

# BOIDS SIMULATION USING SCICOS

## BATCH SIMULATION IN SCILAB USING XCOS\_SIMULATE

Graph Models and Simulation (Seminar course)

15.08.2016

Aditya Raj



# Outline

# Outline

- Problem Statement

# Outline

- Problem Statement
- Introduction and Background

# Outline

- Problem Statement
- Introduction and Background
- Boids Model in Scicos

# Outline

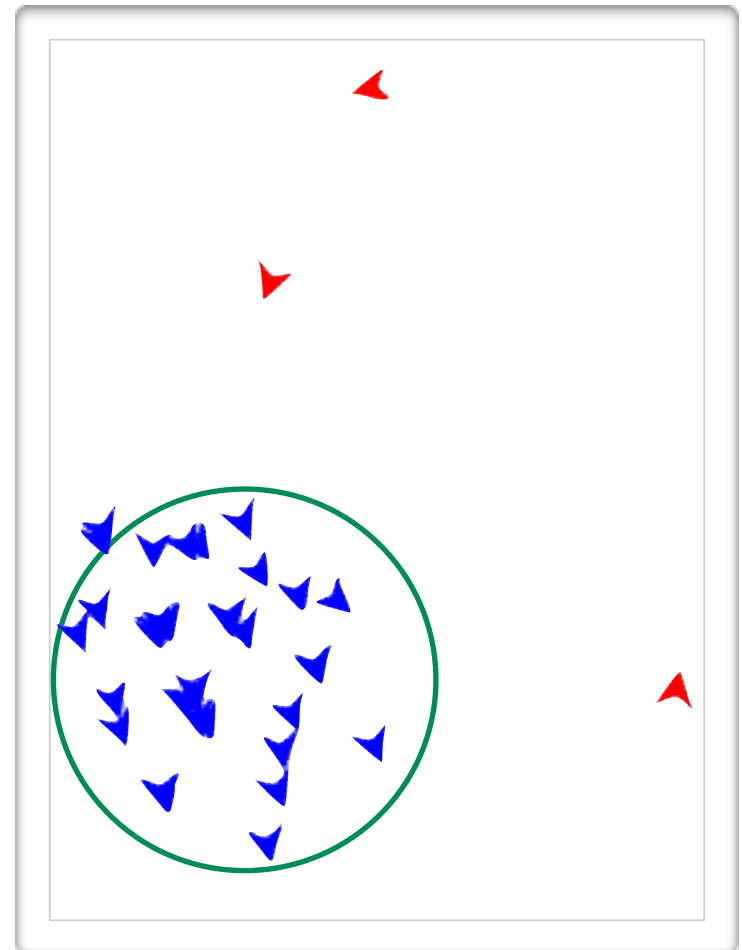
- Problem Statement
- Introduction and Background
- Boids Model in Scicos
- Simulation Results & Evaluation

# Outline

- Problem Statement
- Introduction and Background
- Boids Model in Scicos
- Simulation Results & Evaluation
- Improvements

# Problem

- Simulate Boids behaviour in Scicos.
- In short, to demonstrate an Agent Model using the following Scilab features
  - Program new blocks in C
  - Run simulations in batch mode from Scilab environment





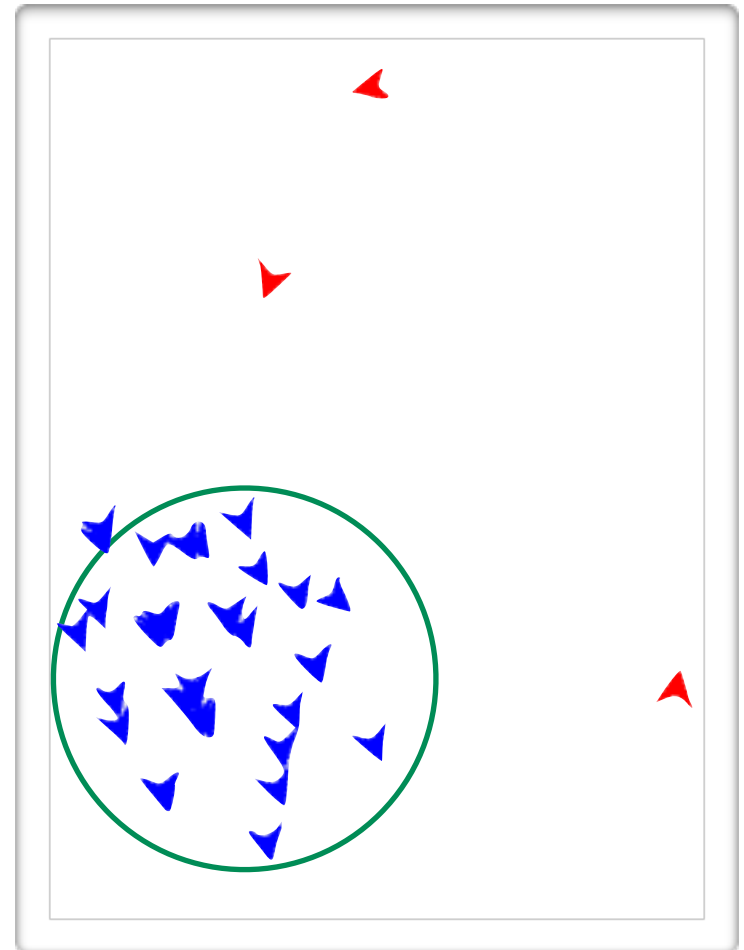
## Introduction -> Scicos

The screenshot displays the Scilab 5.5.2 interface with the Xcos environment. The main window shows a block diagram of a control system. The 'Palette browser - Xcos' window is open, showing various blocks like ANDBLK, BCSOM\_f, CMSCOPE, CONST\_m, CONVERT, CSCOPXY, DEMUX, DOLLAR\_f, INTEGRAL\_f, IN\_f, LOGICAL\_OP, MUX, PRODUCT, RELATIONALOP, SATURATION, and SWITCH2\_m. The 'batch\_simulation' window shows a block diagram with a sine wave input, a gain block, a summing junction, a counter, and a termination block. The 'trialsum.sce' script is visible in the bottom right, defining a function to calculate the absolute value of a vector.

```

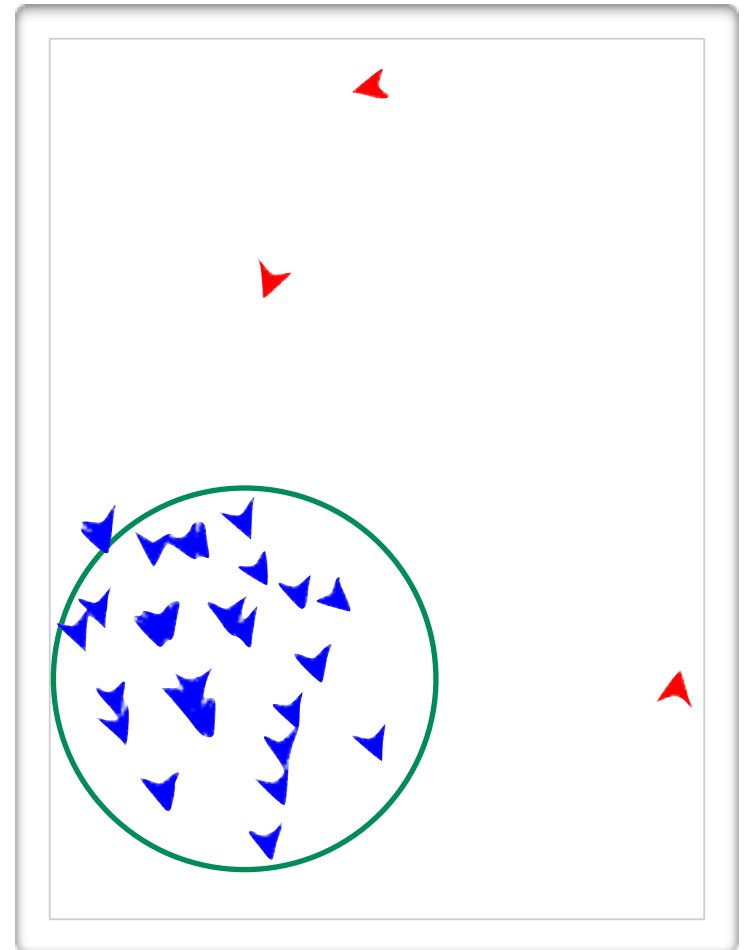
1 //Calculate the absolute value of
  a vector
2 function res= calculate_abs(a)
3     res = sqrt(a(1)^2 + a(2)^2)
4 endfunction
5
6 importXcosDiagram("SCI/modules/xco
  s/demos/ABM/final_combined_block.x
  cos")
7
8 num_agents = 6;
9 pos = floor(10*rand(2,num_agents))
10;
11 vel = floor(5*rand(2,num_agents));
  
```

# Introduction -> BOIDS: bird-like object



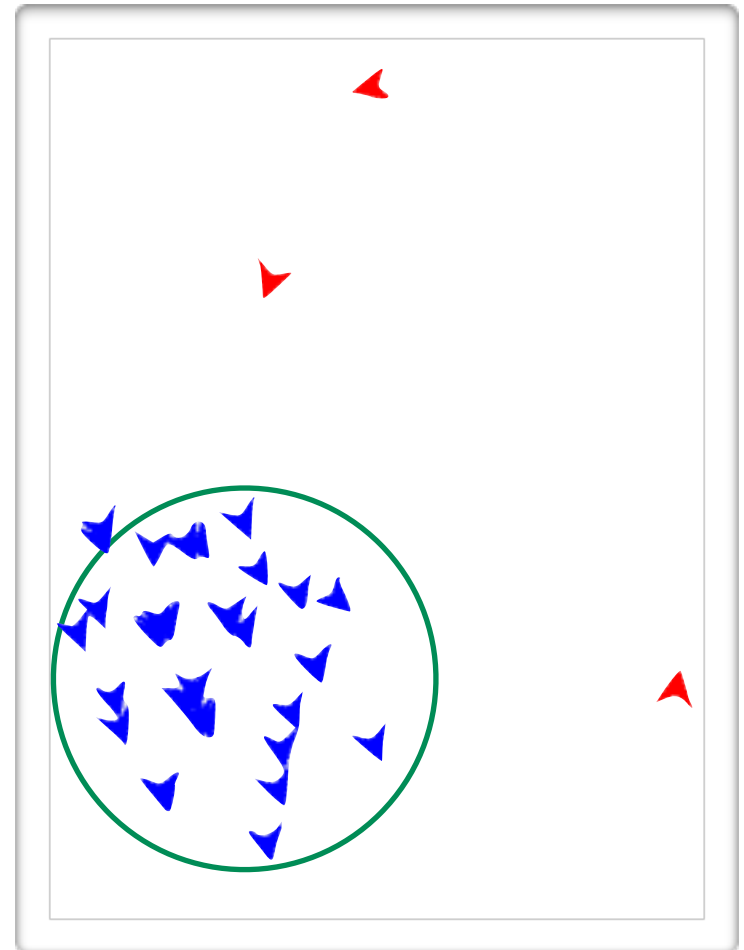
# Introduction -> BOIDS: bird-like object

- Developed by Craig Reynolds in 1986



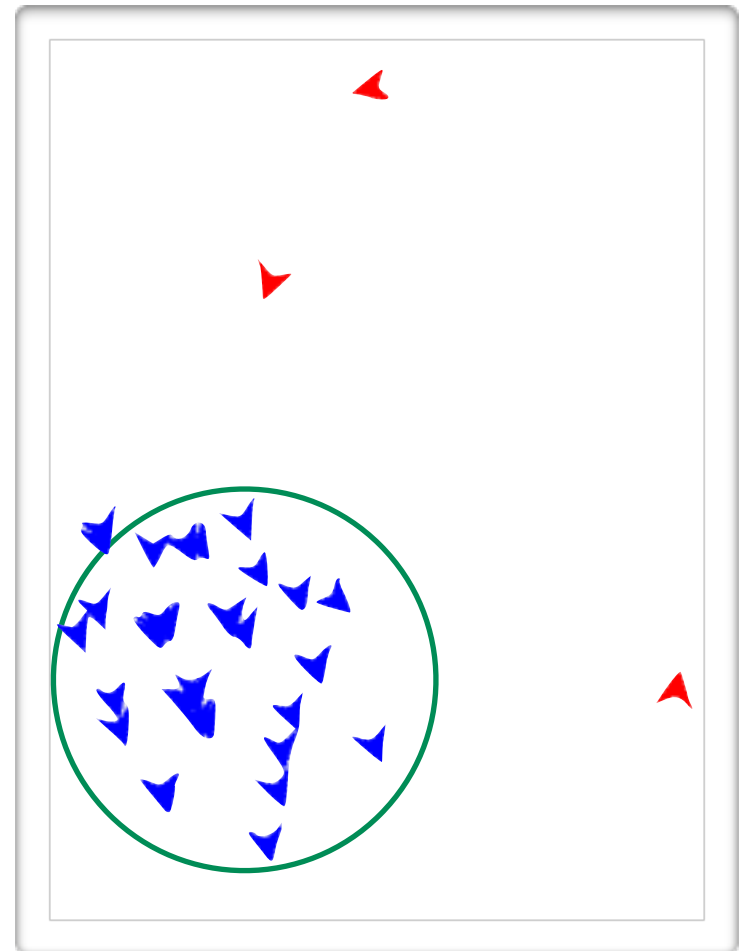
# Introduction -> BOIDS: bird-like object

- Developed by Craig Reynolds in 1986
- A simple agent-based model



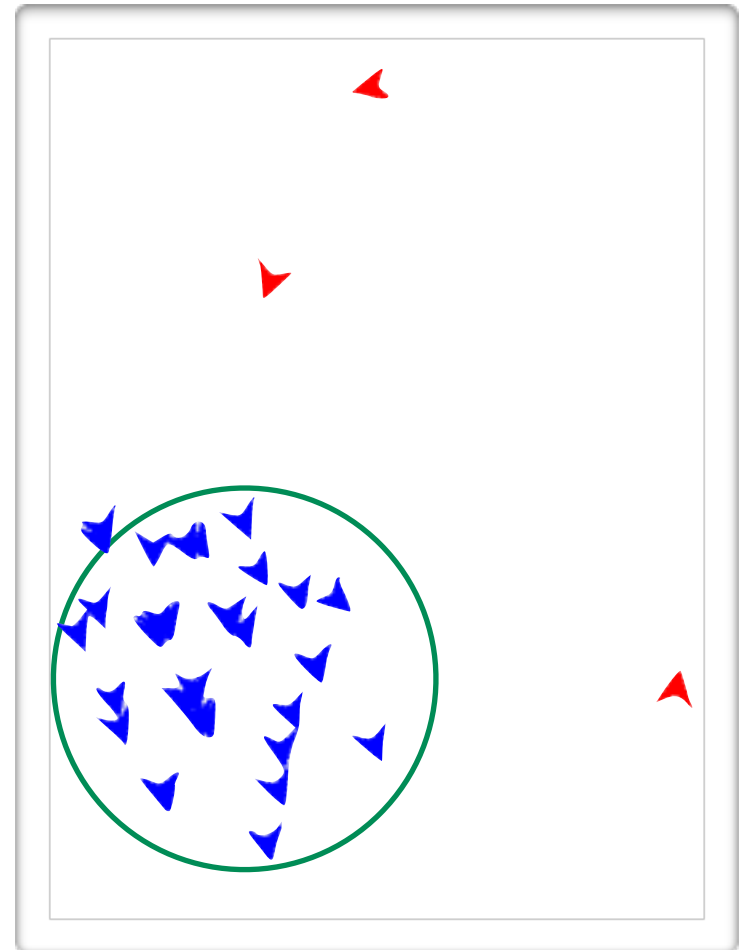
# Introduction -> BOIDS: bird-like object

- Developed by Craig Reynolds in 1986
- A simple agent-based model
- Simulates the flocking behaviour of birds

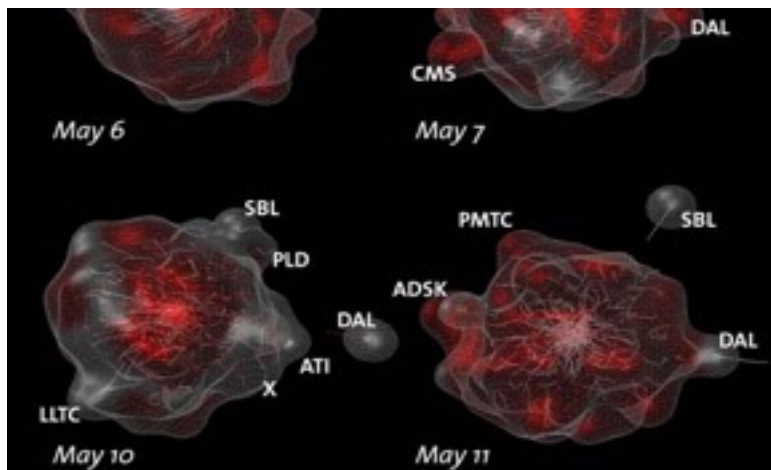


# Introduction -> BOIDS: bird-like object

- Developed by Craig Reynolds in 1986
- A simple agent-based model
- Simulates the flocking behaviour of birds
- The agent checks other agents in its neighbourhood
  - And further **ALIGN**, **SEPARATE** and **COHERE**.



# BIDS: Application Areas



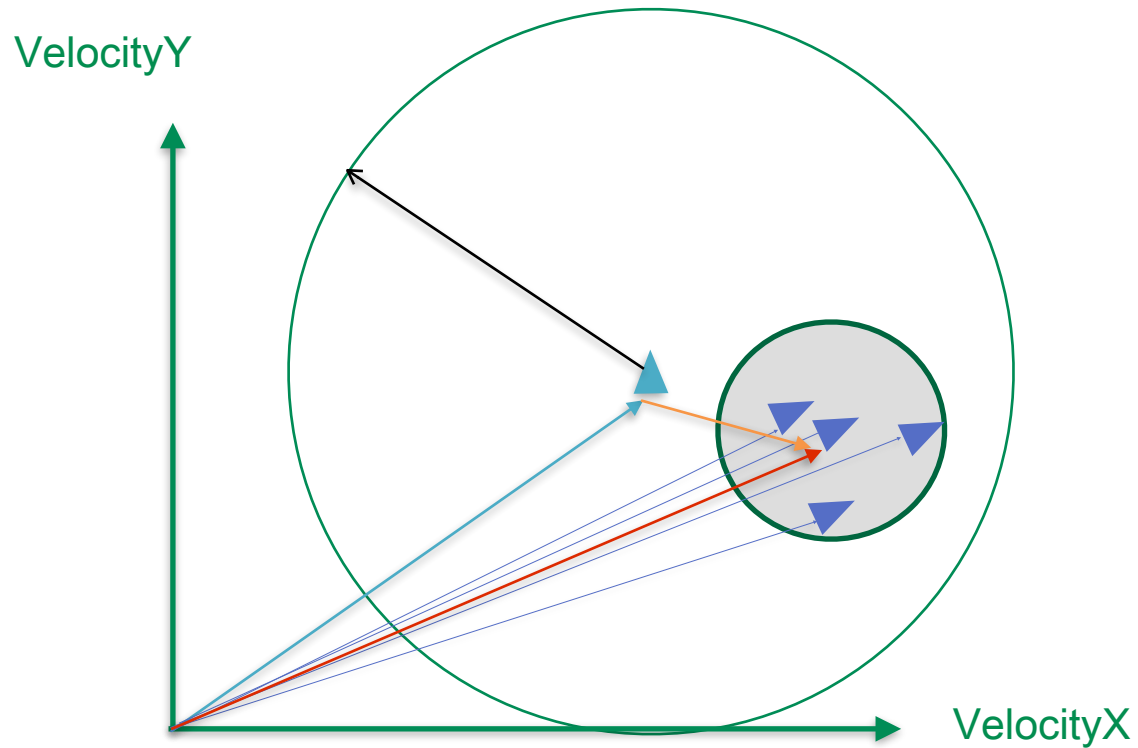
# BACKGROUND

## PHYSICS BEHIND BOIDS MOTION

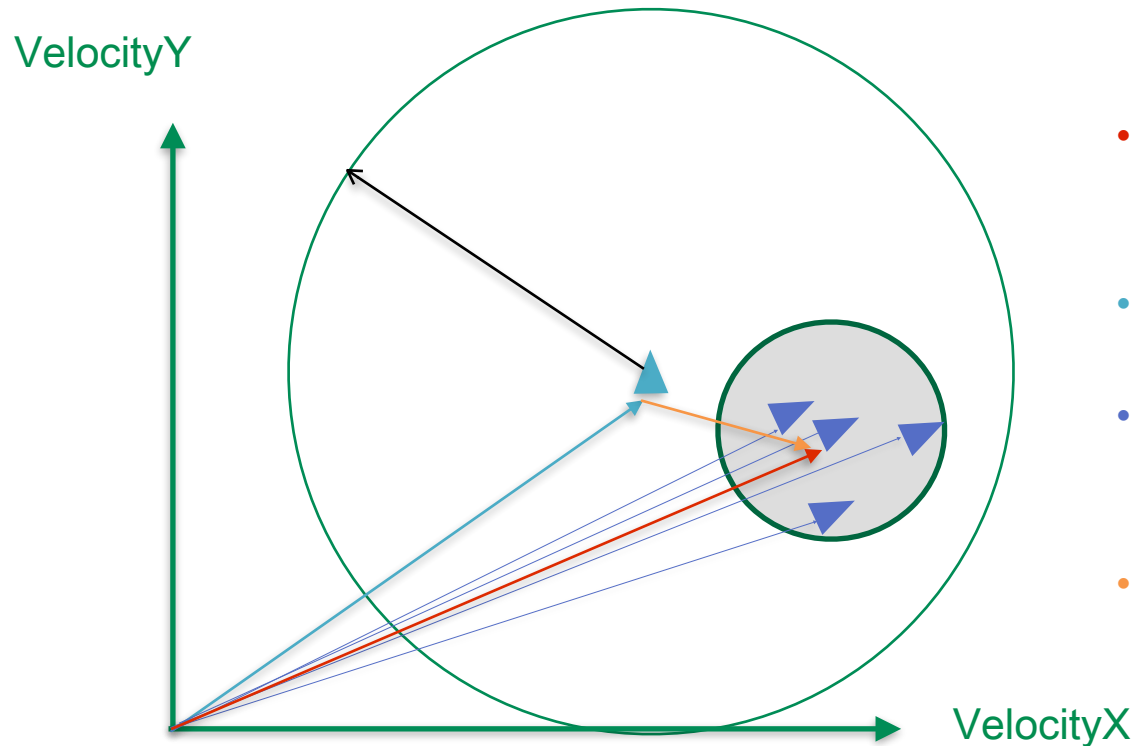


# Boids Model - Alignment

# Boids Model - Alignment



# Boids Model - Alignment



- Average Heading of the flocks
- Current Position Vector
- Actual Heading of individual bonds
- Resultant Vector

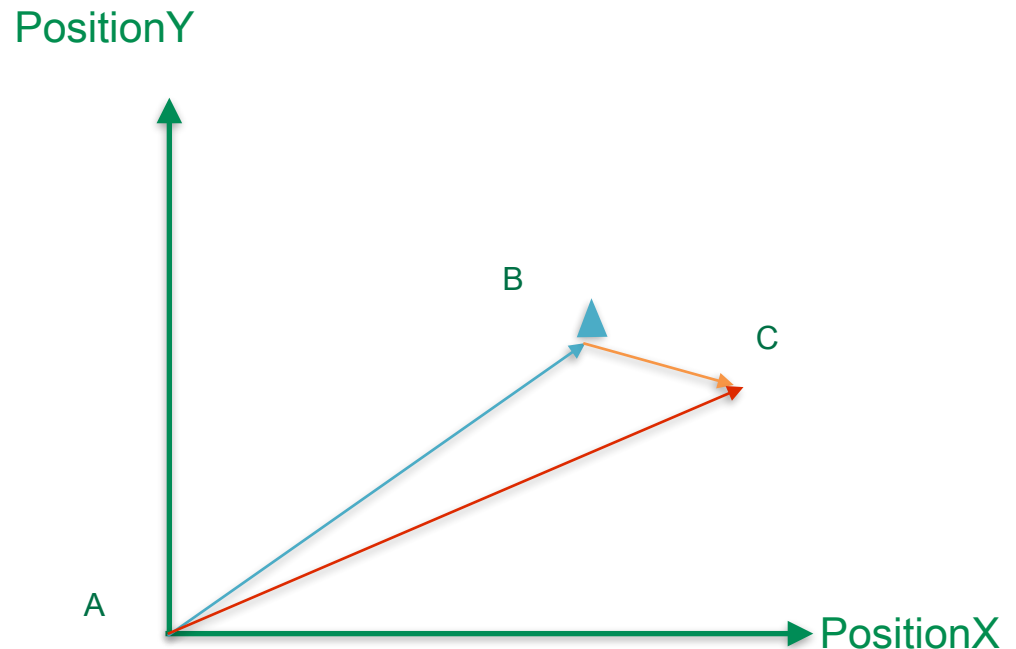
# Boids Model - Alignment

# Boids Model - Alignment

- For every nearby boid in the system, calculate the average velocity
- Normalise it
- Calculate Steer/Resultant  
 $AB + BC = AC$  By vector law of addition

# Boids Model - Alignment

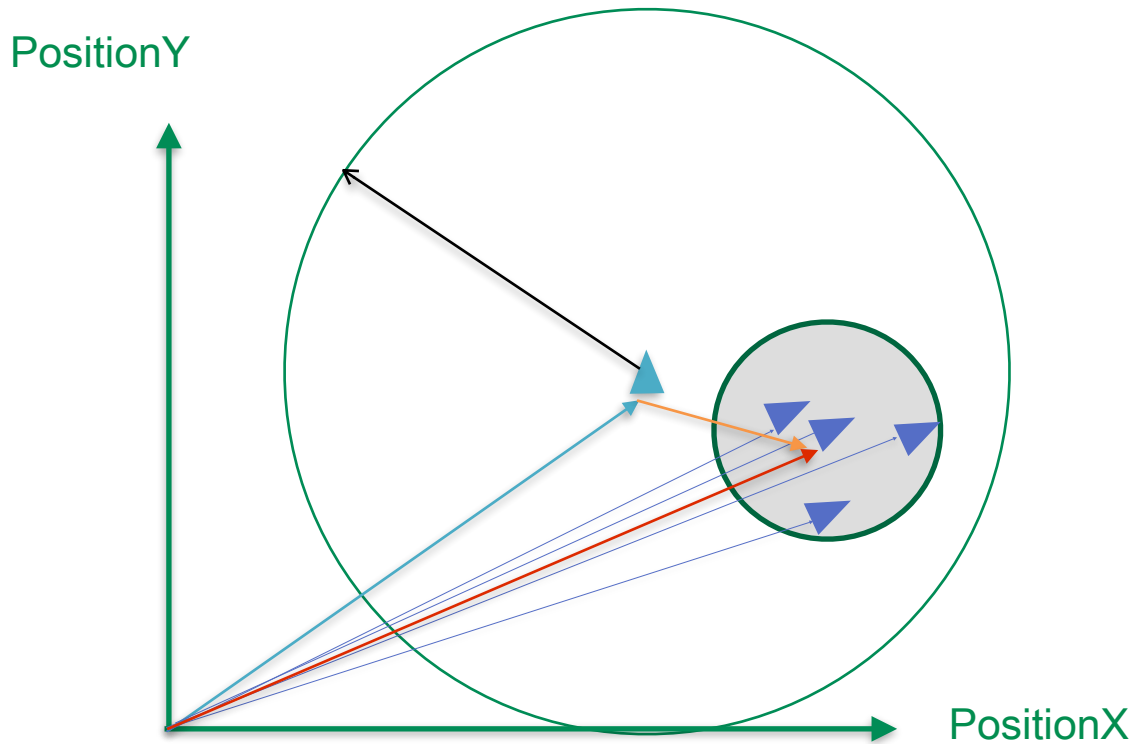
- For every nearby boid in the system, calculate the average velocity
- Normalise it
- Calculate Steer/Resultant  
 $AB + BC = AC$  By vector law of addition





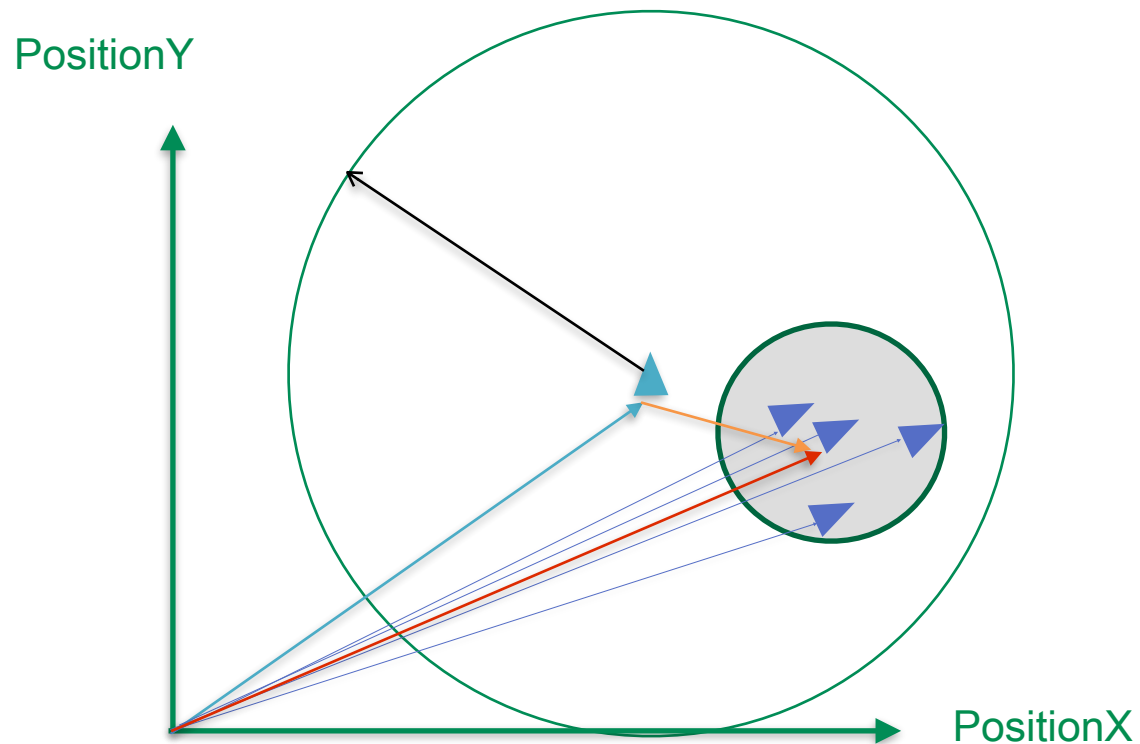
# Boids Model - Cohesion

# Boids Model - Cohesion





# Boids Model - Cohesion



- Average Heading of the flocks
- Current Position Vector
- Actual Heading of individual bonds
- Resultant Vector
- Flocking Radius

# Boids Model - Cohesion

# Boids Model - Cohesion

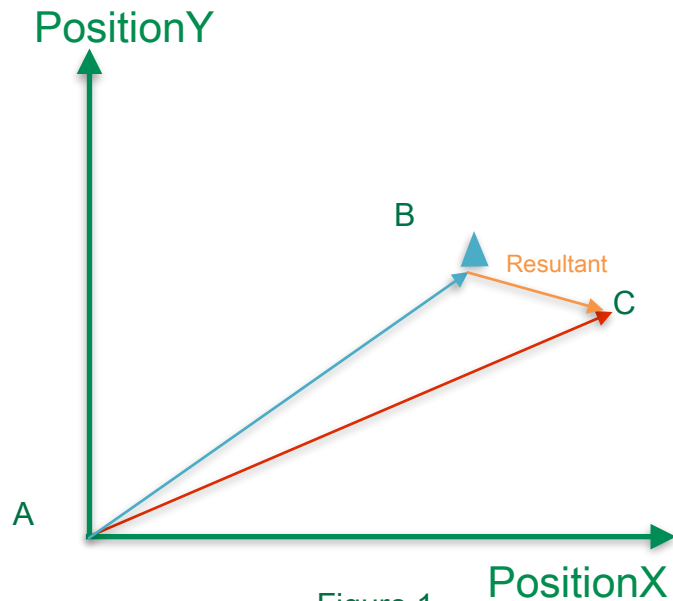
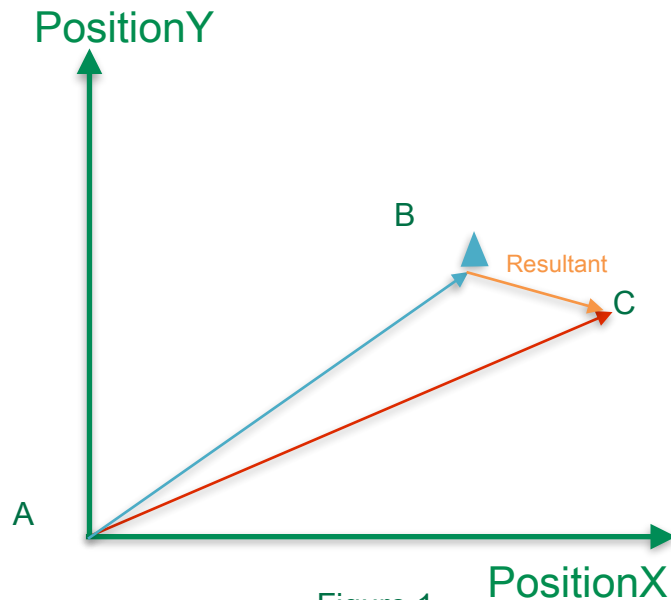


Figure 1

# Boids Model - Cohesion



- For the average location (i.e. Center) of all nearby boids
- Calculate a vector pointing from the location to the target
- Normalise it

# Boids Model - Cohesion

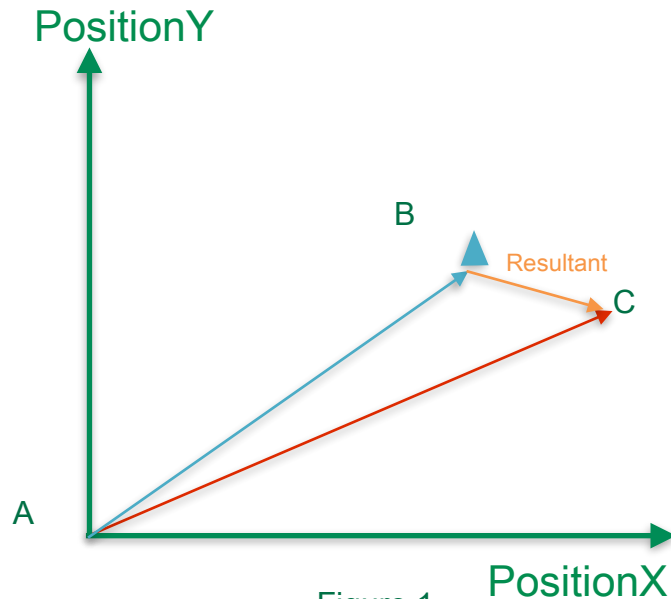


Figure 1

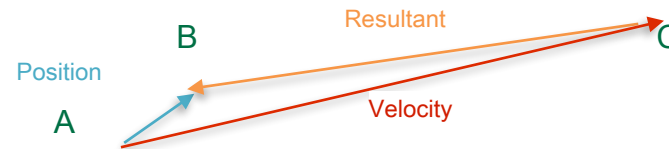


Figure 2

- For the average location (i.e. Center) of all nearby boids
- Calculate a vector pointing from the location to the target
- Normalise it

# Boids Model - Cohesion

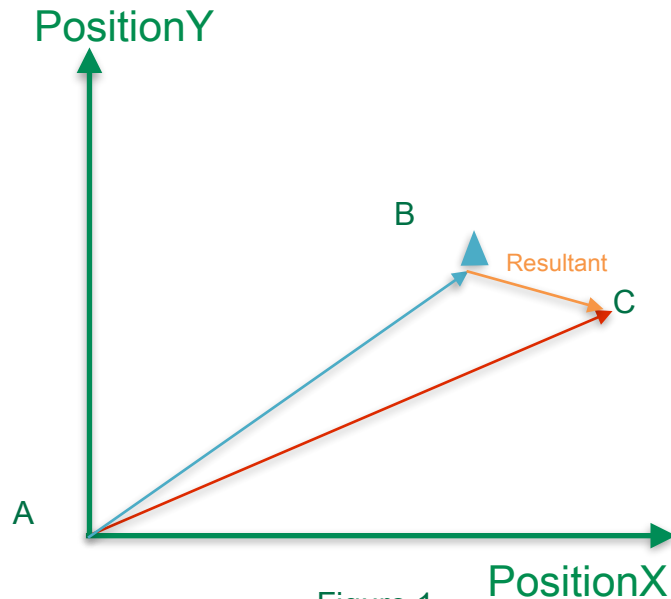


Figure 1

- For the average location (i.e. Center) of all nearby boids
- Calculate a vector pointing from the location to the target
- Normalise it

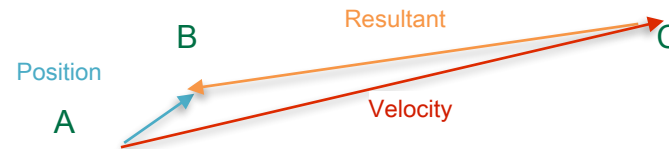
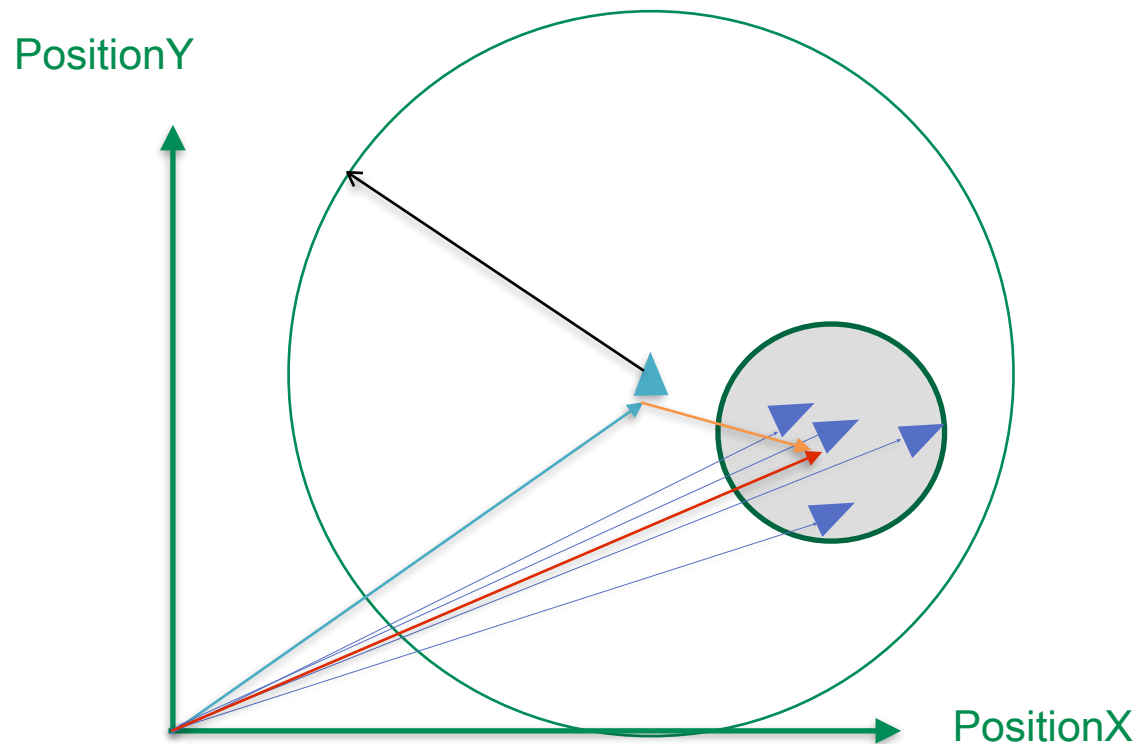


Figure 2

- Calculate steering vector towards that location (Steering = Desired - Velocity)
- Steering Vector
- Resultant = Position - Velocity

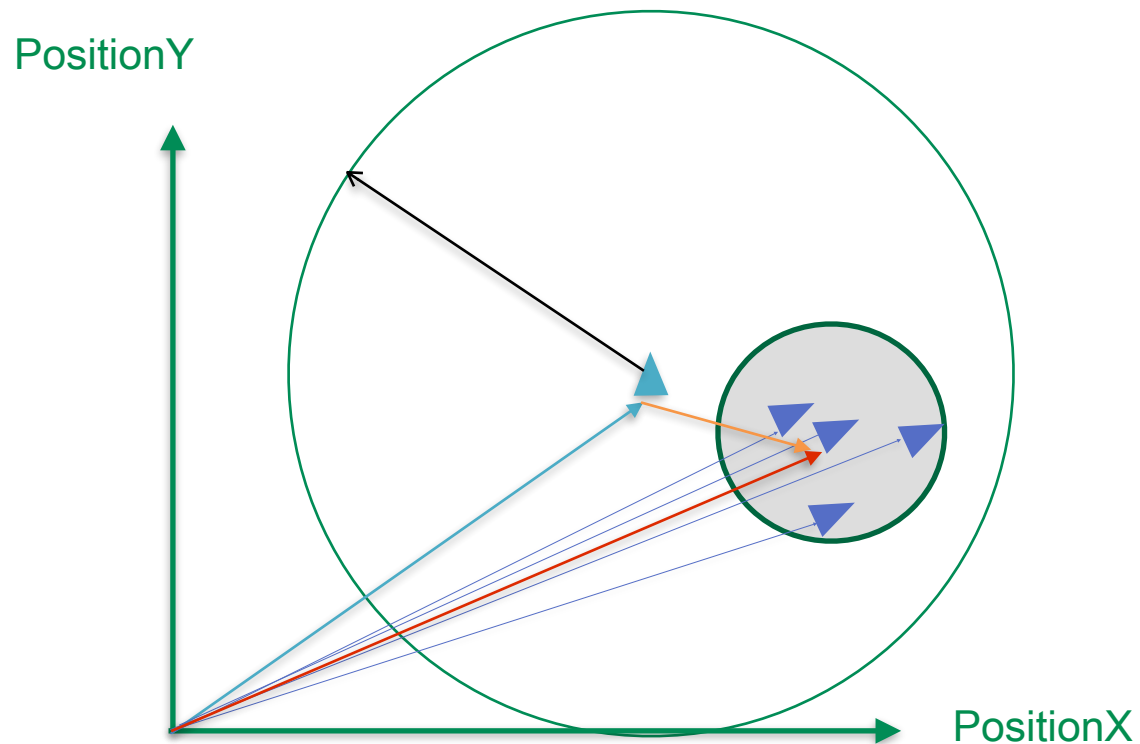
# Boids Model - Separation

# Boids Model - Separation





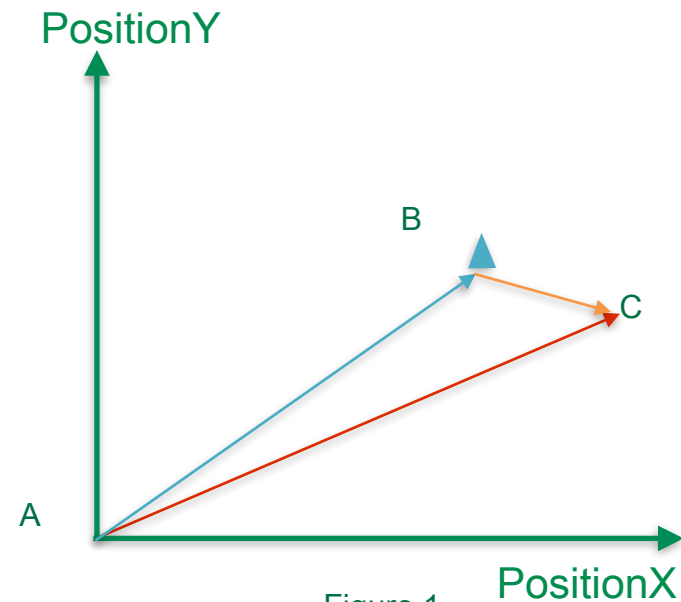
# Boids Model - Separation



- Average Heading of the flocks
- Current Position Vector
- Actual Heading of individual bonds
- Resultant Vector
- Flocking Radius

# Boids Model - Separation

- Checks for nearby boids and calculate the separation vector from each individual boid
- Normalise it
- Weight by distance (For smoothness)
- Loop and find average





# THE MODEL IMPLEMENTATION



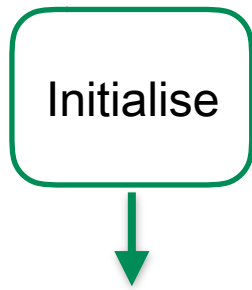
# Workflow

# Workflow

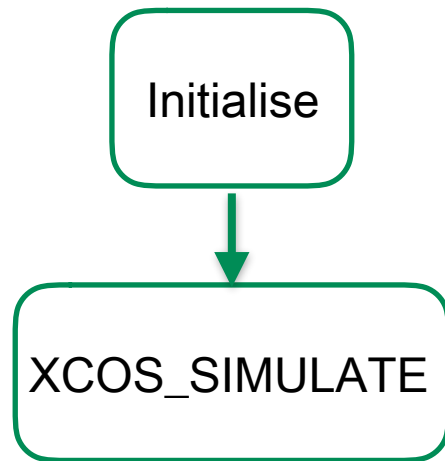


Initialise

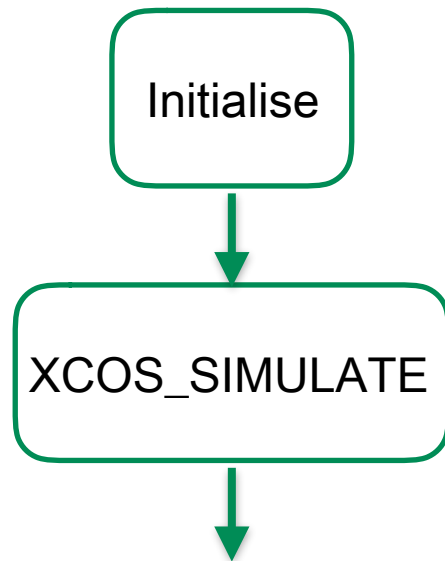
# Workflow



## Workflow

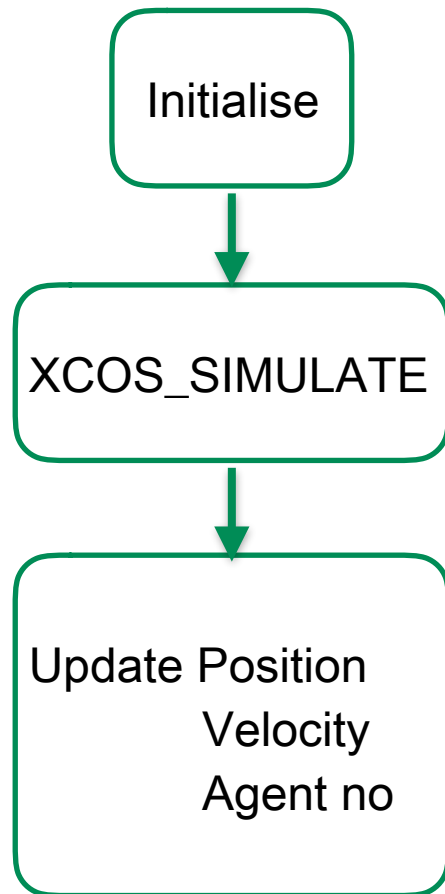


## Workflow

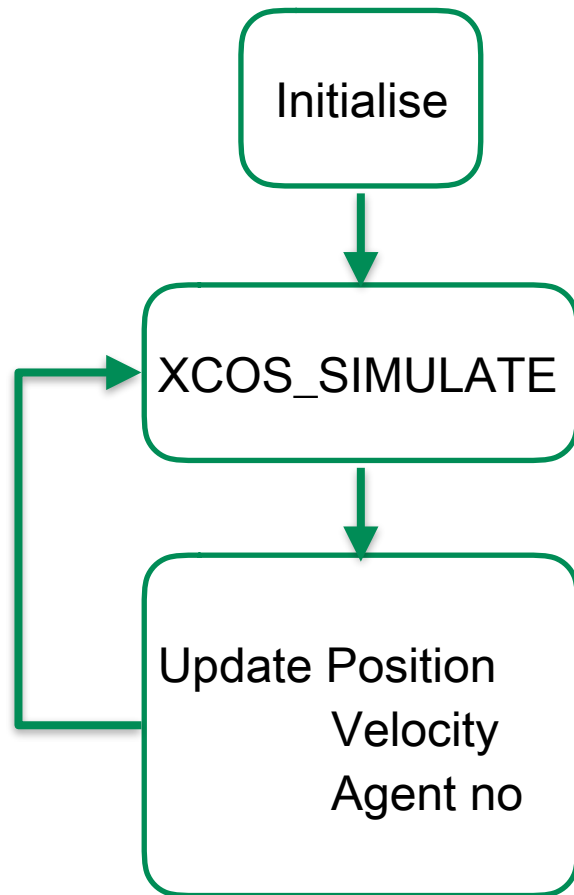




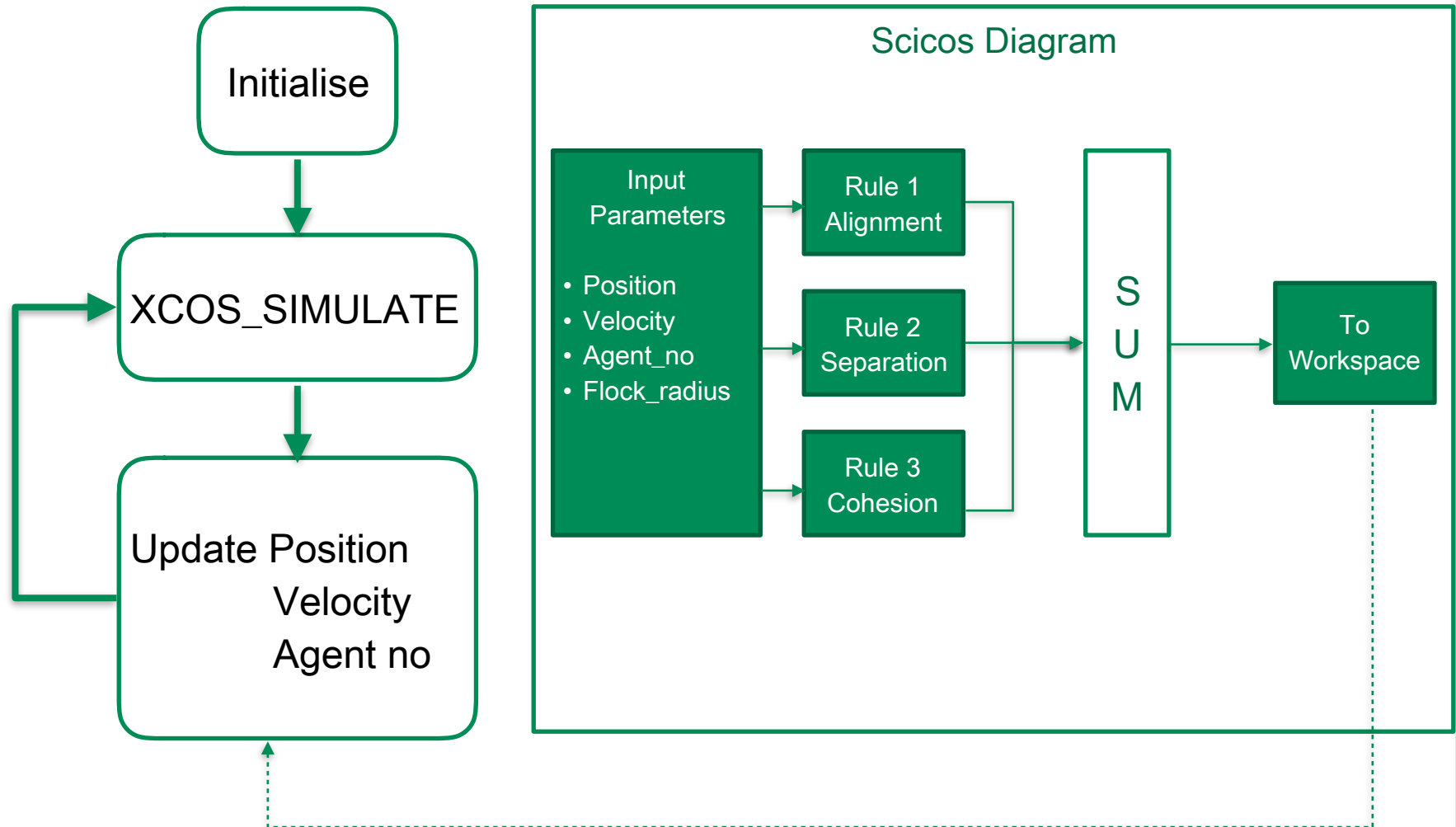
## Workflow



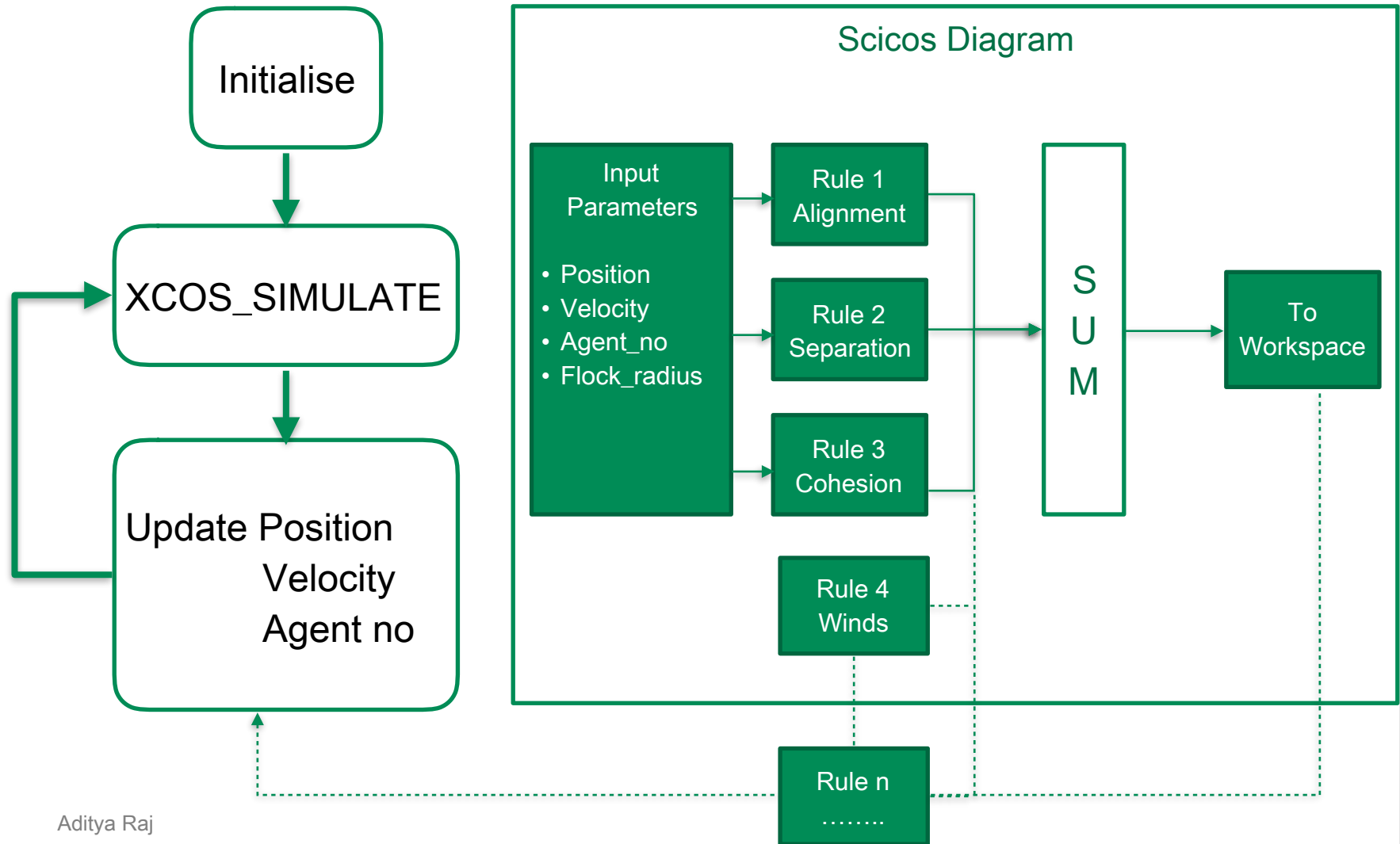
## Workflow



# Workflow

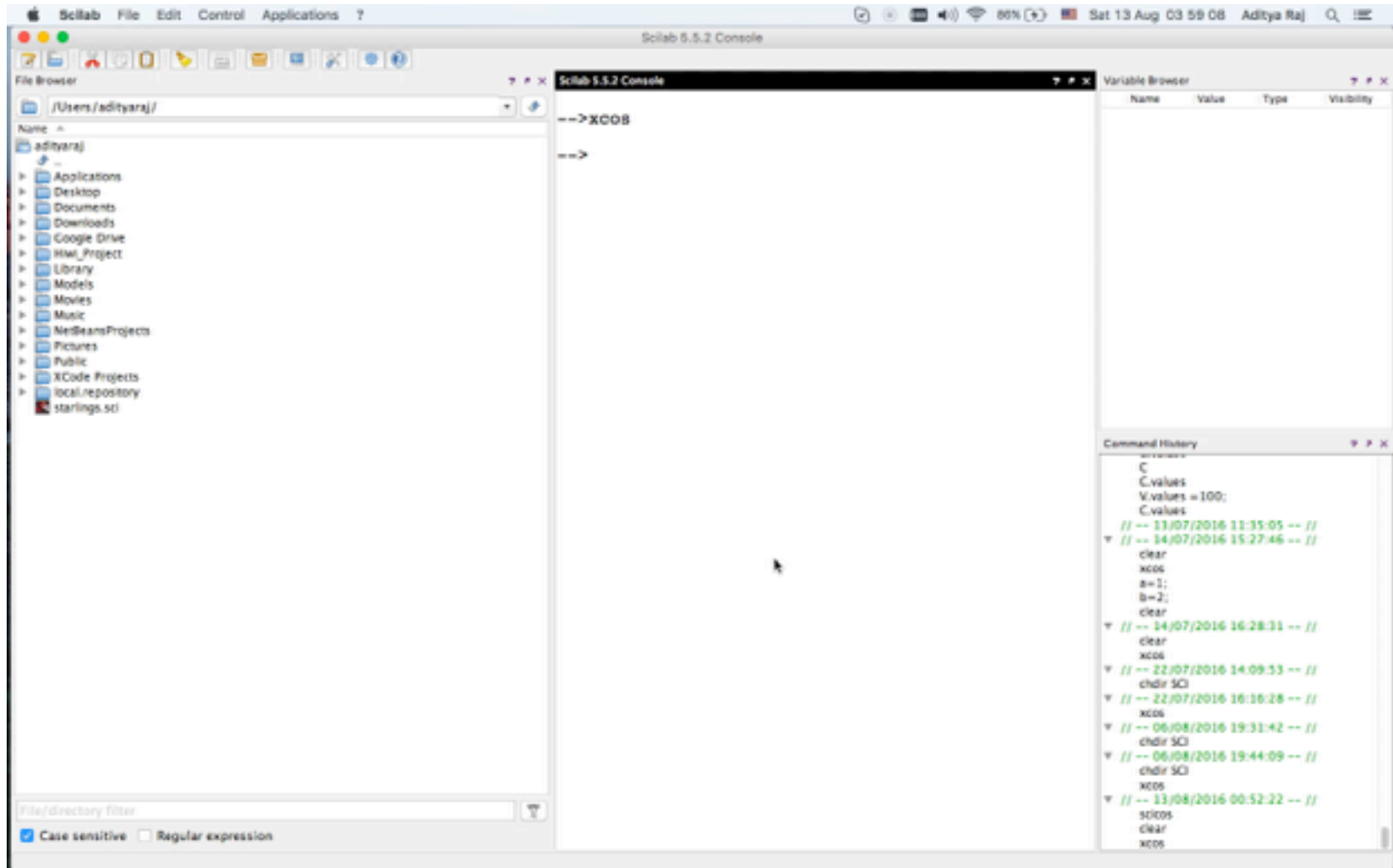


# Workflow



# SIMULATION RESULTS

## Results



The screenshot displays the Scilab 5.5.2 interface with three main panels:

- File Browser:** Shows the file system structure under `/Users/adityaraj/`. The `adityaraj` directory is expanded, showing subdirectories like `Applications`, `Desktop`, `Documents`, `Downloads`, `Google Drive`, `HWI_Project`, `Library`, `Models`, `Movies`, `Music`, `NetBeansProjects`, `Pictures`, `Public`, `XCode Projects`, `local.repository`, and `starlings.sci`.
- Scilab 5.5.2 Console:** Displays the command `-->XCOS` and the prompt `-->`.
- Variable Browser:** A table showing the state of variables in the workspace.

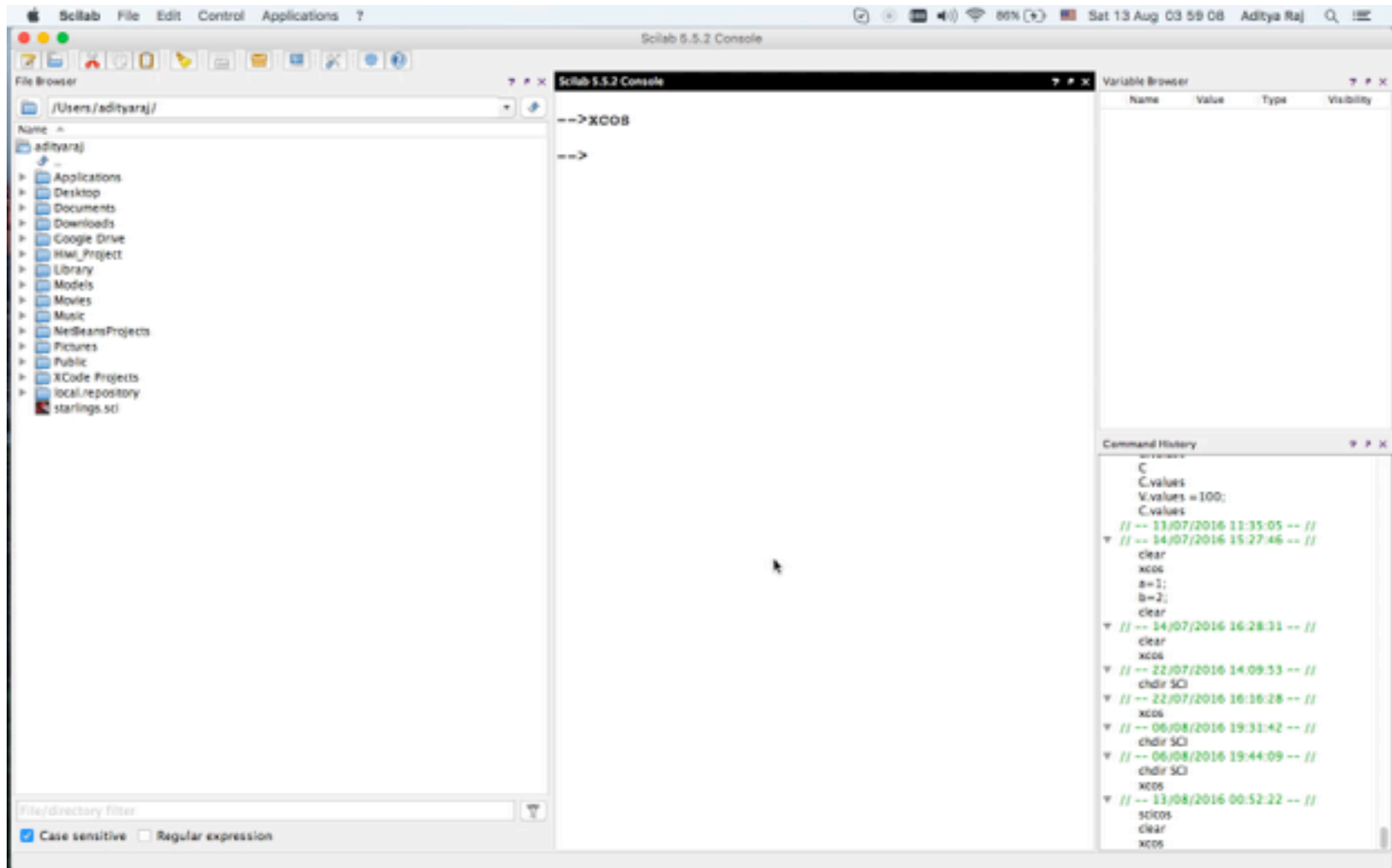
Name	Value	Type	Visibility
C	C.values		
V.values	100;		
C.values			
clear			
xcos			
a=1;			
b=2;			
clear			
clear			
xcos			
chdir SCI			
xcos			
chdir SCI			
xcos			
chdir SCI			
xcos			
chdir SCI			
clear			
xcos			

**Command History:**

```

C
C.values
V.values = 100;
C.values
// -- 13/07/2016 11:35:05 -- //
// -- 14/07/2016 15:27:46 -- //
clear
xcos
a=1;
b=2;
clear
// -- 14/07/2016 16:28:31 -- //
clear
xcos
// -- 22/07/2016 14:09:53 -- //
chdir SCI
// -- 22/07/2016 16:16:28 -- //
xcos
// -- 06/08/2016 19:31:42 -- //
chdir SCI
// -- 06/08/2016 19:44:09 -- //
chdir SCI
xcos
// -- 13/08/2016 00:52:22 -- //
scicos
clear
xcos
  
```

# Results



The screenshot displays the Scilab 5.5.2 interface. The top menu bar includes 'Scilab', 'File', 'Edit', 'Control', and 'Applications'. The status bar at the top right shows 'Sat 13 Aug 03 59 08' and 'Aditya Raj'.

The interface is divided into three main panels:

- File Browser:** Located on the left, it shows the file system structure under '/Users/adityaraj/'. The 'adityaraj' directory is expanded, listing various folders like 'Applications', 'Desktop', 'Documents', 'Downloads', 'Google Drive', 'Hwaj\_Project', 'Library', 'Models', 'Movies', 'Music', 'NetBeansProjects', 'Pictures', 'Public', 'XCode Projects', 'local.repository', and 'starlings.sci'.
- Scilab 5.5.2 Console:** The central panel, currently showing the command prompt '-->XCOS' and a subsequent '-->'.
- Variable Browser:** Located on the right, it displays a table with columns 'Name', 'Value', 'Type', and 'Visibility'. It is currently empty.

At the bottom right, a **Command History** panel lists previous commands and their execution times, including:

- C
- C.values
- V.values = 100;
- C.values
- // -- 13/07/2016 11:35:05 -- //
- // -- 14/07/2016 15:27:46 -- //
- clear
- xcos
- a=1;
- b=2;
- clear
- // -- 14/07/2016 16:28:31 -- //
- clear
- xcos
- // -- 22/07/2016 14:09:53 -- //
- chdir SCI
- // -- 22/07/2016 16:16:28 -- //
- xcos
- // -- 06/08/2016 19:31:42 -- //
- chdir SCI
- // -- 06/08/2016 19:44:09 -- //
- chdir SCI
- xcos
- // -- 13/08/2016 00:52:22 -- //
- scicos
- clear
- xcos

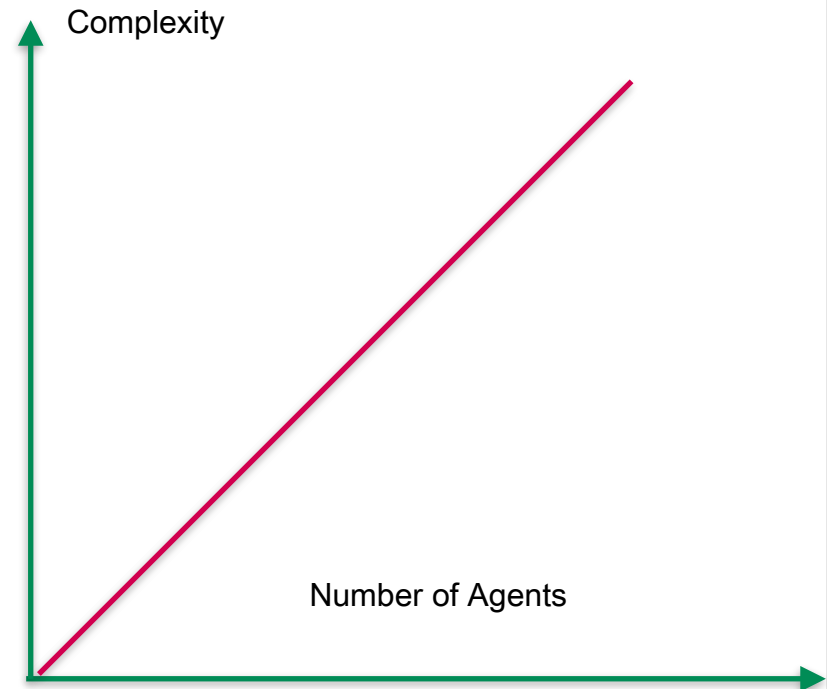
# EVALUATION

## COMPLEXITY



# Complexity

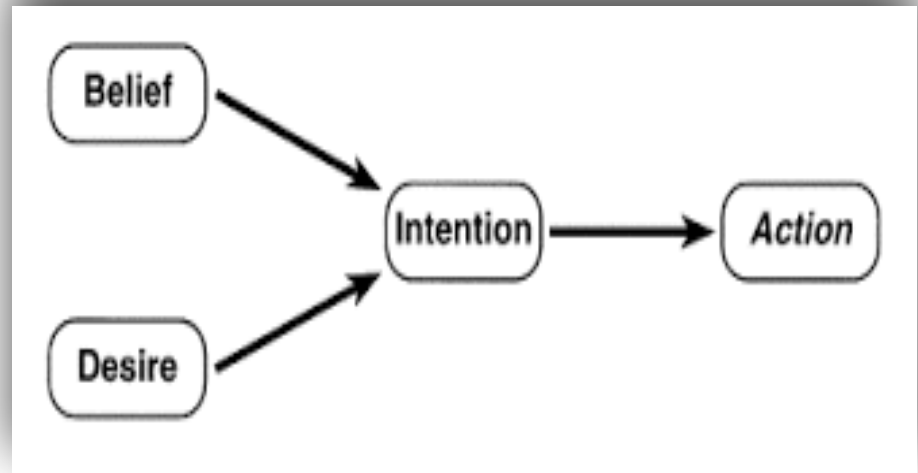
- Compilation of User-Defined Blocks.
- Let  $n$  = Number of Agents. Complexity for Alignment
  - Alignment:  $O(n)$  for finding it's neighbours and further adding it to the sum vector
  - Separation: Same as of alignment
  - Cohesion: Same as of alignment
- Workspace read and write delay.
- $O(n)$  for each cycle of plot.



## Future Improvements



**Stream API**



# References

1. Reynolds, Craig (1987). "Flocks, herds and schools: A distributed behavioral model.". SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. Association for Computing Machinery: 25–34. doi:10.1145/37401.37406. ISBN 0-89791-227-6.
2. Min, Hongkyu; Wang, Zhidong (2011). Design and analysis of Group Escape Behavior for distributed autonomous mobile robots. IEEE International Conference on Robotics and Automation (ICRA).
3. Saska, Martin; Jan, Vakula; Libor, Preucil (2014). Swarms of micro aerial vehicles stabilized under a visual relative localization. IEEE International Conference on Robotics and Automation (ICRA).
4. Moere, A V (2004). "Time-Varying Data Visualization Using Information Flocking Boids". Proceedings of the IEEE Symposium on Information Visualization. pp. 97–104. doi:10.1109/INFVIS.2004.65.
5. Cui, Zhihua; Shi, Zhongzhi (2009). "Boid particle swarm optimisation". International Journal of Innovative Computing and Applications. 2 (2): 77–85. doi:10.1504/IJICA.2009.031778.
6. Lebar Bajec, Iztok; Heppner, Frank H. (2009). "Organized flight in birds" (PDF). Animal Behaviour. pp. 777–789. doi:10.1016/j.anbehav.2009.07.007.
7. Weblink: <https://processing.org/examples/flocking.html>
8. Tutorial Creating C Function BLOCKS: <http://www.scicos.org/ScicosCBlockTutorial.pdf>

Any questions?

Aditya Raj  
e: [araj14@tu-clausthal.de](mailto:araj14@tu-clausthal.de)

