One rule. Endless detail.

Mandelbrot & Julia sets: eight proofs, carefully explained

Based on Raghav Sharma’s original mathematical notes and Java implementation. This study guide adds clear notation, worked examples, visual explanations and practical experiments.

This guide follows the eight mathematical notes and the recurrence calculated by the explorer. Each proof is explained in plain language, with examples you can check yourself.

How to use this book

Read the definitions first. Work through each numbered argument, check its assumptions, then try the experiment in Learn or Inspect. Proofs 1.1-1.5 explain the recurrence; 2.1-2.3 connect the plane to the interface.

Important correction

Proof 1.3 checks a tempting but false inequality: a small modulus does not force growth. For example, c = 0 and z = 0 is a fixed orbit. The valid radius-2 escape argument is explained step by step.

Edition: 20 October 2026. A finite image is an approximation, not a proof of membership.

START HERE

The language of the picture

Complex numbers

Write $z = a + ib$ with $i^2 = -1$. The real coordinate a runs horizontally; b is the imaginary coordinate. The modulus $|z| = \sqrt{a^2 + b^2}$ is distance from the origin.

An orbit

Choose c and a starting point z_0, then repeatedly calculate $z_{n+1} = z_n^2 + c$. The resulting sequence is the orbit. Bounded means one finite radius contains every term, however many steps are taken.

Two different planes

Mandelbrot fixes $z_0 = 0$ and varies c from pixel to pixel. A filled Julia set fixes c and varies z_0. Its boundary is the Julia set. Selecting a Mandelbrot coordinate therefore selects an entire Julia experiment.

A worked orbit

For $c = 1$ and $z_0 = 0$, the first values are 0, 1, 2, 5, 26. For $c = 0$, the Mandelbrot orbit stays at 0. For $c = -2$, it goes 0, -2, 2, 2, ... Equality at radius 2 does not prove escape.

Pixels, steps and color

Resolution controls how many locations are sampled. Iterations control how long each location is tested. Color shows the original Java escape-count mapping; changing hue changes presentation, not membership.

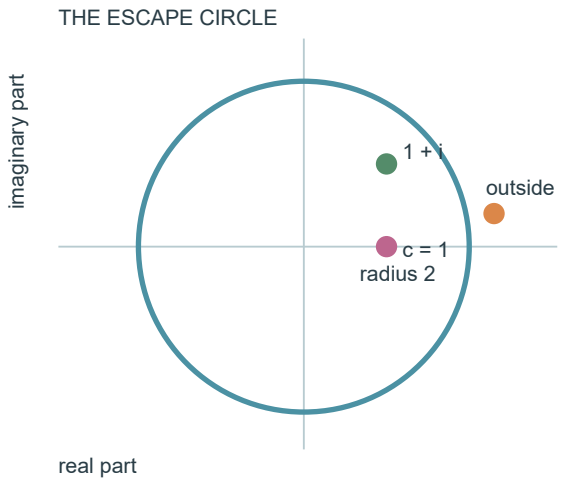
A sensible workflow

Navigate Mandelbrot at 200 iterations. Julia starts at 1,000. Settle the view, then use Capture for up to 100,000 steps and a larger PNG. Work is roughly pixel count multiplied by allowed steps, although many points escape earlier.

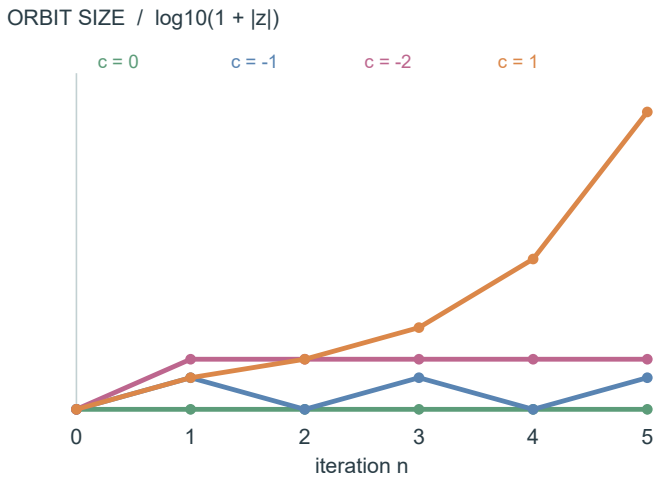
SEE THE MATHEMATICS

Two pictures of one recurrence

A radius-2 circle gives a quick escape test for Mandelbrot parameters. Crossing it proves escape; staying inside does not prove membership.



The chart compares four short orbits. The vertical axis uses log10(1 + |z|) so rapid growth and bounded examples remain visible together.



Read the lines: c = 0 stays fixed; c = -1 alternates between 0 and -1; c = -2 reaches the boundary value 2; c = 1 grows beyond it and escapes.

PROOF 1.1

Why the Mandelbrot set lies within radius 2

Statement and notation

$z_0 = 0$; $z_1 = c$; M □ {c : |c| ≤ 2}

1. The first iterate equals the parameter: $z_1 = c$. Let C = |c|. If C > 2, the orbit already starts beyond radius 2.

2. For a later radius r ≥ C, the reverse triangle inequality gives $|z^2 + c| \geq r^2 - C \geq r^2 - r = r(r - 1)$. Since r > 2, this is larger than r by a factor greater than 1.

3. The radii keep growing without bound. Therefore |c| > 2 excludes a Mandelbrot parameter. Being inside radius 2 is only a necessary condition: c = 1 is still outside the set.

This parameter shortcut applies to Mandelbrot. Applying it to every Julia constant is not a general Julia membership test; Inspect explains when this renderer shortcut applies.

Work it through

Trace c = 1 and c = 0 in Inspect. Both lie within radius 2, but their orbits behave differently.

For C > 2, put q = C - 1 > 1. Since every later r stays at least C, the estimate r(next) >= q r gives r(n) >= C q^(n-1). This geometric lower bound proves unbounded growth, rather than only showing that consecutive radii increase.

Source: original note 1.1 in Mandel Brot and Julia set Equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 1.2

A distance test without square roots

Statement and notation

$|c|^2 = c_a^2 + c_b^2$; $|c| > 2 \Leftrightarrow c_a^2 + c_b^2 > 4$

1. A complex number $c = c_a + ic_b$ is the point (c_a, c_b). Pythagoras gives its distance from the origin.

2. Both the modulus and radius 2 are nonnegative, so squaring preserves the inequality. Compare the sum of squares with 4 instead of evaluating a square root.

3. This is the same geometric circle expressed in arithmetic that the Java renderer uses for its initial parameter check.

Work it through

For c = 1 + i, the squared modulus is 2. For c = 2 + i, it is 5, beyond the radius-2 circle.

Check both sides numerically: c = 1 + i has |c| = sqrt(2) and squared modulus 2, so neither test crosses the threshold. For 2 + i, sqrt(5) > 2 and 5 > 4 give the same answer. Squaring is safe here because both quantities are nonnegative.

Source: original note 1.2 in Mandel Brot and Julia set Equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 1.3

Why crossing radius 2 proves escape

Statement and notation

$|z_{n+1}| \geq |z_n|^2 - |c| \geq r^2 - 2 > r$, when $|c| \leq 2$ and $r = |z_n| > 2$

1. The reverse triangle inequality bounds $|z^2 + c|$ from below by $|z|^2 - |c|$. For candidate Mandelbrot parameters, $|c| \leq 2$, so $r^2 - 2$ is a valid lower bound.

2. The growth condition $r^2 - 2 > r$ factors as $(r - 2)(r + 1) > 0$. Since a modulus cannot be negative, $r > 2$ is the relevant region.

3. Once r exceeds 2, every later radius remains above it. The positive growth gap cannot shrink below its first value, so the orbit is unbounded.

Correction to the original PDF, branch 3b: $|c| \leq 2$ does not imply $|c| - r^2 \geq 2 - r^2$. The claim that every orbit with $r < 1$ must grow is false: $c = 0$, $z = 0$ stays at 0. The $r > 2$ escape argument above remains valid. The source PDF is kept unchanged.

Work it through

Trace $c = 1$: 0, 1, 2, 5. Equality at radius 2 does not trigger escape; the next term does.

Strictly increasing radii alone would not prove divergence. Here the gap $g(r) = r^2 - r - 2$ increases for $r > 2$. Starting at $r0 > 2$, each later step grows by at least $g(r0) > 0$, so $r(k) >= r0 + k\, g(r0)$, which tends to infinity.

Source: original note 1.3 in Mandel Brot and Julia set Equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 1.4

The loop’s squared escape condition

Statement and notation

Continue while $z_a^2 + z_b^2 \le 4$ and $n < \text{iterationLimit}$

1. Apply the distance formula to the current orbit value z_n, rather than the parameter c. Its squared modulus is $z_a^2 + z_b^2$.

2. A strict crossing above 4 ends the original loop. Equality stays in the loop: replacing ≤ with < would change the behavior at radius 2.

3. The iteration limit is the other stopping condition. Reaching it means not escaped within this finite test, not proven bounded forever.

The radius-2 justification here assumes $|c| \le 2$. For a general Julia constant, a sufficient escape radius is $\max(2, |c|)$; this guide describes the original implementation without changing it.

Work it through

For c = -2, the Mandelbrot orbit reaches -2, then 2, then remains at 2. It must not be classified as escaped solely at equality.

If $|c| > 2$ in a general Julia problem, choose $R = \max(2, |c|)$. For $r > R$, $r^2 - |c| >= r^2 - r = r(r-1) > r$. The original renderer instead applies its parameter shortcut first, so it is not a complete general-purpose Julia implementation for such constants.

Source: original note 1.4 in Mandel Brot and Julia set Equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 1.5

Split the iteration into real and imaginary parts

Statement and notation

$a' = a^2 - b^2 + c_a$; $b' = 2ab + c_b$

1. Expand $(a + ib)^2 = a^2 + 2iab + i^2b^2$. Since $i^2 = -1$, its real part is $a^2 - b^2$ and imaginary part is $2ab$.

2. Adding $c_a + ic_b$ produces the two update equations. Both use the old a and b; store the new values in temporaries before replacing them.

3. Mandelbrot starts at $z_0 = 0$ and varies c per pixel. Julia fixes c and varies the initial z_0 per pixel. The same recurrence supports both views.

Work it through

With $z = 1 + i$ and $c = 0$, the next value is $2i$. Use the real and imaginary equations to check it.

For $z = 1 + i$ and $c = 0$, $a(new) = 1 - 1 = 0$ and $b(new) = 2$. Updating a before calculating b would accidentally use the new zero and give the wrong result. Temporaries preserve the simultaneous recurrence.

Source: original note 1.5 in Mandel Brot and Julia set Equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 2.1

Map a pixel to a complex coordinate

Statement and notation

$$x = x_{min} + p_x/W \cdot (x_{max} - x_{min}); \quad y = y_{min} + (H - 1 - p_y)/H \cdot (y_{max} - y_{min})$$

1. Normalize a pixel index by image width or height, scale it by the corresponding complex-plane span, then add the minimum bound.

2. The original Java image writes rows in reverse vertical order. A displayed row py therefore maps to source row H - 1 - py. The formula above includes that flip.

3. Actual pixel indices run from 0 to W - 1 and 0 to H - 1. The final sampled coordinate is just short of the maximum bound. Inspect uses this pixel convention, rather than a rounded screen percentage.

Work it through

Select the top and bottom rows. The top corresponds to the larger imaginary value, even though screen y grows downward.

For W = 4, H = 3 and bounds [-2,2] by [-1.5,1.5], the displayed pixel (0,0) samples (-2,0.5), while (3,2) samples (1,-1.5). This asymmetry comes from the original endpoint-excluding samples and row reversal; it is not a continuous corner-coordinate convention.

Source: original note 2.1 in UIUX equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 2.2

Zoom while keeping the pointer anchored

Statement and notation

newSpan = oldSpan / zoomFactor; newMin = anchor - fraction · newSpan

1. Find the pointer’s fraction across the image. Translate that fraction into its complex-plane anchor coordinate.

2. Divide the span by a factor above 1 to zoom in; multiply by that factor to zoom out. Set the new minimum so the anchor stays at the same image fraction.

3. Apply this independently to x and y, accounting for screen y pointing downward. Both spans scale equally, preserving the shape.

Work it through

Point near a boundary and double-left-click repeatedly. The same feature stays under the pointer. Double-right-click reverses the zoom.

For x bounds [-2,2], pointer fraction f = 0.75 and zoom factor 2, the anchor is 1. The new span is 2, and newMin = 1 - 0.75*2 = -0.5. The new maximum is 1.5. The anchor remains three quarters of the way across.

Source: original note 2.2 in UIUX equations. Open the matching numbered proof PDF from the Learn & proofs page.

PROOF 2.3

Turn a drag into a viewport translation

Statement and notation

$$\Delta x = dx/W \cdot xSpan;\ \Delta y = dy/H \cdot ySpan;\ xmin' = xmin - \Delta x;\ ymin' = ymin + \Delta y$$

1. Here dx and dy mean release position minus drag-start position. Divide the pixel displacement by the displayed image dimensions.

2. Scale by the mathematical spans. Dragging the picture right shifts the viewed x bounds left, so subtract Δx from both x bounds.

3. Dragging down moves the mathematical bounds upward because screen and complex y directions are opposite. Add Δy to both y bounds. Spans remain unchanged, so panning does not zoom.

Work it through

Drag right, then back by the same amount. Compare the zoom label: it remains constant. During drag, the existing PNG moves; Java renders the translated view on release.

With a 1,000-pixel-wide image and x span 4, a drag 100 pixels to the right gives deltaX = 0.4. Subtract 0.4 from both x bounds. Every feature shifts right on screen, while the span and zoom stay unchanged.

Source: original note 2.3 in UIUX equations. Open the matching numbered proof PDF from the Learn & proofs page.

FIELD GUIDE

From wonder to an experiment

The “God theory”: what the nickname means

“The thumbprint of God” is a poetic nickname for the Mandelbrot set, shared with a wide audience by Arthur C. Clarke’s documentary The Colours of Infinity. The phrase compares its intricate, fingerprint-like boundary with a pattern that seems to grow endlessly from one short rule.

It is not a mathematical theory or evidence for a religious claim. Mathematics describes which complex parameters have bounded orbits. The nickname describes the awe some viewers feel while exploring that detail.

Zoom near a boundary. Spirals, branches and miniature shapes appear at many scales. A computer only samples finitely many points at finite numerical precision, so it cannot reveal infinite detail in one image.

Source: The Colours of Infinity; documentary catalog reference, American University Library: https://subjectguides.library.american.edu/content.php?pid=227712

Your first investigation

1. Trace c = 0, 1 and -2. Predict each orbit before pressing Trace.

2. Enter a Mandelbrot coordinate as c. Open its Julia set and notice that the view centers on the chosen coordinate.

3. Compare 200 and 1,000 iterations at identical resolution. Which pixels change?

4. Change resolution without changing iterations. Which differences are spatial sampling?

5. Capture a settled view. Record bounds, c, resolution and iterations alongside the image.

Controls and interpretation

Drag to pan; scroll or double-left-click to zoom in; double-right-click to zoom out. Plus/minus and arrow keys work when the image is focused. Home resets; Escape exits fullscreen. Select Learn & proofs in the header for the eight numbered proof PDFs and this study guide.

A point colored as non-escaping at N steps is only known not to have escaped during those N steps. Use a proof, rather than a dark pixel, to establish membership.