

Phase 1: Link Fundamentals

Eric Araújo, and Jonathan Hill

Keywords agent-based modeling, social sciences, computational methods

1. LEARNING OBJECTIVES

By the end of this phase, you will be able to:

- Create links between turtles using `create-links-with`.
- Access connected neighbors using `link-neighbors`.
- Reference link endpoints using `end1` and `end2`.
- Modify link properties dynamically.
- Understand the difference between undirected and directed links.

2. CONCEPTUAL FRAME

Before we model polarization, let's learn how to represent relationships in NetLogo. Links are how we show that two people are connected—they have a relationship. Think of your own social network: you have friends, and your friends have friends. Links represent those friendships.

Links are the innovation that makes this module different from previous modules. In earlier modules (segregation, contagion), agents interacted based on proximity—who was nearby. Now, agents interact through choice. People link to whom they want to connect with. This opens up entirely new possibilities for modeling social phenomena where relationships matter.

3. KEY CONCEPTS IN PHASE 1

3.a. Undirected vs. Directed Links

Undirected links represent symmetric relationships (friendship, collaboration, mutual following):

- If A is linked to B, B is automatically linked to A
- Use `undirected-link-breed` for these

Directed links represent asymmetric relationships (following, parent-child, authority):

- If A links to B, B is not automatically linked to A
- Use `directed-link-breed` for these

In the polarization model, we use **undirected links** because friendship and social influence are symmetric.

3.b. Link Properties

Just like turtles can have properties, links can too:

```
undirected-link-breed [social-bonds social-bond]
social-bonds-own [strength]
```

This creates a link type called `social-bonds` with a property called `strength`. This allows us to track characteristics of relationships, not just their existence.

3.c. Accessing Neighbors

Once links exist, accessing connected turtles is easy:

```
ask turtle 0 [
  ask link-neighbors [set color green] ; Highlight all neighbors
]
```

`link-neighbors` is incredibly useful—it returns an agentset of all turtles connected to the asking turtle.

3.d. Starter Code for Phase 1

```
breed [people person]
undirected-link-breed [social-bonds social-bond]

people-own [opinion]
social-bonds-own [strength]

to setup
  clear-all

  ; Create 20 people with random opinions
  create-people 20 [
    set opinion random 100
    setxy random-xcor random-ycor
    set size 1.5
    set color blue
  ]

  ; Each person creates links to 3 random others
  ask people [
    create-social-bonds-with n-of 3 other people [
      set strength random-float 1.0 ; Random strength 0-1
      set color grey
    ]
  ]

  reset-ticks
end

to go
  tick
end
```

After creating this code, add a Setup and Go button to run the model. When you click Setup, you should see 20 blue turtles connected by grey links.

4. CLASSROOM ACTIVITIES - PHASE 1

4.a. Activity 1.1: Explore the Network

Students run the setup and explore:

- “How many links were created?” (Should be 30: $20 \text{ people} \times 3 \text{ links} \div 2$ for undirected).
- Write: ask links [set color red] to turn all links red.
- Write: ask links [set thickness 0.5] to make links thicker.
- Write: ask links [set thickness thickness * 2] repeatedly to watch links grow.

Learning point: Links are agents too; they can be modified just like turtles.

4.b. Activity 1.2: Understanding link-neighbors

Add a command button called highlight-neighbors with this code:

```
to highlight-neighbors
  ask people [set color blue] ; Reset all to blue
  ask one-of people [
    ask link-neighbors [set color green]
  ]
end
```

Answer:

- “What does link-neighbors show us?”
- “Why is this useful?”
- “What if I click repeatedly?”

Learning point: link-neighbors gives us access to interaction partners.

4.c. Activity 1.3: Accessing Link Endpoints

Add this code to your Code tab and create a button show-link-info to call it:

```
to show-link-info
  let example-link one-of links
  ask example-link [
    show (word "Link between turtle " end1 " and turtle " end2)
    show (word "Strength: " strength)
  ]
end
```

Answer:

- “How do we refer to the two turtles at each end of a link?”
- “Why would we need to access endpoints?”
- “Can we ask the endpoints questions? How?”

Learning point: Links connect turtles, and we can reference those turtles through end1 and end2.

4.d. Activity 1.4: Creating Links Conditionally

Add this code to your Code tab and create a button create-links-by-color to call it:

```
to create-links-by-color
  ask people with [color = blue] [
    create-social-bonds-with other people with [color = blue] [
      set strength 0.8
      set color blue
    ]
  ]
end
```

Answer:

- “What does this do?”
- “What if I run it multiple times?”
- “How could we prevent duplicate links?”

Learning point: We can use conditions to control link formation.

4.e. Assessment - Phase 1

Students should be able to:

- Write code to count total links: count links
 - Write code to find the most-connected person: max-one-of people [count link-neighbors]
 - Write code to color links based on link strength: ask links [set color scale-color red strength 0 1]
 - Explain what link-neighbors returns and why it’s useful
-

5. FINAL CODE

The final code for Phase 1 [can be found here](#).
