

Paper folding is a wonderfully tactile way to help students visualize congruency criteria—especially in triangles—using real-world geometry. Here's how folding square and rectangular papers can justify different congruence criteria:

## 1. Square Paper Folding

### Example: Diagonal Fold (AC or BD)

- Fold a square along one of its diagonals.
- This creates two right-angled triangles.
- Since the diagonal is common, and the two sides adjacent to it are equal (sides of the square), the triangles satisfy the **Side-Side-Side (SSS)** criterion.
- Also, the included angles are equal, so **Side-Angle-Side (SAS)** is also justified.

### Example: Midpoint Fold (joining midpoints of opposite sides)

- Fold the square so that opposite sides align.
- This creates two congruent rectangles or triangles depending on the fold.
- You can justify **Angle-Side-Angle (ASA)** or **Angle-Angle-Side (AAS)** depending on the configuration.

## 2. Rectangular Paper Folding

### Example: Fold along the longer side's midpoint

- Fold the rectangle lengthwise so that the long edges align.
- This creates two congruent rectangles.
- If you draw diagonals before folding, you can form triangles and justify **SAS** or **SSS**.

### Example: Fold corner to corner (diagonal)

- This creates two right-angled triangles.
- The hypotenuse is the diagonal, and the legs are the rectangle's sides.
- Again, **SSS** and **SAS** can be demonstrated.

## Why It Works

Folding is a physical demonstration of **rigid motions**—translations, reflections, and rotations—which preserve distance and angle. When one triangle folds perfectly onto another, it proves congruence through these motions.

## **Worksheet: Exploring Triangle Congruency with Paper Folding**

**Grade Level:** Class 9 **Topic:** Congruency of Triangles (SSS, SAS, ASA, AAS) **Materials Needed:**

- Square and rectangular papers
- Pencil
- Ruler
- Scissors (optional)

### **Activity 1: Folding a Square Along the Diagonal**

**Steps:**

1. Take a square sheet and fold it along diagonal AC.
2. Open the fold and label the triangle halves as  $\triangle ABC$  and  $\triangle DCB$ .
3. Mark equal sides and angles.

**Questions:**

- Which sides are equal?
- Is the diagonal common in both triangles?
- What congruency criteria is satisfied? > *Answer: SSS and SAS*

### **Activity 2: Folding a Rectangle Corner-to-Corner**

**Steps:**

1. Take a rectangular sheet and fold from one corner to the opposite (forming a diagonal).
2. Label the resulting triangles.

**Questions:**

- Are the legs of both triangles equal?
- Is the diagonal shared?
- What congruence rule applies? > *Answer: SSS and SAS*

### **Activity 3: Folding Square to Align Opposite Sides**

**Steps:**

1. Fold the square to bring opposite sides together.
2. Observe the two new shapes formed.

#### Questions:

- Are the resulting shapes congruent?
- Is any angle or side preserved?
- Which criteria (ASA or AAS) can be inferred? > *Answer: ASA or AAS*

#### Activity 4: Creative Fold & Predict

#### Challenge:

- Ask students to make any fold of their choice on a square or rectangle.
- Predict which congruence rule might be demonstrated.
- Justify their answers using geometric reasoning.

Expanding the lesson beyond triangle congruence can really deepen your students' geometric intuition. Here are some **related geometric concepts** that naturally connect with congruency and can be explored through folding or visual reasoning:

#### ◆ Symmetry

- **Line symmetry:** Folding shapes to see if both halves match.
- **Rotational symmetry:** Rotating folded shapes to check for congruence.
- Perfect for linking with congruent halves in paper folding.

#### 📐 Rigid Transformations

- **Translation, rotation, and reflection** preserve shape and size.
- These are the backbone of congruency—folding is essentially a reflection!

#### ▲ Properties of Triangles

- **Angle sum property:** Useful when verifying ASA or AAS.
- **Isosceles and equilateral triangles:** Naturally lead to congruent parts.

#### ═ Similarity vs. Congruence

- Introduce **similar figures** (same shape, different size).
- Helps contrast with congruent figures and explore scale factors.

### Corresponding Parts of Congruent Triangles (CPCTC)

- Once triangles are proven congruent, students can deduce equal parts.
- Great for logical reasoning and proof-based questions.

### Coordinate Geometry

- Use coordinates to prove congruence algebraically (distance formula, slope).
- Adds a numerical layer to visual understanding.

### Circle Geometry

- Congruent chords, arcs, and central angles.
- Folding a circle through its diameter can show symmetry and congruence.

Go with details for Coordinate Geometry Use coordinates to prove congruence algebraically (distance formula, slope). Adds a numerical layer to visual understanding.

You're right on target, Jagdev—coordinate geometry brings a whole new dimension (literally!) to congruence. Here's how you can teach or explore triangle congruence algebraically using coordinates, **distance formula**, and **slope**, giving students both visual and numerical perspectives.

#### Using Coordinate Geometry to Prove Congruence

##### 1. Distance Formula

To prove **SSS (Side-Side-Side)** or **SAS (Side-Angle-Side)** congruence:

- Use the formula:

**Distance between A(x<sub>1</sub>,y<sub>1</sub>) and B(x<sub>2</sub>,y<sub>2</sub>):d=(x<sub>2</sub>-x<sub>1</sub>)<sup>2</sup>+(y<sub>2</sub>-y<sub>1</sub>)<sup>2</sup>** text{Distance between } A(x\_1, y\_1) \text{ and } B(x\_2, y\_2): \quad d = \sqrt{(x\_2 - x\_1)^2 + (y\_2 - y\_1)^2}

$$\text{Distance between } A(x_1, y_1) \text{ and } B(x_2, y_2): \quad d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- By calculating side lengths, if all three are equal in two triangles, you've got **SSS**.
- If two sides and the included angle (verified using slopes) are equal, then **SAS** holds.

**Example:** Points A(1, 2), B(4, 6), C(7, 2) form triangle ABC Points P(2, -1), Q(5, 3), R(8, -1) form triangle PQR Find distances AB, BC, AC and compare with PQ, QR, PR to check for SSS.

## 2. Slope Formula for Angle Verification

To show **SAS** or **ASA** (Angle-Side-Angle):

- Use:

**Slope of a line through A(x<sub>1</sub>,y<sub>1</sub>),B(x<sub>2</sub>,y<sub>2</sub>): $m=y_2-y_1/x_2-x_1$**  \text{Slope of a line through } A(x\_1, y\_1), B(x\_2, y\_2): \quad m = \frac{y\_2 - y\_1}{x\_2 - x\_1}

$$\text{Slope of a line through } A(x_1, y_1), B(x_2, y_2): \quad m = \frac{y_2 - y_1}{x_2 - x_1}$$

- If two lines have the **same slope**, they're parallel (angle between them is preserved).
- If two lines have **negative reciprocal slopes**, they're perpendicular (angle = 90°).

**Example:** To prove right angle at vertex B:

- Find slopes of AB and BC.
- If slope of AB × slope of BC = -1, then ∠B = 90°, aiding right triangle congruence proofs.

## 3. Midpoint Formula (Optional Add-On)

Useful in proving **reflected or bisected sides** are equal:

**Midpoint of A(x<sub>1</sub>,y<sub>1</sub>),**

**B(x<sub>2</sub>,y<sub>2</sub>): $M=(x_1+x_2, y_1+y_2)/2$**

**A(x<sub>1</sub>, y<sub>1</sub>), B(x<sub>2</sub>, y<sub>2</sub>): $M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$**

**A(x<sub>1</sub>, y<sub>1</sub>), B(x<sub>2</sub>, y<sub>2</sub>): $M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$**

$$\text{Midpoint of } A(x_1, y_1), B(x_2, y_2): \quad M = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

 **Suggested Activity (Paper + Plotting):****"Plot & Prove" Task:**

1. Provide students with coordinates of two triangles.
2. Ask them to:
  - o Plot both triangles on graph paper.
  - o Calculate and compare side lengths.
  - o Use slopes to verify angle congruence.
3. Conclude which congruence criterion is satisfied.

Your interactive worksheet on **Coordinate Geometry for Triangle Congruence** is being prepared! Here's the detailed activity set, complete with examples, formulas, and tasks for students to explore congruence algebraically.

**Worksheet: Proving Triangle Congruence Using Coordinate Geometry**

**Grade Level:** Class 9 **Topic:** Coordinate Geometry and Triangle Congruence **Objective:** Use distance formula, slope formula, and midpoint formula to prove triangle congruence algebraically.

**Key Formulas****1. Distance Formula:**

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Used to calculate the length of sides.

**2. Slope Formula:**

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Used to verify angles (parallel or perpendicular lines).

### 3. Midpoint Formula:

$$M = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

$$M = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Used to check bisected or reflected sides.

#### Activity 1: Prove SSS Congruence

Task:

1. Plot the following triangles on graph paper:
  - Triangle ABC with vertices A(1, 2), B(4, 6), C(7, 2).
  - Triangle PQR with vertices P(2, -1), Q(5, 3), R(8, -1).
2. Calculate the lengths of sides using the distance formula:
  - AB, BC, AC for Triangle ABC.
  - PQ, QR, PR for Triangle PQR.
3. Compare the side lengths of both triangles.

Questions:

- Are all three sides equal?
- Does this satisfy the SSS criterion?

#### Activity 2: Prove SAS Congruence

Task:

1. Use the same triangles from Activity 1.
2. Calculate the slopes of AB and BC for Triangle ABC, and PQ and QR for Triangle PQR using the slope formula.
3. Verify if the included angle between the two sides is preserved.

Questions:

- Are two sides equal and the included angle congruent?
- Does this satisfy the SAS criterion?

### **Activity 3: Prove ASA Congruence**

**Task:**

1. Plot Triangle DEF with vertices D(0, 0), E(4, 0), F(2, 3).
2. Plot Triangle XYZ with vertices X(1, 1), Y(5, 1), Z(3, 4).
3. Calculate the slopes of DE and EF for Triangle DEF, and XY and YZ for Triangle XYZ.
4. Verify if two angles and the included side are congruent.

**Questions:**

- Are two angles equal and the included side congruent?
- Does this satisfy the ASA criterion?

### **Activity 4: Challenge Task - Midpoint and Reflection**

**Task:**

1. Plot Triangle GHI with vertices G(2, 2), H(6, 2), I(4, 5).
2. Fold the graph paper along the line  $x = 4$ .
3. Reflect Triangle GHI to create Triangle G'H'I'.
4. Calculate the midpoints of GH, HI, and GI using the midpoint formula.

**Questions:**

- Are the midpoints of corresponding sides equal?
- Does reflection preserve congruence?

**Extension Questions:**

1. Can you prove congruence using only slopes and angles?
2. What happens if one triangle is scaled up or down? Does congruence still hold?

Would you like me to create geometric diagrams for these activities or format this into a PDF for classroom use? Let me know, and I'll finalize it for you! 

Here we go—your geometric diagrams for the coordinate geometry activities are on the way! These visuals will make it easier for students to grasp how triangle congruence is established numerically and spatially on a coordinate plane.

Once you've had a look, I can help integrate them into a printable worksheet or even animate the fold-reflection activity if you're going digital. Let's keep building! 📐✍️✨

<https://copilot.microsoft.com/shares/HABJaMY7aDYsU1ktmjTfp>

## Computer Programming work

Based on these

Absolutely, Jagdev! You can create **step-by-step geometric diagrams** using Python—particularly with libraries like **Matplotlib**, **NumPy**, and **Sympy**. These are excellent for plotting points, lines, and shapes on a coordinate grid. Here's an overview of how you might approach it:

### 1. Plotting Triangles and Coordinates (Matplotlib)

python

```
import matplotlib.pyplot as plt
```

```
# Coordinates for triangle ABC
```

```
A, B, C = (1, 2), (4, 6), (7, 2)
```

```
x = [A[0], B[0], C[0], A[0]]
```

```
y = [A[1], B[1], C[1], A[1]]
```

```
plt.plot(x, y, 'bo-') # 'bo-' means blue circles with lines
```

```
plt.text(*A, 'A', fontsize=12)
```

```
plt.text(*B, 'B', fontsize=12)
```

```
plt.text(*C, 'C', fontsize=12)
```

```
plt.grid(True)
```

```
plt.axis("equal")
```

```
plt.title("Triangle ABC on Coordinate Plane")
```

```
plt.xlabel("X")
```

```
plt.ylabel("Y")
```

```
plt.show()
```

## 2. Add Sides, Slopes, or Reflected Triangles

Use functions to:

- Calculate distances
- Annotate slopes on sides
- Reflect triangles by multiplying coordinates (e.g.,  $(-x,y)(-x, y)$ )

## 3. Automating Step-by-Step Visuals

Create a loop or function that saves each diagram as an image:

```
python
```

```
for step in steps:
```

```
    plot_diagram(step)  
    plt.savefig(f'step_{step_number}.png')
```

You can combine these images into an animation using imageio, or compile them in a GUI (like with Tkinter) or a web app (using Streamlit) to simulate step-by-step folding or plotting.

If you'd like, I can build a ready-to-run Python script that renders each activity from your worksheet as step-by-step coordinate visuals. Would you prefer code for one triangle congruence proof first—like SSS or ASA—to test it out? 

CODE:

### 1. app.py

- `from flask import Flask, render_template, send_from_directory`
- `import matplotlib.pyplot as plt`
- `import os`
- 
- `# Clean previous step images`
- `for file in os.listdir('static'):`
- `if file.startswith("step_") and file.endswith(".png"):`
- `os.remove(os.path.join('static', file))`
- 
- `app = Flask(__name__)`

```
•
•     @app.route('/')
•     def index():
•         steps = sorted(os.listdir('static'))
•         return render_template('index.html', steps=steps)
•
•
•     @app.route('/static/<filename>')
•     def static_files(filename):
•         return send_from_directory('static', filename)
•
•
•     # Define plotting function
•     def plot_diagram(step, step_number):
•         points = step["points"]
•         plt.figure(figsize=(6, 6))
•
•
•         # Draw connecting sides
•         for (start, end) in step["connect"]:
•             x_vals = [points[start][0], points[end][0]]
•             y_vals = [points[start][1], points[end][1]]
•             plt.plot(x_vals, y_vals, 'bo-')
•
•
•         # Label points
•         for label in step["highlight"]:
•             x, y = points[label]
•             plt.text(x + 0.1, y + 0.1, label, fontsize=12,
•                      color='darkred')
•
•
•         # Format plot
•         plt.title(step["title"])
•         plt.grid(True)
•         plt.axis('equal')
•         plt.xlabel("X-axis")
•         plt.ylabel("Y-axis")
•         plt.savefig(f'static/step_{step_number}.png')
•         plt.close()
•
•
•     # Define sequence of steps
•     steps = [
•         {
```

```

•     "title": "Step 1: Plot Triangle ABC",
•     "points": {"A": (1, 2), "B": (4, 6), "C": (7, 2)},
•     "highlight": ["A", "B", "C"],
•     "connect": [("A", "B"), ("B", "C"), ("C", "A")]
•   },
•   {
•     "title": "Step 2: Plot Triangle PQR",
•     "points": {"P": (2, -1), "Q": (5, 3), "R": (8, -1)},
•     "highlight": ["P", "Q", "R"],
•     "connect": [("P", "Q"), ("Q", "R"), ("R", "P")]
•   },
•   {
•     "title": "Step 3: Compare Side Lengths for SSS",
•     "points": {
•       "A": (1, 2), "B": (4, 6), "C": (7, 2),
•       "P": (2, -1), "Q": (5, 3), "R": (8, -1)
•     },
•     "highlight": ["A", "B", "C", "P", "Q", "R"],
•     "connect": [
•       ("A", "B"), ("B", "C"), ("C", "A"),
•       ("P", "Q"), ("Q", "R"), ("R", "P")
•     ]
•   }
• ]
•
• # Generate and save all diagrams
• for idx, step in enumerate(steps, start=1):
•     plot_diagram(step, idx)

```

### 3. index.html

```

• <html>
• <!DOCTYPE html>
• <html>
•   <head>
•     <title>Triangle Congruence Steps</title>
•   </head>
•   <body>
•     <h1>Visualizing Triangle Congruence</h1>

```

- `{% for step in steps %}`
- `<div>`
- ``
- `<h3 align=center>{{ step }}</h3>`
- `</div>`
- `{% endfor %}`
- `</body>`
- `</html>`
- 

1.