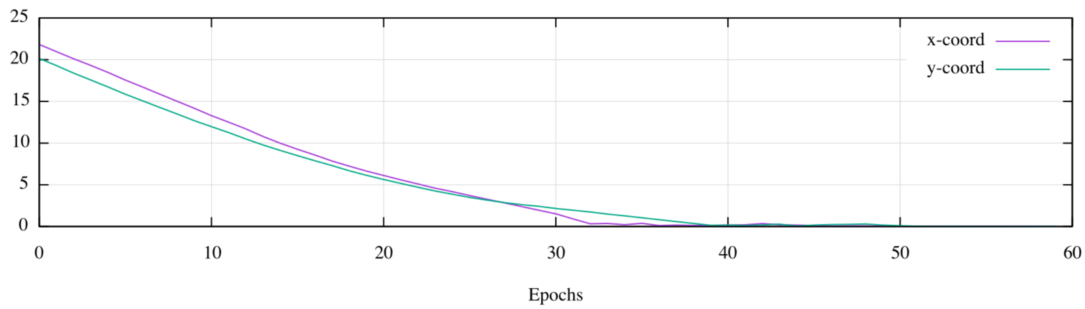
overview.html

Psize=20, I=0.40, C=0.00, S=3.00

Percent within Range

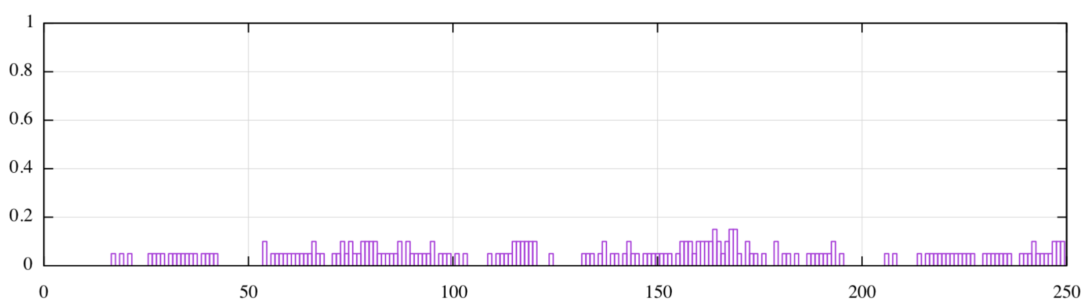


Average Error

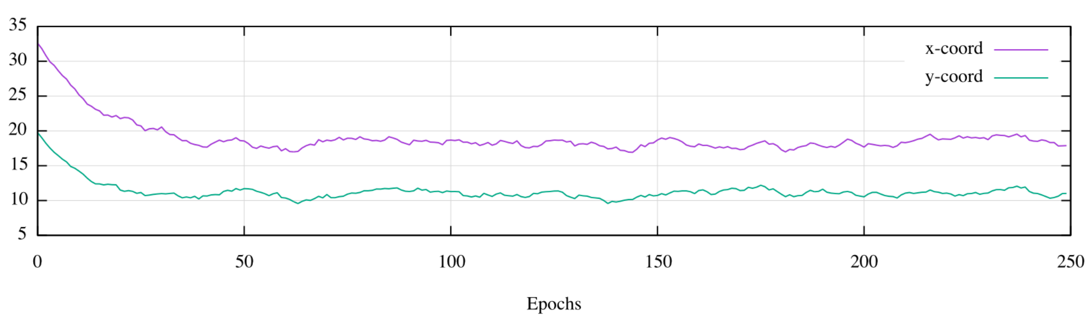


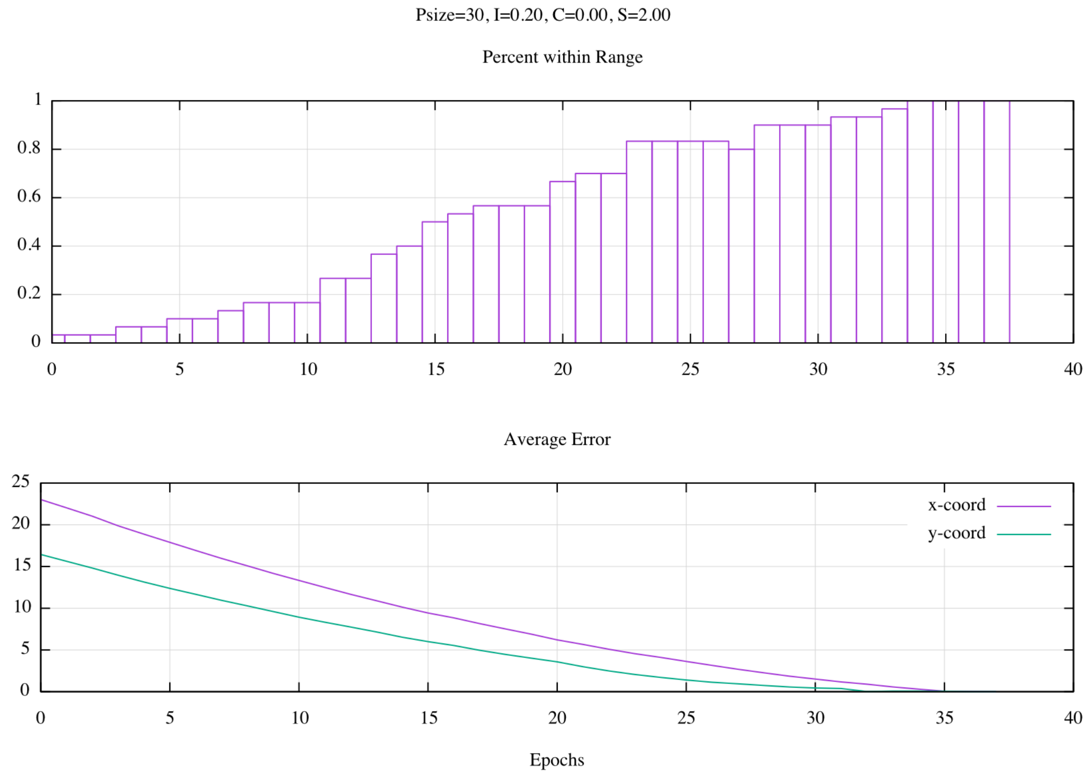
Psize=20, I=0.80, C=3.00, S=2.00

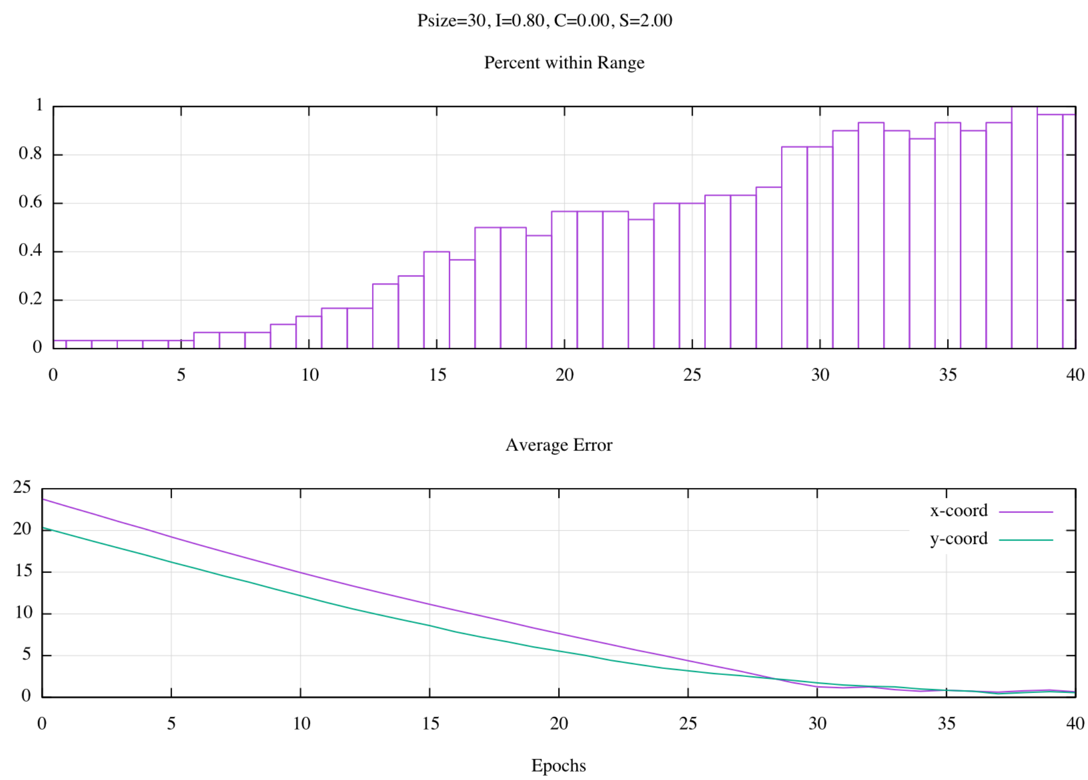
Percent within Range



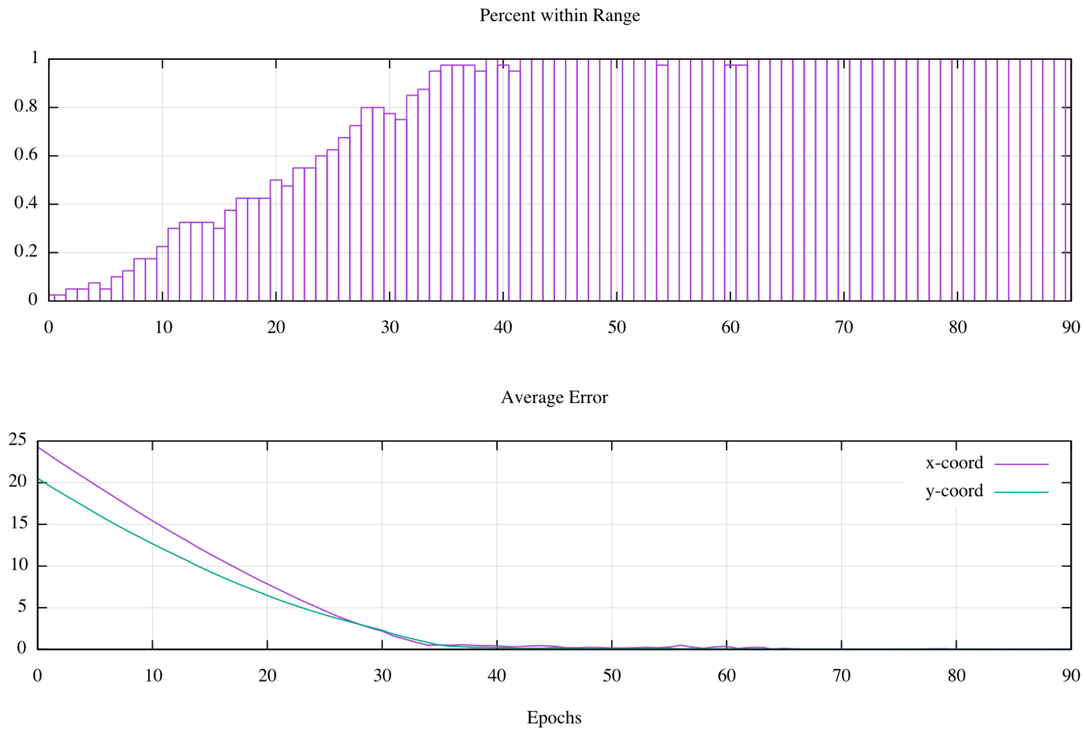
Average Error



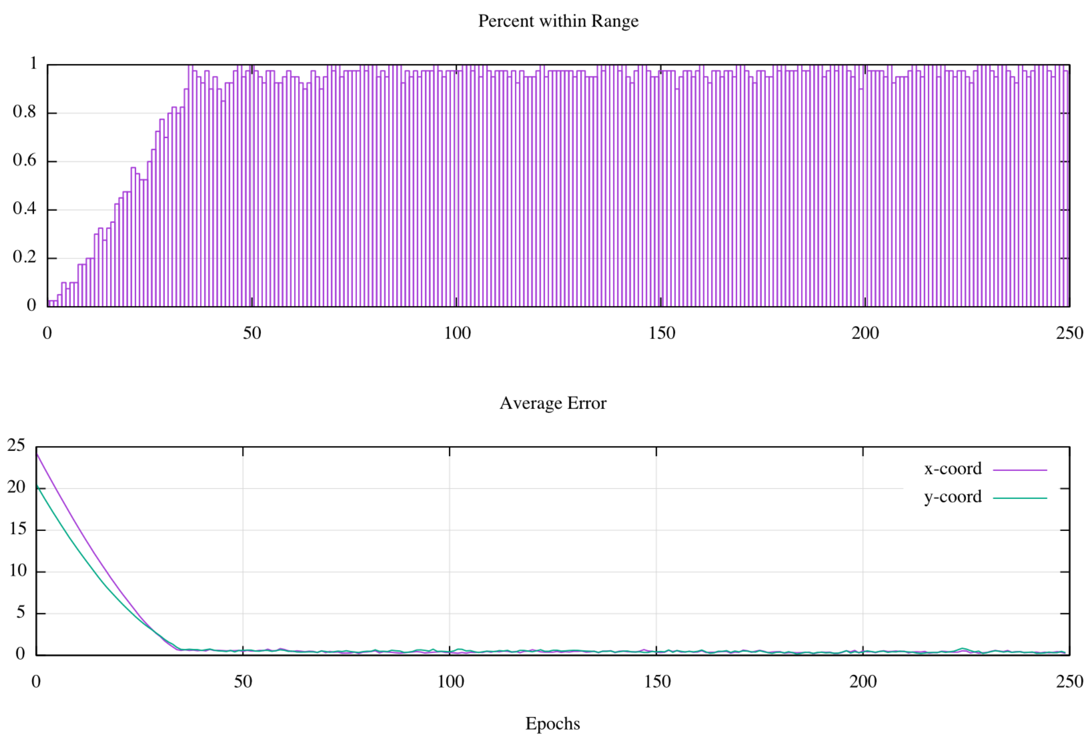

Psize=30, I=0.20, C=0.00, S=2.00
Percent within Range
Average Error
x-coord
y-coord
Epochs


Psize=30, I=0.80, C=0.00, S=2.00
Percent within Range
Average Error
x-coord
y-coord
Epochs

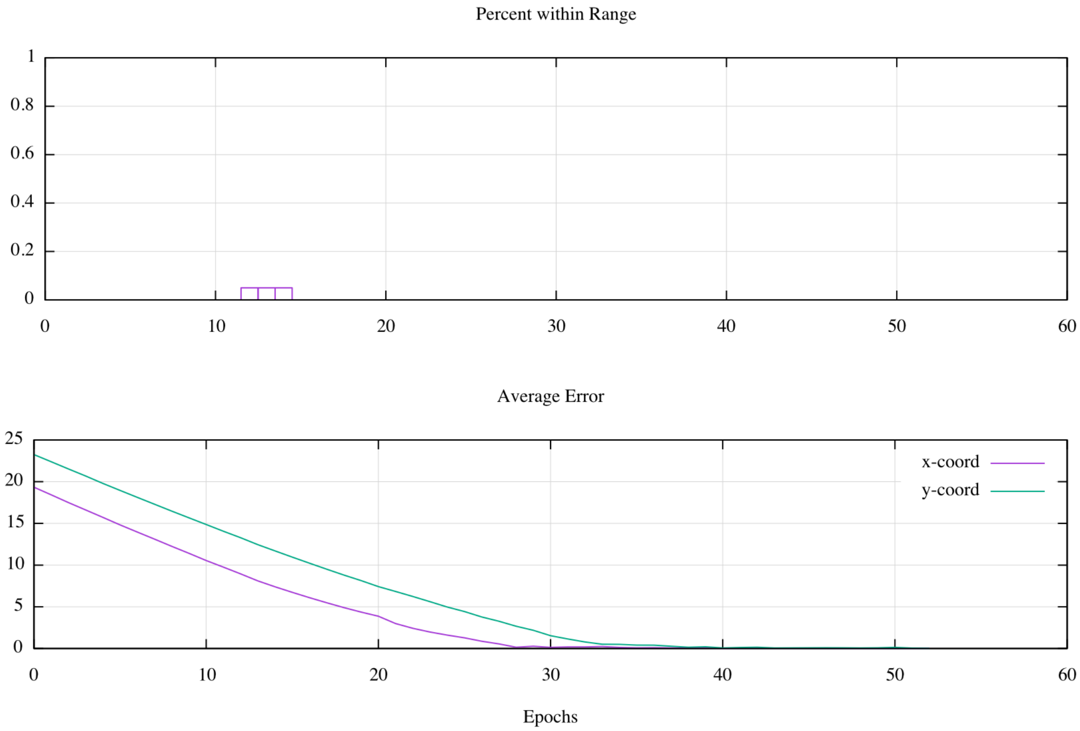
Psize=40, I=0.60, C=0.00, S=3.00



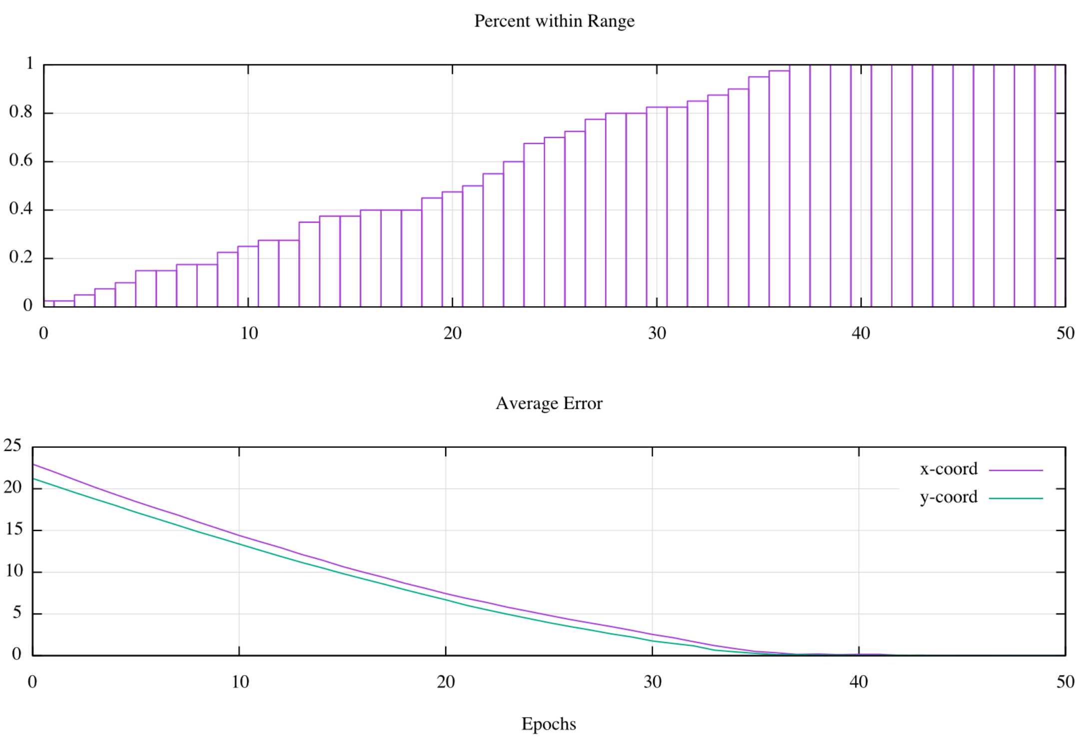
Psize=40, I=0.60, C=0.00, S=4.00

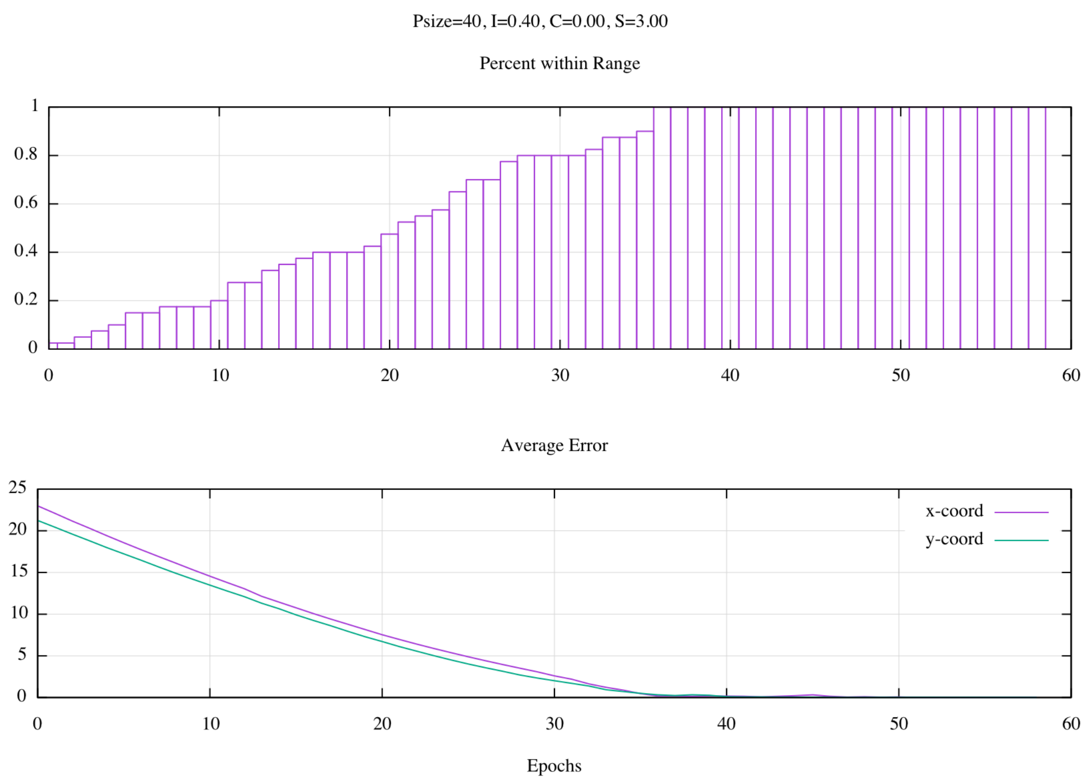


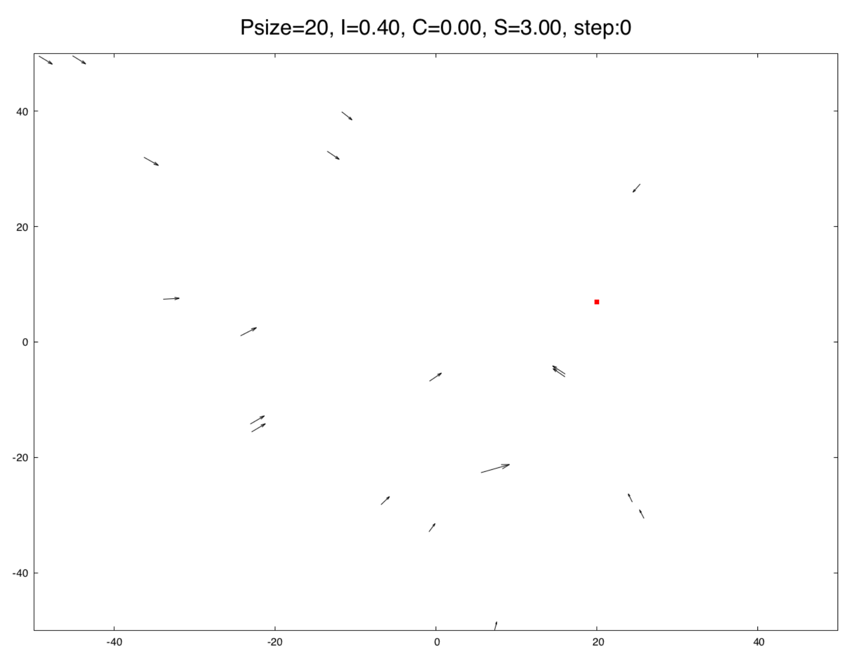
Psize=20, I=0.40, C=0.00, S=3.00



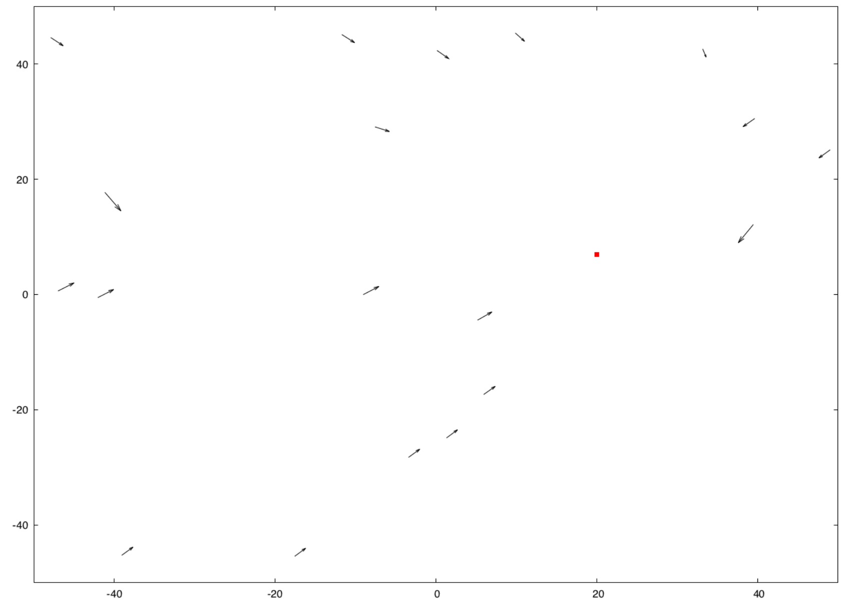
Psize=40, I=0.40, C=0.00, S=2.00



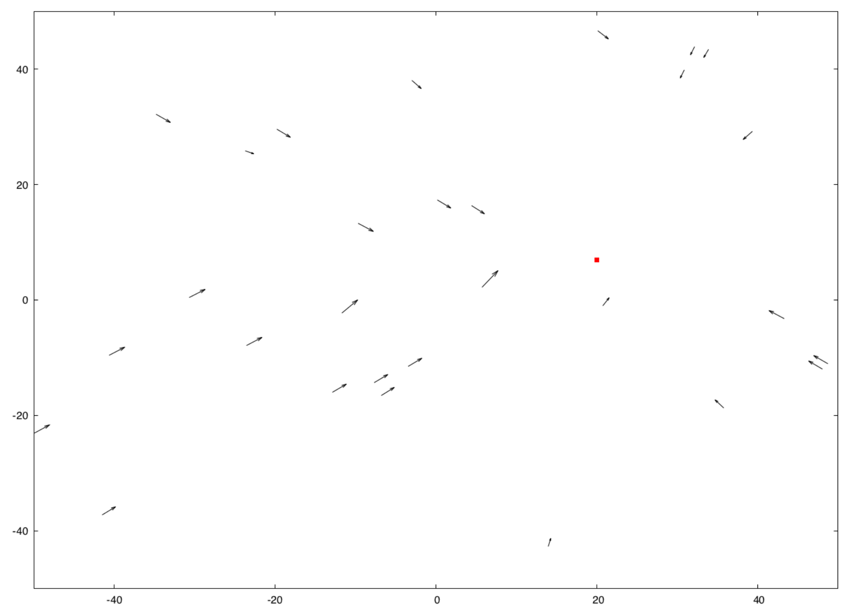

Psize=40, I=0.40, C=0.00, S=3.00
Percent within Range
Average Error
x-coord
y-coord
Epochs

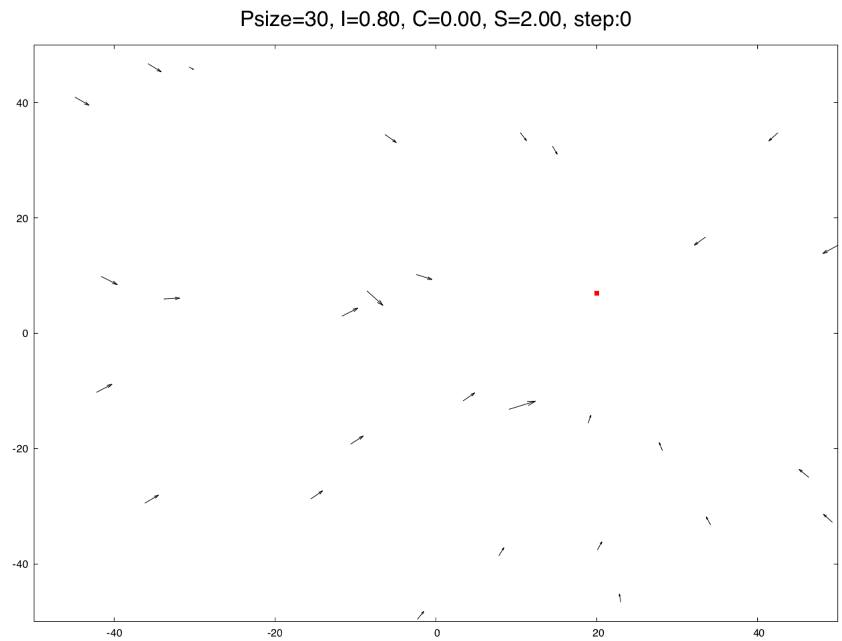

Psize=20, I=0.40, C=0.00, S=3.00, step:0

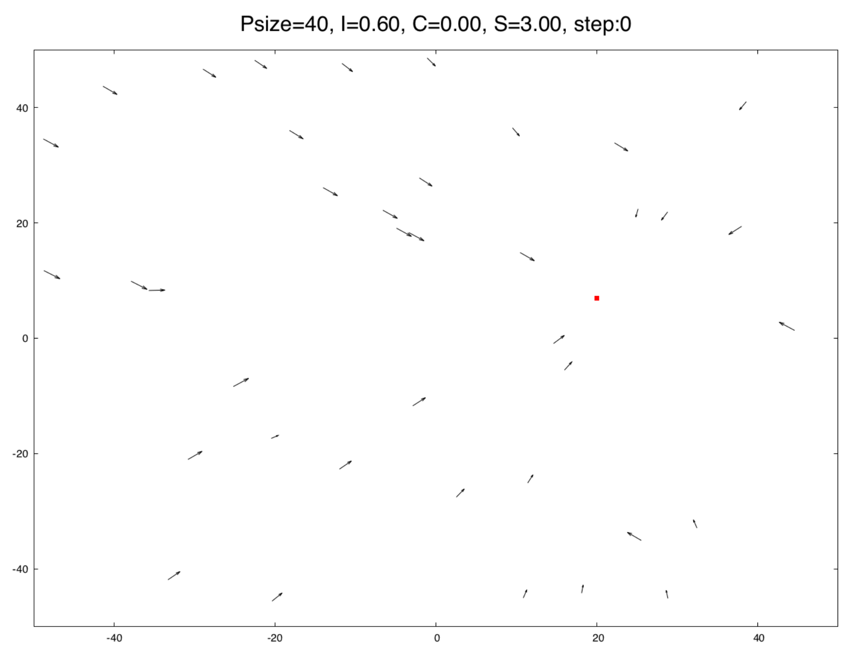
Psize=20, I=0.80, C=3.00, S=2.00, step:0

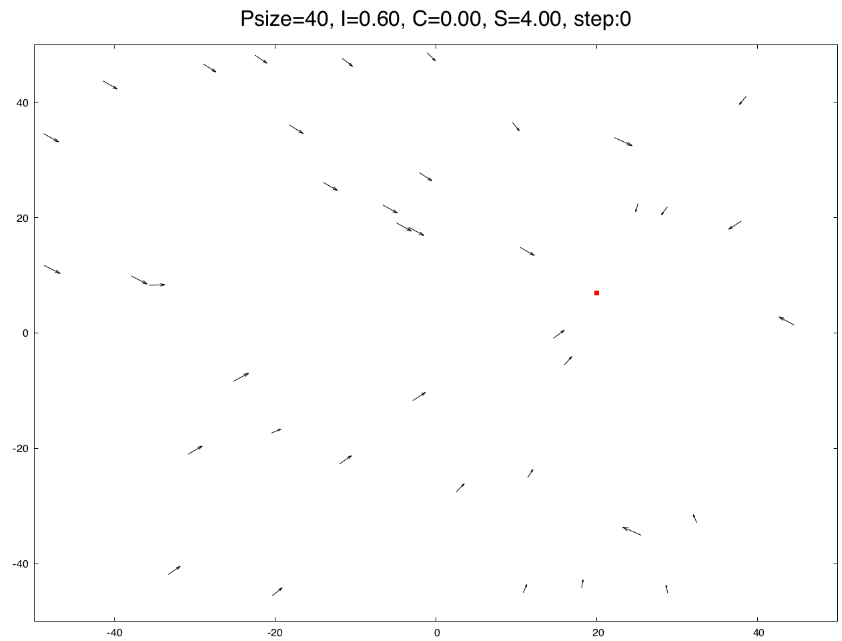


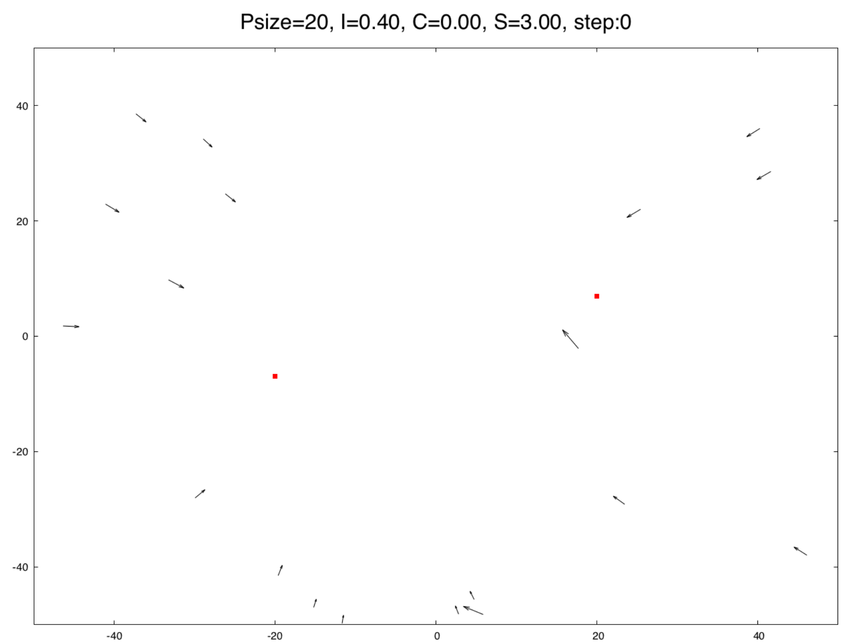
Psize=30, I=0.20, C=0.00, S=2.00, step:0



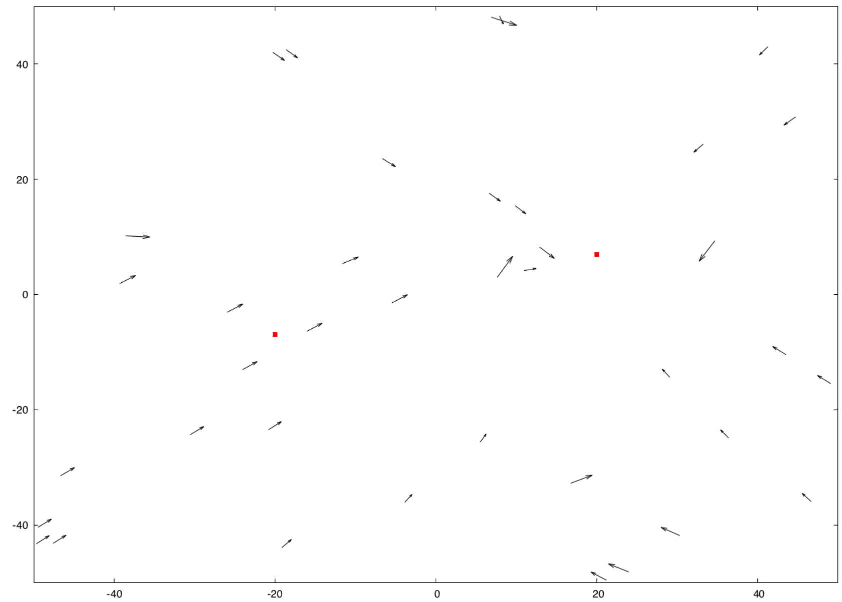

Psize=30, I=0.80, C=0.00, S=2.00, step:0
40
20
0
-20
-40
-40
-20
0
20
40


Psize=40, I=0.60, C=0.00, S=3.00, step:0
40
20
0
-20
-40
-40
-20
0
20
40


Psize=40, I=0.60, C=0.00, S=4.00, step:0
40
20
0
-20
-40
-40
-20
0
20
40


Psize=20, I=0.40, C=0.00, S=3.00, step:0
40
20
0
-20
-40
-40
-20
0
20
40

Psize=40, I=0.40, C=0.00, S=2.00, step:0



Psize=40, I=0.40, C=0.00, S=3.00, step:0