A VISUAL REFERENCE · MetaPost

MetaPost, one change at a time

From a first path to computed geometry, scientific plots and styled illustrations. Complete source. A small change. Its actual result.

The Org file is the canonical source. Read the HTML guide beside your work, or download its PDF and source for offline reading.

> **Last changes: 2026-11-01.** Added 18 drawing and styling examples: botanical and object illustrations, a coastal scene, an isometric house, stroke caps and joins, tapered ribbons, ink treatments, shadows, palette roles, road casings, hatching, halftones, guilloche, woven bands, wave tiles and circular lettering. The tour now has 102 reproducible examples in 17 chapters.

102
worked examples

17
topic families

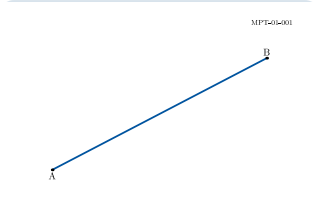
1
canonical Org source

All figures at a glance

Choose a figure to open its MetaPost source, explanation and downloads.

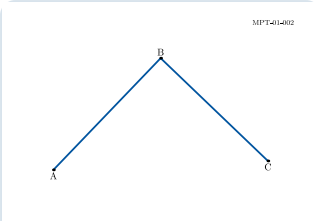
01 · First paths and coordinates

6 figures



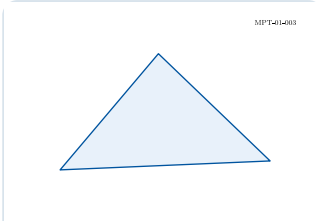
MPT-01-001

Join two coordinate pairs



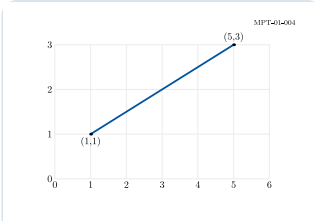
MPT-01-002

Make a polyline



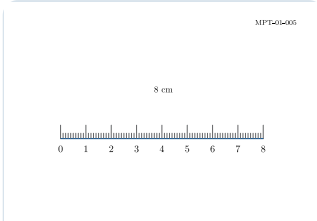
MPT-01-003

Close and fill a shape



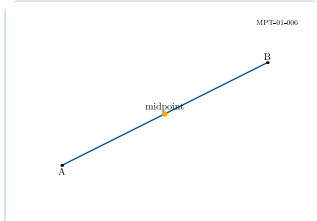
MPT-01-004

See the coordinate grid



MPT-01-005

Use centimetres and millimetres

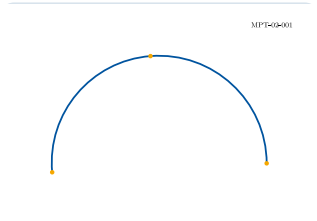


MPT-01-006

Interpolate between pairs

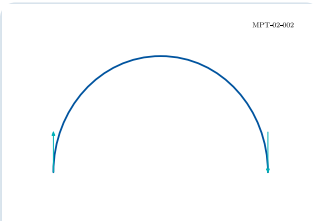
02 · Smooth curves and control points

6 figures



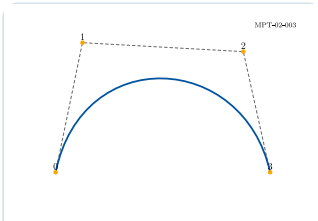
MPT-02-001

Let a smooth path pass through knots



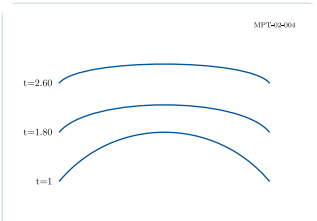
MPT-02-002

Specify endpoint directions



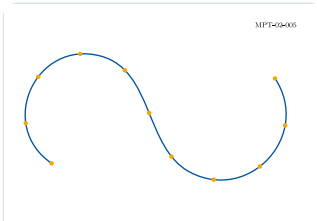
MPT-02-003

Expose cubic control handles



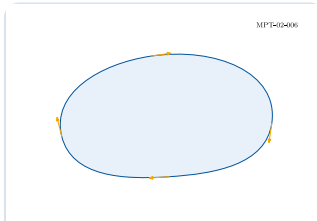
MPT-02-004

Compare path tension



MPT-02-005

Place dots by arc length

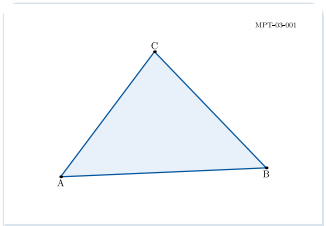


MPT-02-006

Inspect tangents on a closed path

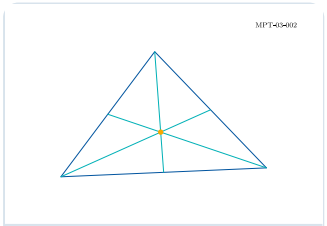
03 · Constraints and computed geometry

6 figures



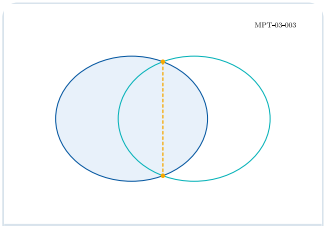
MPT-03-001

Solve a triangle from relationships



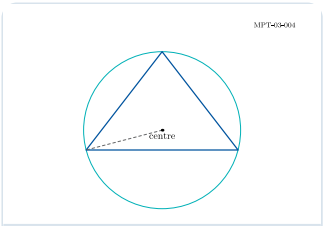
MPT-03-002

Construct medians and a centroid



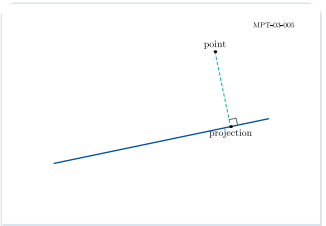
MPT-03-003

Compute the chord of two ellipses



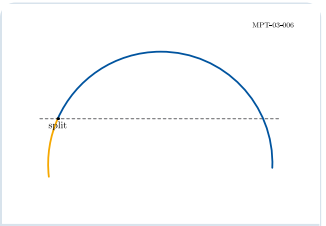
MPT-03-004

Solve perpendicular bisectors



MPT-03-005

Project onto a line

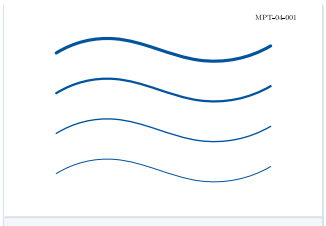


MPT-03-006

Split a curve at its intersection time

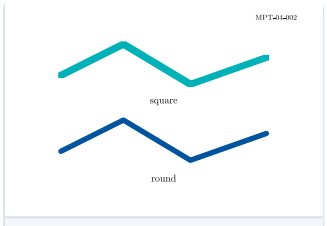
04 · Pens, dashes and colour

6 figures



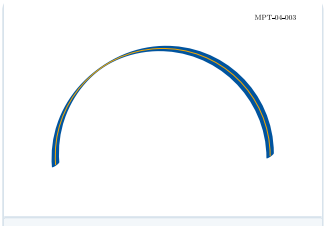
MPT-04-001

Change pen width



MPT-04-002

Compare round and square pens



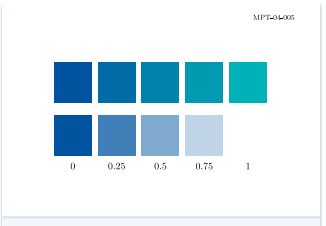
MPT-04-003

Use a broad elliptical pen



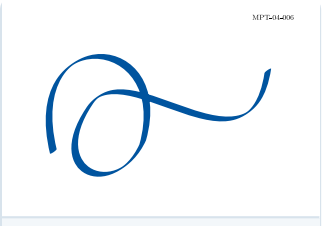
MPT-04-004

Define dash patterns



MPT-04-005

Interpolate colours and tints

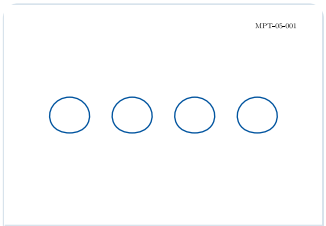


MPT-04-006

Draw a broad-nib flourish

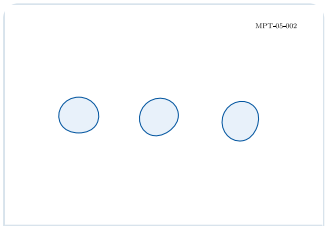
05 · Transformations and reusable pictures

6 figures



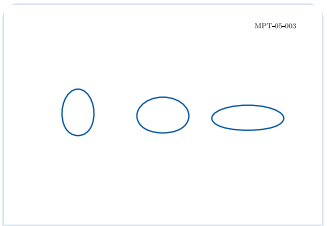
MPT-05-001

Translate a stored motif



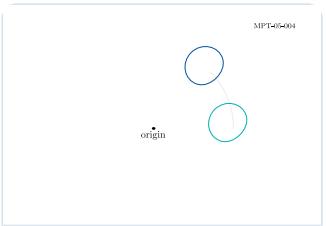
MPT-05-002

Rotate copies



MPT-05-003

Scale each coordinate independently



MPT-05-004

See why transformation order matters



MPT-05-005

Reflect an asymmetric shape

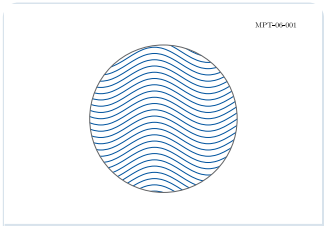


MPT-05-006

Name and reuse an affine transform

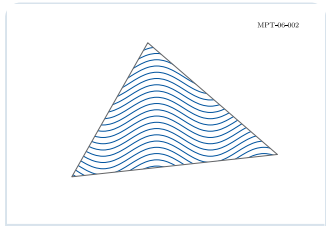
06 · Clipping, layers and cut-outs

6 figures



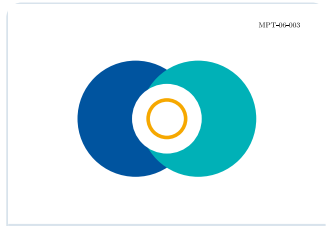
MPT-06-001

Clip a wave field to a circle



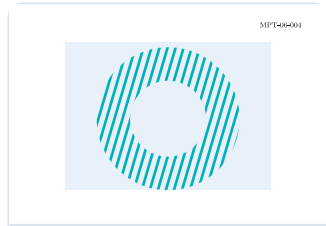
MPT-06-002

Reuse the field with another mask



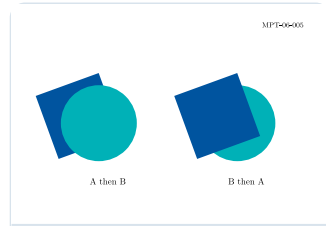
MPT-06-003

Build an emblem with opaque layers



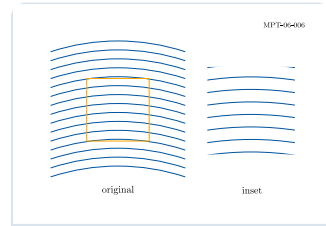
MPT-06-004

Make an annular clipping region



MPT-06-005

Change the stacking order of pictures

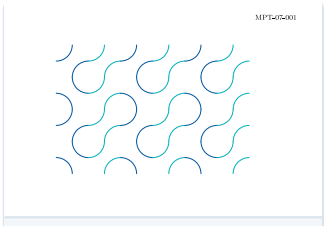


MPT-06-006

Crop a copy for a magnified inset

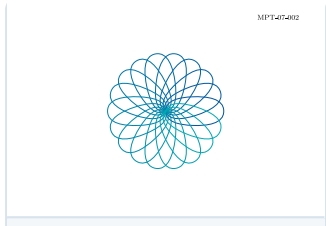
07 · Loops, macros and generative patterns

6 figures



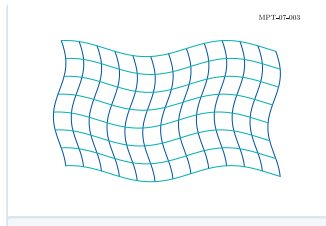
MPT-07-001

Repeat rotated quarter-circle tiles



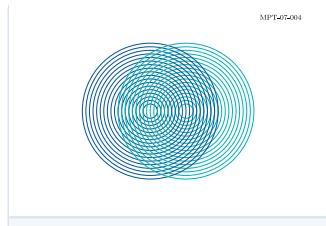
MPT-07-002

Generate a rosette from one petal



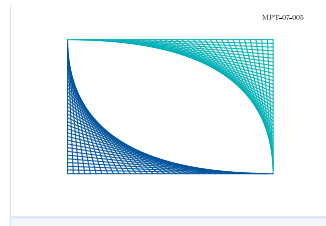
MPT-07-003

Warp a connected mesh



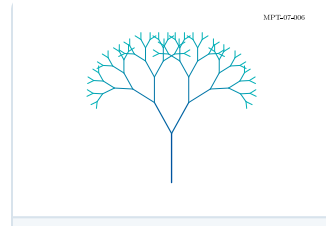
MPT-07-004

Reveal a moire field



MPT-07-005

Make curved envelopes from straight segments



MPT-07-006

Grow a bounded recursive tree

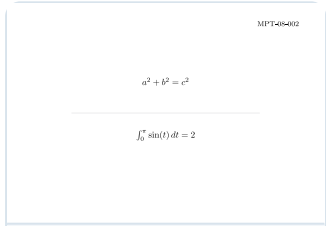
08 · Labels, mathematics and diagrams

6 figures



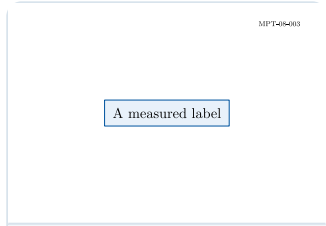
MPT-08-001

Position labels around an anchor



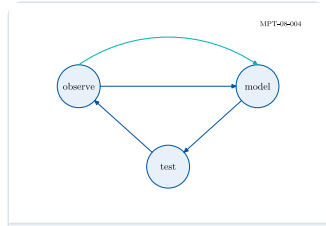
MPT-08-002

Typeset mathematics with TeX



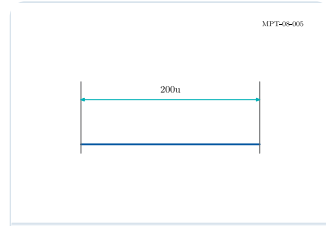
MPT-08-003

Measure and frame a label



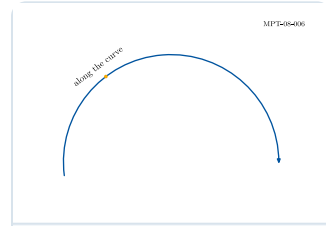
MPT-08-004

Connect nodes at their boundaries



MPT-08-005

Annotate a measured dimension

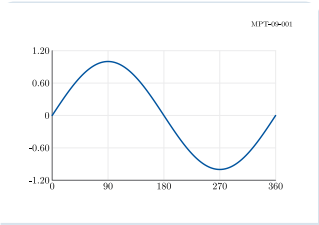


MPT-08-006

Rotate a label along a curve

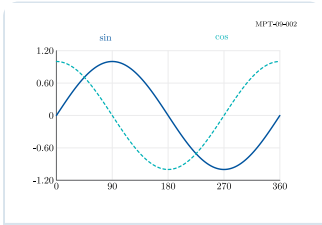
09 · Functions, envelopes and parametric curves

6 figures



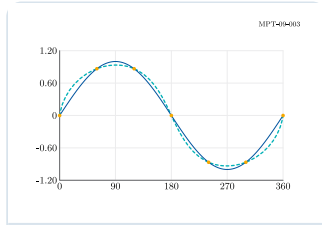
MPT-09-001

Sample a sine function



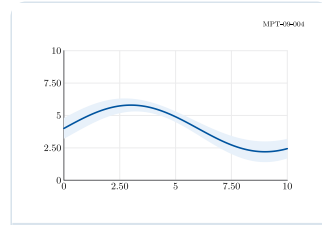
MPT-09-002

Compare two functions



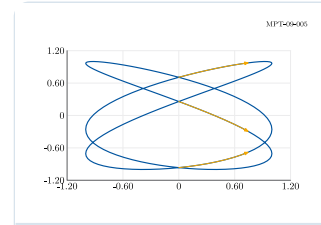
MPT-09-003

Compare knots with a smooth interpolation



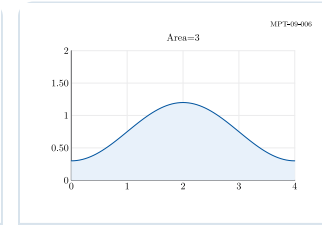
MPT-09-004

Fill between two envelopes



MPT-09-005

Draw a parametric Lissajous curve

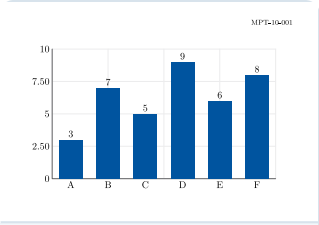


MPT-09-006

Fill an area and estimate its integral

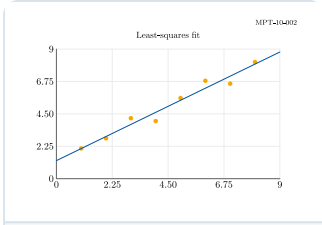
10 · Data and statistical graphics

6 figures



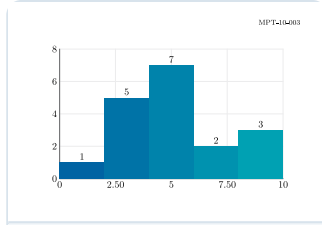
MPT-10-001

Build a labelled bar chart



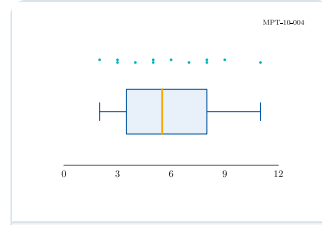
MPT-10-002

Fit a line to a scatter plot



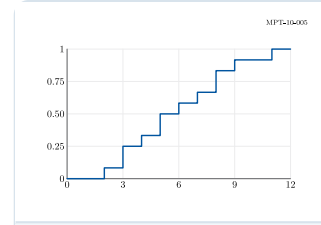
MPT-10-003

Bin raw observations into a histogram



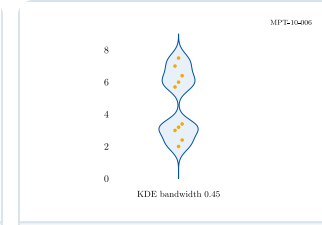
MPT-10-004

Calculate a box plot from observations



MPT-10-005

Draw an empirical cumulative distribution

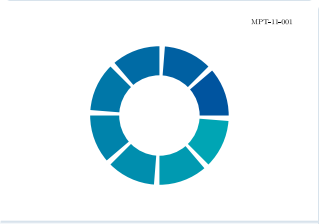


MPT-10-006

Construct an empirical violin

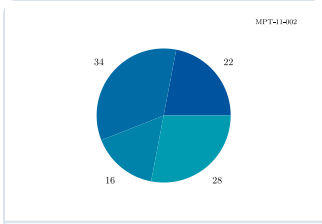
11 · Circular, polar and projected plots

6 figures



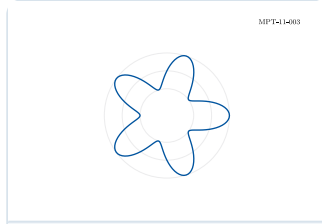
MPT-11-001

Build a segmented annulus



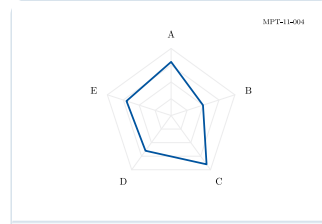
MPT-11-002

Turn values into pie-sector angles



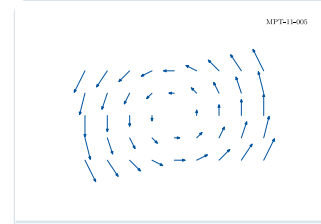
MPT-11-003

Draw a polar flower



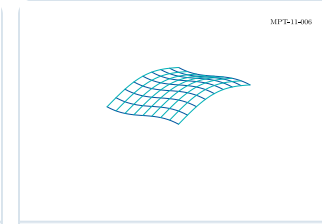
MPT-11-004

Compare values on a radar polygon



MPT-11-005

Render a rotational vector field

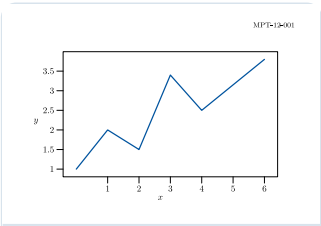


MPT-11-006

Project a wireframe surface

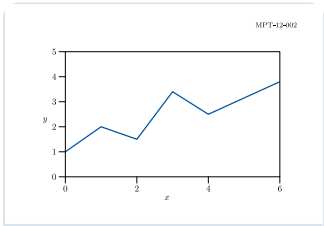
12 · The graph library and publication

6 figures



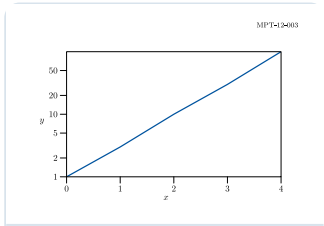
MPT-12-001

Let graph.mp choose coordinate ranges



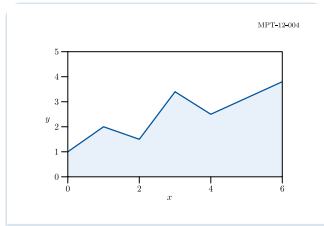
MPT-12-002

Set explicit graph ranges



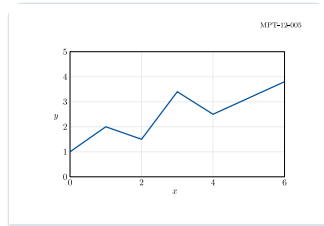
MPT-12-003

Use a logarithmic graph axis



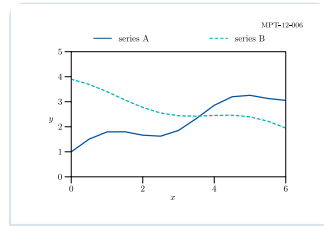
MPT-12-004

Fill data in graph coordinates



MPT-12-005

Add an automatic graph grid

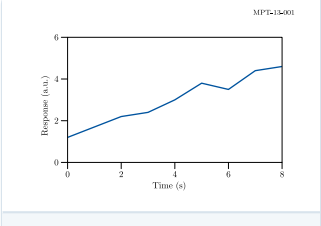


MPT-12-006

Compose two data series and a legend

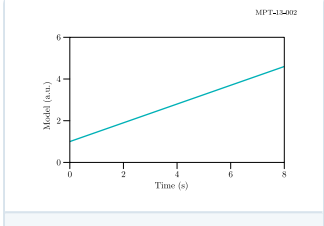
13 · External data and incomplete observations

6 figures



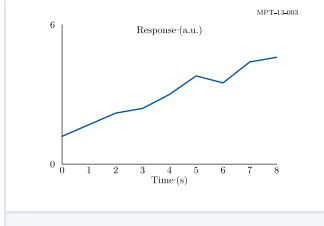
MPT-13-001

Plot a two-column file with graph.mp



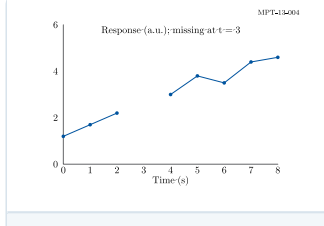
MPT-13-002

Select columns from a wider table



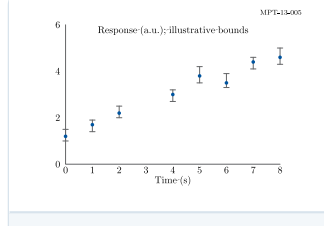
MPT-13-003

Read pairs and skip an empty line



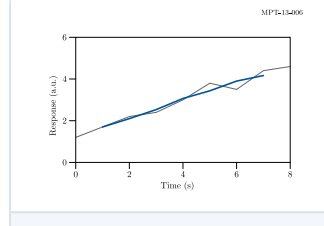
MPT-13-004

Leave a gap for a missing observation



MPT-13-005

Draw asymmetric intervals from columns

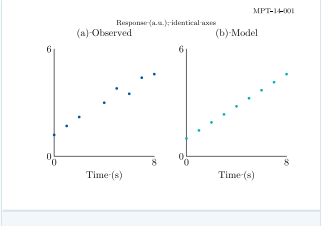


MPT-13-006

Compute a centred moving mean

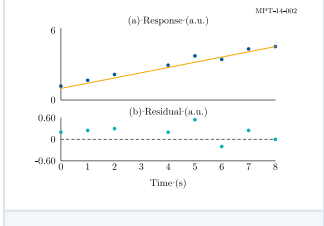
14 · Complete publication figures

6 figures



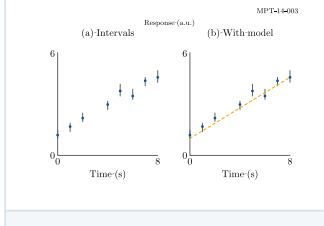
MPT-14-001

Give two panels the same scale



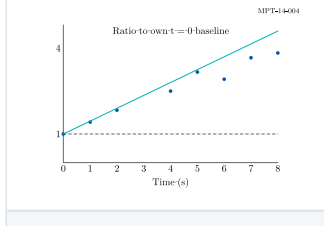
MPT-14-002

Align observations and residuals



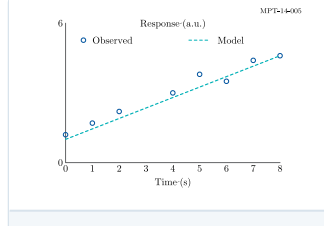
MPT-14-003

Reuse panel pictures without rescaling text



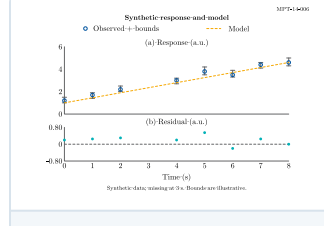
MPT-14-004

Normalise two series to a shared baseline



MPT-14-005

Make the legend survive greyscale printing

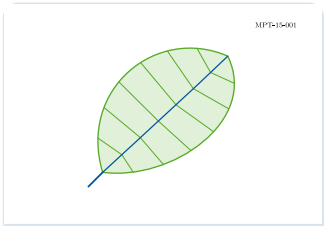


MPT-14-006

Assemble a titled figure with an audit trail

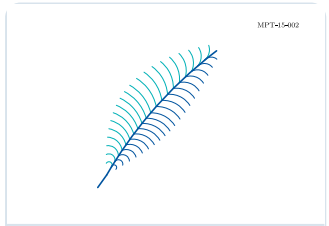
15 · Drawing recognisable objects

6 figures



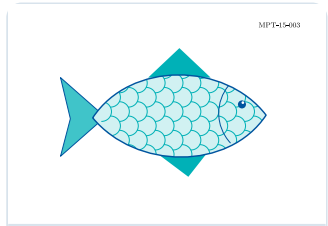
MPT-15-001

Draw a leaf from a silhouette and veins



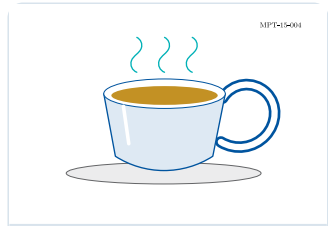
MPT-15-002

Grow a feather along a curved spine



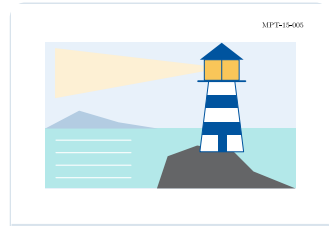
MPT-15-003

Clip a scale texture inside a fish



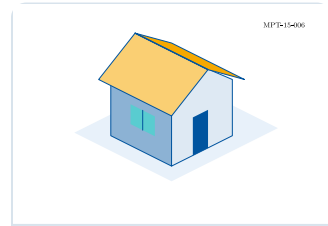
MPT-15-004

Assemble a cup with a handle and steam



MPT-15-005

Build a coastal scene in depth layers

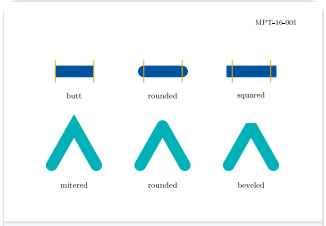


MPT-15-006

Project a small isometric house

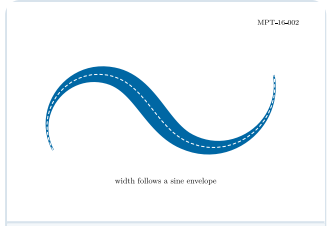
16 · Stroke styles and visual systems

6 figures



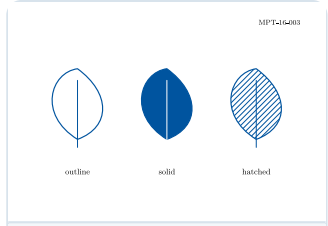
MPT-16-001

Compare line caps and corner joins



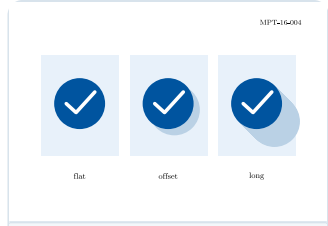
MPT-16-002

Construct a tapered ribbon around a path



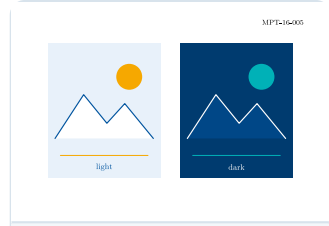
MPT-16-003

Give one motif three ink treatments



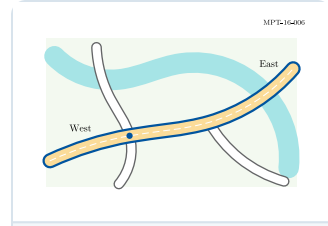
MPT-16-004

Compare flat, offset and long shadows



MPT-16-005

Assign colour roles for light and dark scenes

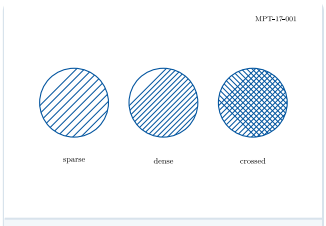


MPT-16-006

Draw a map with cased roads and a bridge

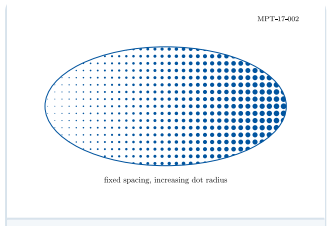
17 · Textures, ornament and lettering

6 figures



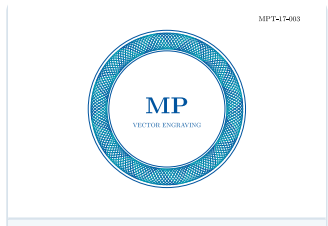
MPT-17-001

Shade three discs with clipped hatching



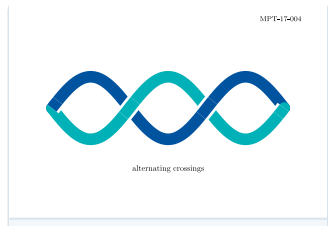
MPT-17-002

Make a halftone field from variable dots



MPT-17-003

Engrave a guilloche border with periodic curves



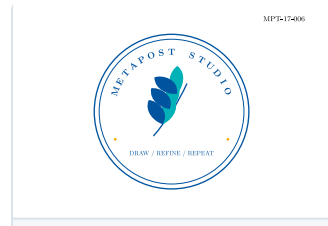
MPT-17-004

Alternate over and under in a woven band



MPT-17-005

Tile a field with nested wave arcs



MPT-17-006

Place lettering around an illustrated seal

READING THE TOUR

Three ways to see a change

MetaPost source

Read the drawing commands and expand the complete standalone program.

Actual vector figure

Compare native MetaPost output with its control points, geometry and data.

One change

Follow a comparison, download the source and reuse SVG, vector PDF or PNG output.

Read from the beginning or jump to a stable example ID. Each example includes its complete input and captured result. Copy link preserves the address of the guide you are reading, including its current domain and path.

The first chapters develop paths, geometric constraints and the picture language. The complete programs define numlabel to round displayed numbers to hundredths, keeping floating-point calculation separate from labels. Later chapters build data graphics from explicit calculations and introduce the standard graph.mp library. The sixteen compositions in the pinned ROOT MetaPost atlas supply the starting ideas; this tour develops them into independent, explained examples and adds scientific plots. Chapters 15-17 return to illustration, stroke styles, textures, ornament and lettering. Start there for drawing recipes and follow their comparisons back to the underlying path, pen and clipping commands.

Every entry shows its focused drawing commands. Expand the complete program or download the .mp source to see the shared white canvas, explicit RWTH colours, output filename and visible example ID. Run a downloaded program with `mpost -numbersystem=double MPT-01-001.mp`.

Coordinates normally grow rightwards and upwards. Trigonometric functions `sind`, `cosd` and `dir` use degrees. The complete programs use native SVG output with outlined font glyphs; their vector PDFs and PNG previews come from that same vector figure. Each vector PDF also carries its complete .mp source.

Chapters 13 and 14 read the bundled synthetic files in data/. Download and extract the source-and-data ZIP, keep its directory structure and run from the extracted root. Their figure PDFs embed both source and declared inputs. The guide PDF attaches an aggregate data-and-source archive for offline reproduction.

References and provenance

- Starting atlas: ROOT MetaPost, pinned commit 26039d49fe2e6399552114160e7c7850fd6c4fb9
- MetaPost User's Manual: paths, equations, pictures, labels and output
- Drawing Graphs with MetaPost
- TeX Users Group MetaPost resources

The atlas's sixteen showcase ideas are developed here with independent complete programs, new coordinates, explanations and comparisons. The poster builder is not imported. Source-to-atlas panel mappings are retained in the Org properties and sources/atlas-provenance.json. All additional plotting fixtures are small and synthetic. MetaPost and the TeX distribution remain external dependencies; their executables, libraries and font files are not redistributed as source code in this tour.

01 · First paths and coordinates

MPT-01-001

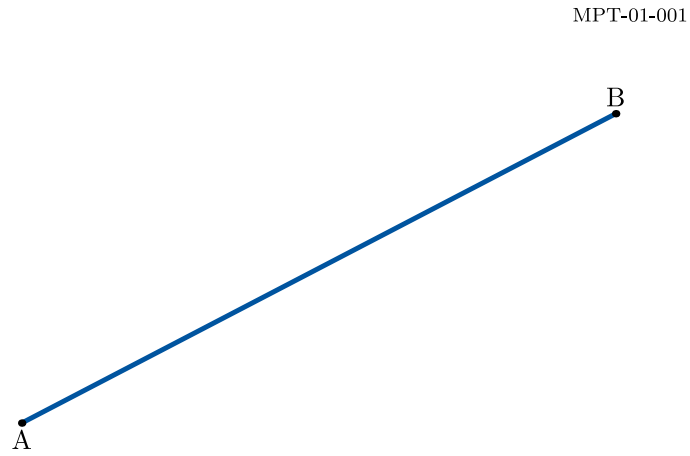
Join two coordinate pairs

A straight segment is already a complete vector figure. The coordinates specify its two endpoints; the round pen controls its thickness.

MetaPost drawing

```
draw (40,45)--(280,170) withpen pencircle scaled 2 withcolor Blue;
dotlabel.bot("A",(40,45)); dotlabel.top("B",(280,170));
```

Actual output



MPT-01-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`(x,y)`	A coordinate pair places a point in the drawing plane.
`--`	Join the two endpoints with a straight segment.
`withpen`	Choose the stroke independently of the geometry.

MPT-01-002

Make a polyline

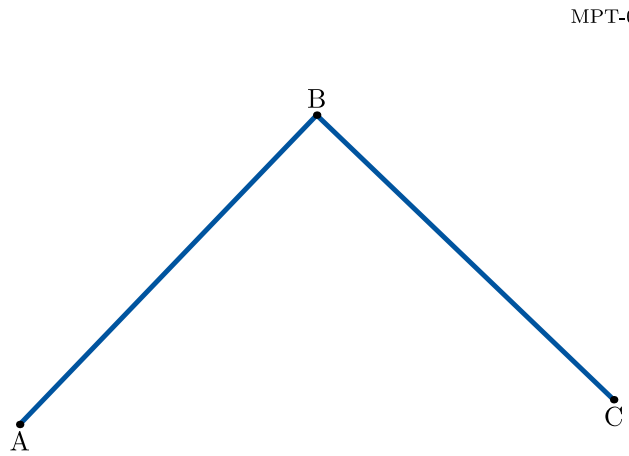
Add a middle point. The same straight-line connector can form a broken line, and dotlabel makes each vertex visible.

Compare with MPT-01-001 · Join two coordinate pairs.

MetaPost drawing

```
draw (40,45)--(160,170)--(280,55) withpen pencircle scaled 2 withcolor Blue;
dotlabel.bot("A",(40,45)); dotlabel.top("B",(160,170));
dotlabel.bot("C",(280,55));
```

Actual output



MPT-01-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`-- between three pairs`	Two independent straight segments meet at the middle vertex.
`dotlabel`	Draw a dot and place its label on the requested side.

MPT-01-003

Close and fill a shape

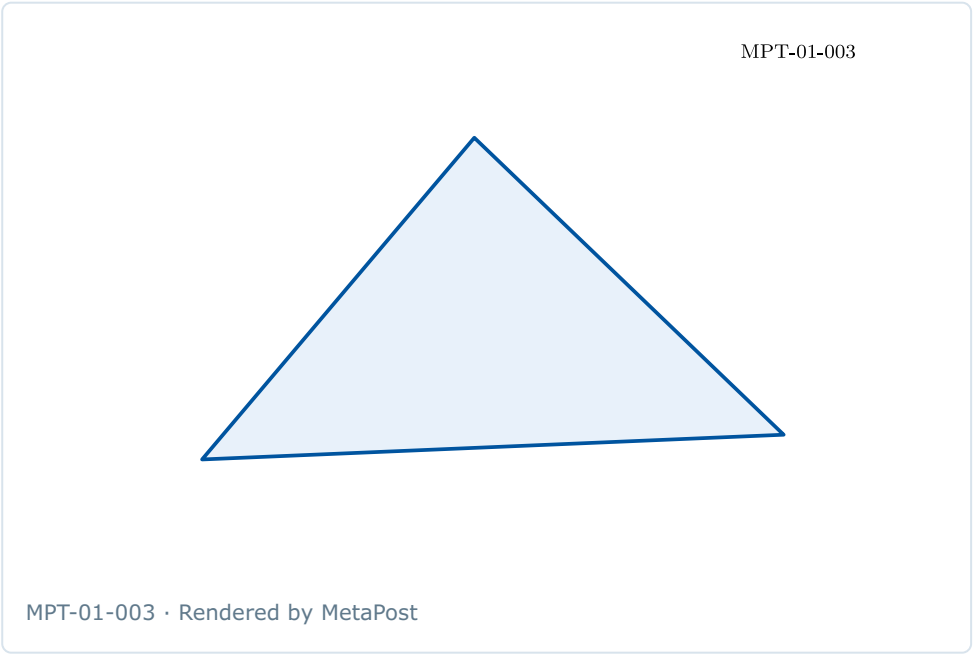
A closed path can be filled as well as stroked. Keep the fill pale so that the outline remains readable.

Compare with MPT-01-002 · Make a polyline.

MetaPost drawing

```
path p;
p:=(45,45)--(155,175)--(280,55)--cycle;
fill p withcolor Wash;
draw p withpen pencircle scaled 1.5 withcolor Blue;
```

Actual output



MPT-01-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`path p`	Store geometry so that fill and draw use the same boundary.
`cycle`	Join the last vertex back to the first.
`fill / draw`	Fill the interior first, then draw the outline.

MPT-01-004

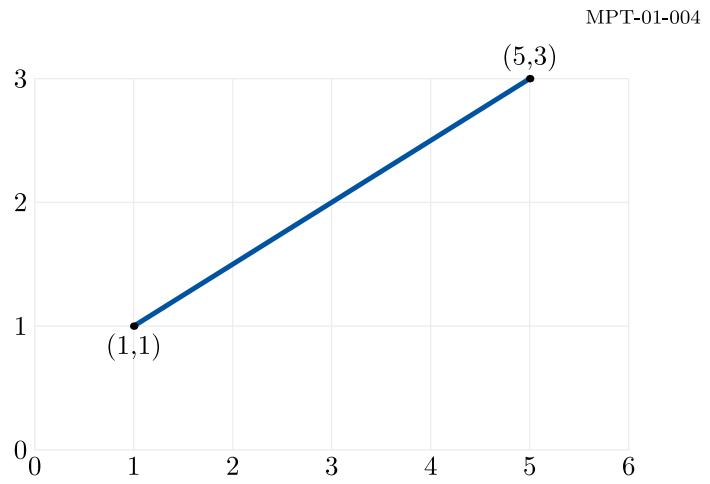
See the coordinate grid

A light grid makes positions explicit. MetaPost supplies loops and arithmetic, so reference lines and their labels can follow the same numbers.

MetaPost drawing

```
for i=0 upto 6:
 draw (40+40i,35)--(40+40i,185) withcolor Light;
 label.bot(decimal i,(40+40i,35));
endfor
for j=0 upto 3:
 draw (40,35+50j)--(280,35+50j) withcolor Light;
 label.lft(decimal j,(40,35+50j));
endfor
draw (80,85)--(240,185) withpen pencircle scaled 2 withcolor Blue;
dotlabel.bot("(1,1)",(80,85)); dotlabel.top("(5,3)",(240,185));
```

Actual output



MPT-01-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`for ... upto`	Generate reference lines from integer indices.
`decimal`	Convert a numeric value into a label string.
`40+40i / 35+50j`	Map logical coordinates onto a rectangular plotting area.

MPT-01-005

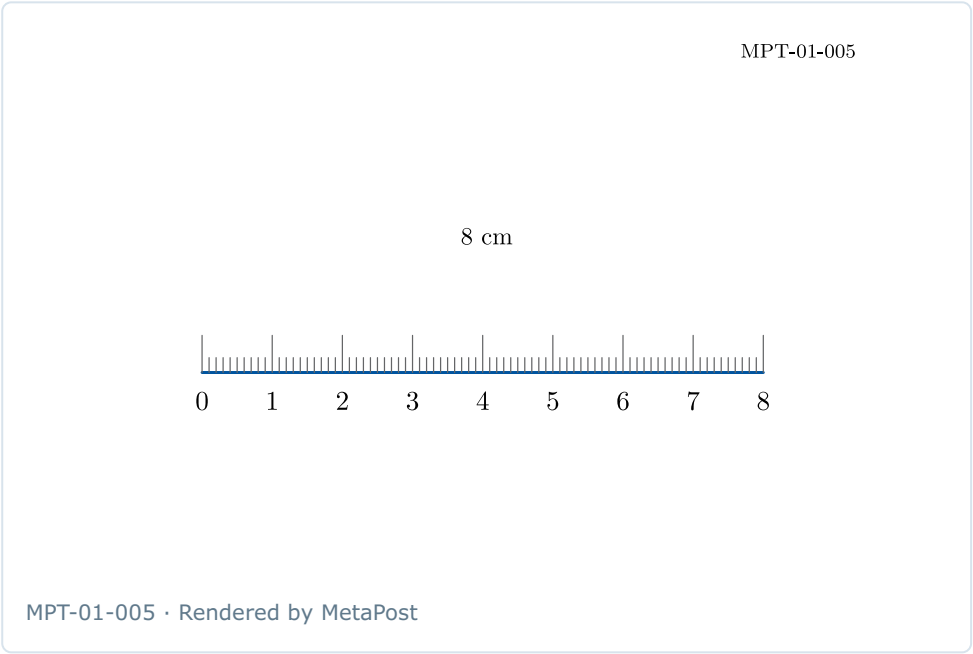
Use centimetres and millimetres

MetaPost has physical units. This ruler is eight centimetres long, with a mark for every millimetre and a number for every centimetre.

MetaPost drawing

```
numeric u; u:=cm;
draw (45,80)--(45+8u,80) withpen pencircle scaled 1.2 withcolor Blue;
for i=0 upto 80:
 draw (45+i*mm,80)--(45+i*mm,80+if i mod 10=0: 15 else: 6 fi)
   withcolor Grey;
 if i mod 10=0: label.bot(decimal(i/10),(45+i*mm,75)); fi
endfor
label(btex 8 cm etex,(160,135));
```

Actual output



MPT-01-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`cm` / `mm`	Use the unit constants rather than guessing a conversion.
`mod`	Choose longer marks at whole centimetres.
`if ... else ... fi`	Select a tick length without duplicating the drawing.

MPT-01-006

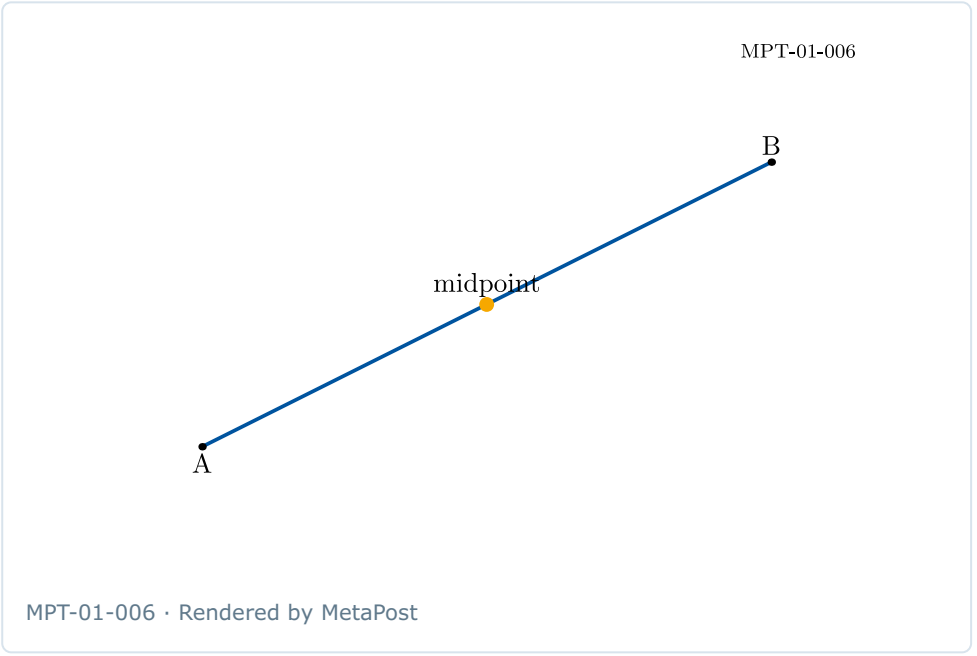
Interpolate between pairs

The notation t[A,B] locates a point along a segment. The midpoint uses t=0.5; extrapolation is possible with values outside zero to one.

MetaPost drawing

```
pair a,b,m;
a:=(45,50); b:=(275,165); m:=.5[a,b];
draw a--b withpen pencircle scaled 1.5 withcolor Blue;
dotlabel.bot("A",a); dotlabel.top("B",b);
fill fullcircle scaled 6 shifted m withcolor Orange;
label.top("midpoint",m);
```

Actual output



MPT-01-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`pair`	A two-dimensional vector or position.
`.5[a,b]`	Compute the midpoint with linear interpolation.
`shifted m`	Move a circle centred at the origin onto the calculated point.

02 · Smooth curves and control points

MPT-02-001

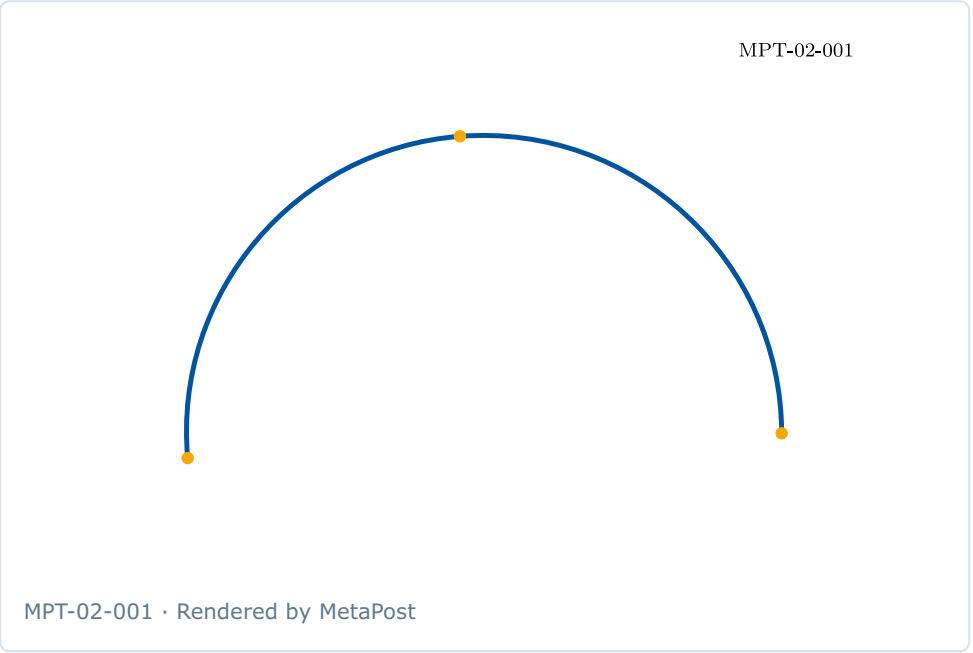
Let a smooth path pass through knots

Replace straight connectors with smooth ones. MetaPost determines cubic controls through the supplied knots.

MetaPost drawing

```
draw (40,45)..(150,175)..(280,55) withpen pencircle scaled 2 withcolor Blue;
for p=(40,45),(150,175),(280,55):
 fill fullcircle scaled 5 shifted p withcolor Orange;
endfor
```

Actual output



MPT-02-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`..`	Ask for a smooth cubic path through the knots.
`Knots`	The orange dots are interpolation points, not control handles.

MPT-02-002

Specify endpoint directions

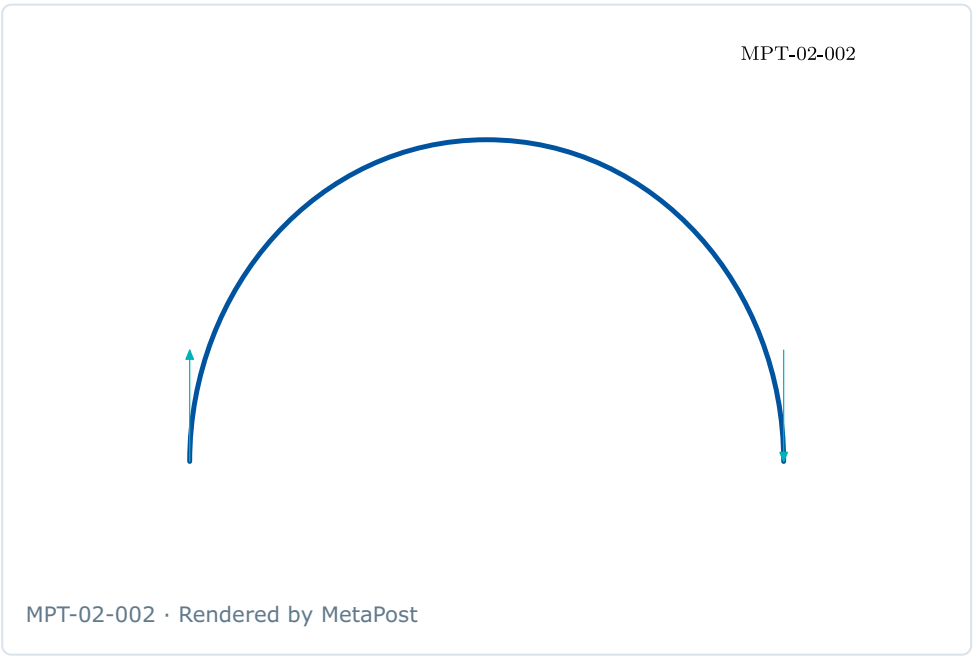
The same endpoints now have explicit upward and downward tangents. Directions constrain the shape without specifying every control point.

Compare with MPT-02-001 · Let a smooth path pass through knots.

MetaPost drawing

```
draw (40,45){up}..(160,175)..{down}(280,45)
 withpen pencircle scaled 2 withcolor Blue;
drawarrow (40,45)--(40,90) withcolor Teal;
drawarrow (280,90)--(280,45) withcolor Teal;
```

Actual output



MPT-02-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`{up} / {down}`	Constrain the tangent direction at the adjacent endpoint.
`drawarrow`	Make the prescribed directions visible.

MPT-02-003

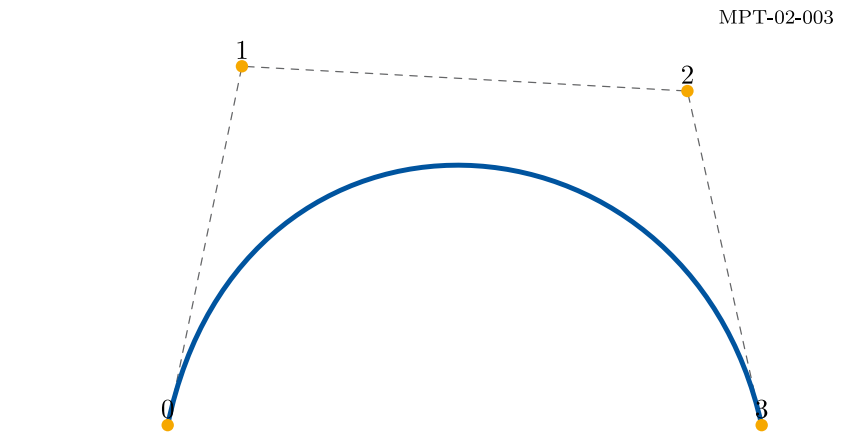
Expose cubic control handles

A cubic Bezier segment uses two control handles. Its dashed control polygon makes the distinction between a knot and a handle visible.

MetaPost drawing

```
z0=(40,45); z1=(70,190); z2=(250,180); z3=(280,45);
draw z0--z1--z2--z3 dashed evenly withcolor Grey;
draw z0..controls z1 and z2..z3 withpen pencircle scaled 2 withcolor Blue;
for i=0 upto 3:
 fill fullcircle scaled 5 shifted z[i] withcolor Orange;
 label.top(decimal i,z[i]);
endfor
```

Actual output



MPT-02-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`controls z1 and z2`	Specify both Bezier controls instead of letting MetaPost choose them.
`z[i]`	The standard paired-coordinate notation also supports indexed points.

MPT-02-004

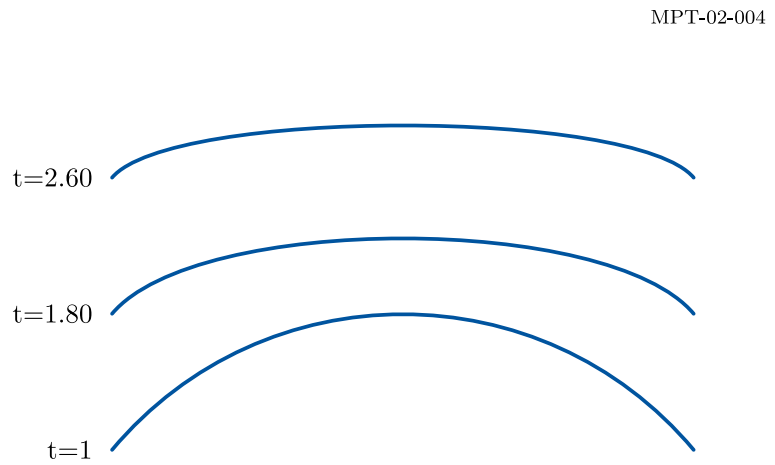
Compare path tension

Keep the endpoints and directions but change tension. The three rows show how the curve draws closer to a direct connection as tension increases.

MetaPost drawing

```
for k=0 upto 2:
 draw (45,35+55k){dir 50}..tension (1+.8k)..{dir -50}(280,35+55k)
  withpen pencircle scaled 1.5 withcolor Blue;
 label.lft("t="&numlabel(1+.8k),(40,35+55k));
endfor
```

Actual output



MPT-02-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`tension`	Control how tightly the interpolating segment approaches its chord.
`dir angle`	Create a direction vector from an angle in degrees.
`Same endpoints`	Only tension changes within each translated row.

MPT-02-005

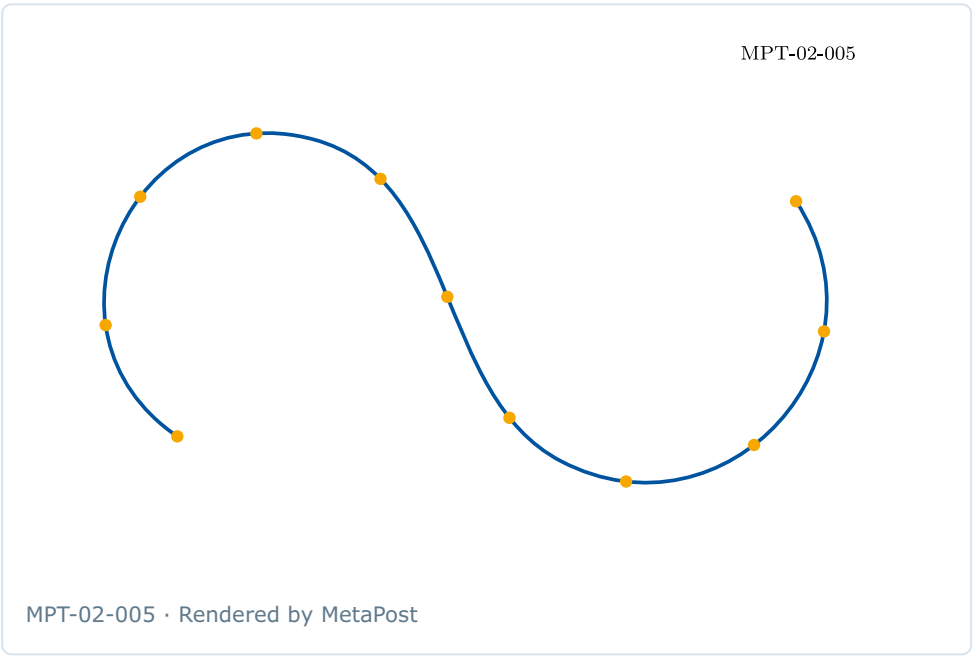
Place dots by arc length

Path time and distance along a curve are different. arctime converts equal distances into path parameters before the points are drawn.

MetaPost drawing

```
path p; numeric t;
p:=(35,55)..(90,175)..(190,45)..(285,150);
draw p withpen pencircle scaled 1.5 withcolor Blue;
for k=0 upto 10:
 t:=arctime (k*arclength p/10) of p;
 fill fullcircle scaled 5 shifted (point t of p) withcolor Orange;
endfor
```

Actual output



MPT-02-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`arclength p`	Measure the length of the entire path.
`arctime ... of p`	Find the path time at the requested distance.
`point t of p`	Evaluate the path at that time.

MPT-02-006

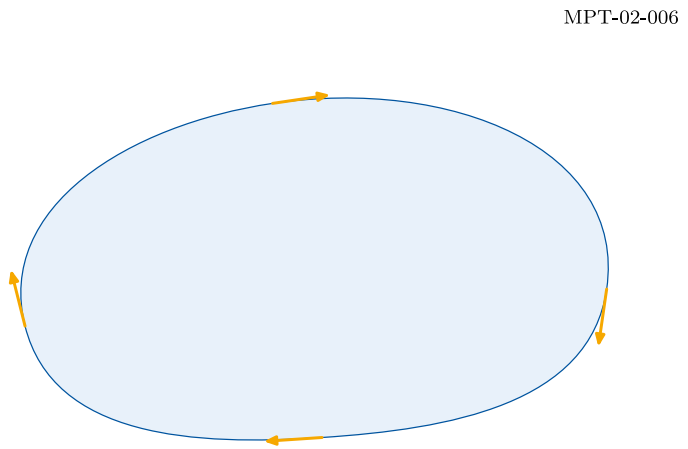
Inspect tangents on a closed path

Smooth closed paths carry tangent directions just like open curves. Short arrows show the direction at each knot.

MetaPost drawing

```
path p; pair a,t;
p:=(45,85)..(145,175)..(280,100)..(165,40)..cycle;
fill p withcolor Wash; draw p withcolor Blue;
for k=0 upto length p-1:
 a:=point k of p; t:=unitvector(direction k of p);
 drawarrow a--(a+22t) withpen pencircle scaled 1.2 withcolor Orange;
endfor
```

Actual output



MPT-02-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`..cycle`	Close the curve smoothly.
`direction`	Obtain a tangent vector at a path time.
`unitvector`	Normalize a nonzero direction before choosing arrow length.

03 · Constraints and computed geometry

MPT-03-001

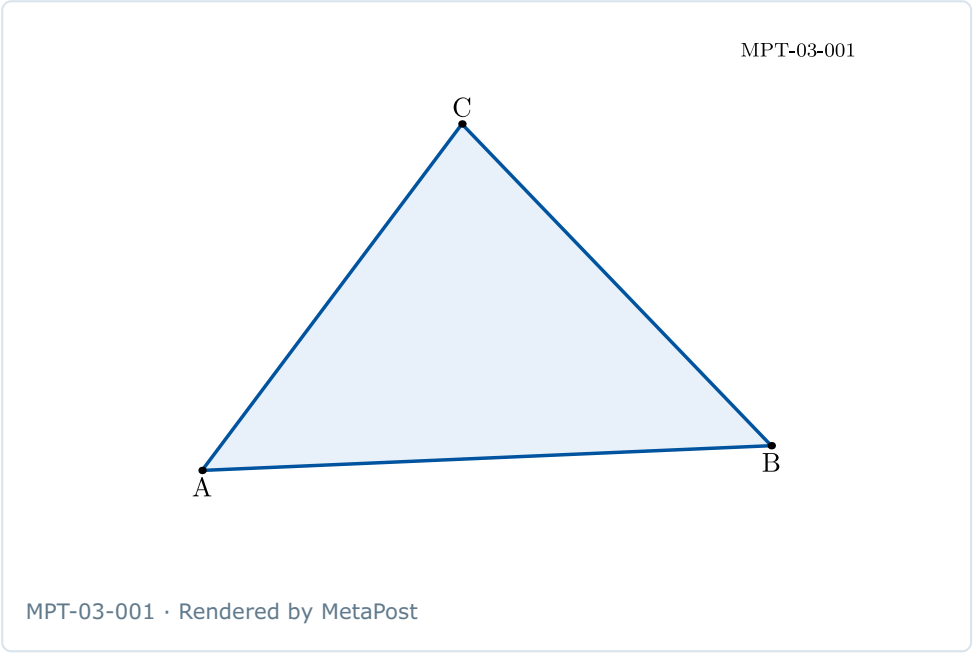
Solve a triangle from relationships

Equations can determine positions. Two vector differences and one anchor are enough to place these three vertices.

MetaPost drawing

```
z0=(45,40); z1-z0=(230,10); z2-z0=(105,140);
fill z0--z1--z2--cycle withcolor Wash;
draw z0--z1--z2--cycle withpen pencircle scaled 1.4 withcolor Blue;
dotlabel.bot("A",z0); dotlabel.bot("B",z1); dotlabel.top("C",z2);
```

Actual output



MPT-03-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
	State a relationship for the linear equation solver.
`z1-z0`	A vector displacement can determine both coordinates.

MPT-03-002

Construct medians and a centroid

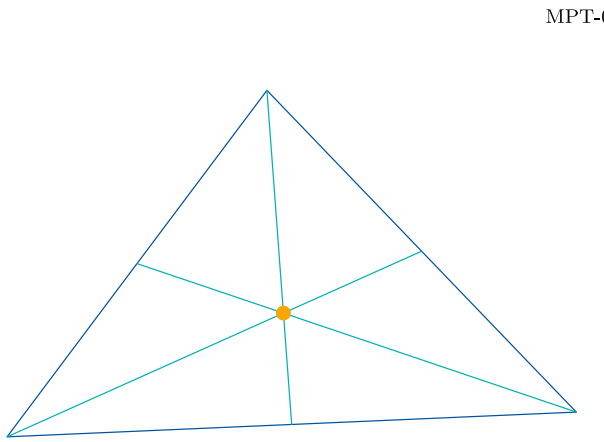
Midpoints and the centroid follow from the triangle. Moving a vertex changes the dependent construction automatically.

Compare with MPT-03-001 · Solve a triangle from relationships.

MetaPost drawing

```
z0=(45,40); z1=(275,50); z2=(150,180);
z3=.5[z0,z1]; z4=.5[z1,z2]; z5=.5[z2,z0];
draw z0--z1--z2--cycle withcolor Blue;
draw z0--z4 withcolor Teal; draw z1--z5 withcolor Teal;
draw z2--z3 withcolor Teal;
fill fullcircle scaled 6 shifted ((z0+z1+z2)/3) withcolor Orange;
```

Actual output



MPT-03-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`.5[...]`	Compute the opposite side midpoints.
`(z0+z1+z2)/3`	Compute the centroid from the three vertices.

MPT-03-003

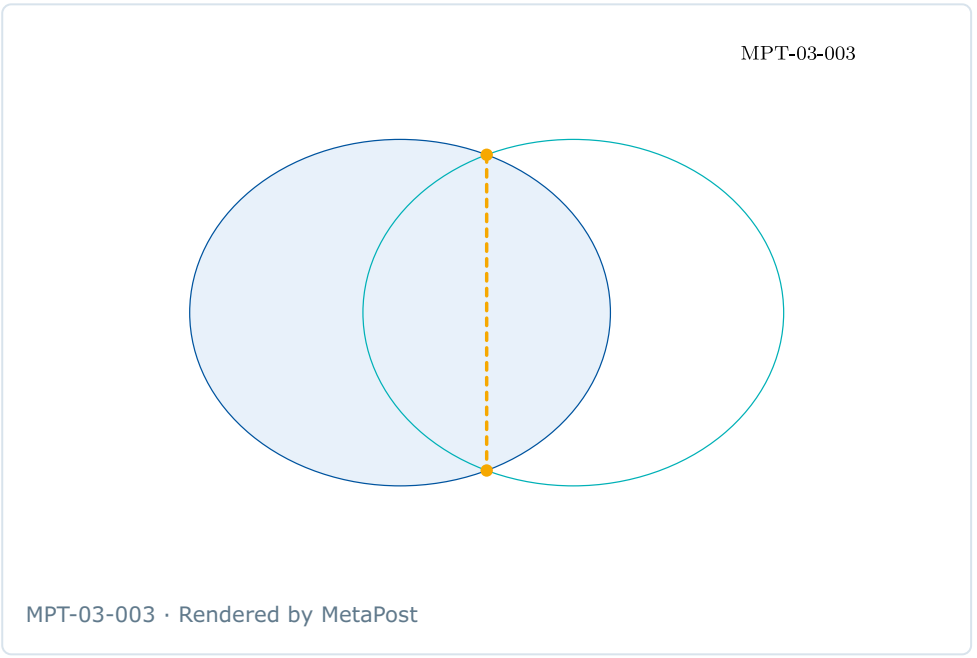
Compute the chord of two ellipses

The endpoints of the common chord are calculated from the paths. Reversing the first path finds the other intersection.

MetaPost drawing

```
path a,b; pair p,q;
a:=fullcircle xscaled 170 yscaled 140 shifted (125,105);
b:=fullcircle xscaled 170 yscaled 140 shifted (195,105);
p:=a intersectionpoint b; q:=(reverse a) intersectionpoint b;
fill a withcolor Wash;
draw a withcolor Blue; draw b withcolor Teal;
draw p--q dashed evenly withpen pencircle scaled 1.4 withcolor Orange;
fill fullcircle scaled 5 shifted p withcolor Orange;
fill fullcircle scaled 5 shifted q withcolor Orange;
```

Actual output



MPT-03-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`intersectionpoint`	Find a point common to the two paths.
`reverse`	Change path traversal order to select the other crossing.

MPT-03-004

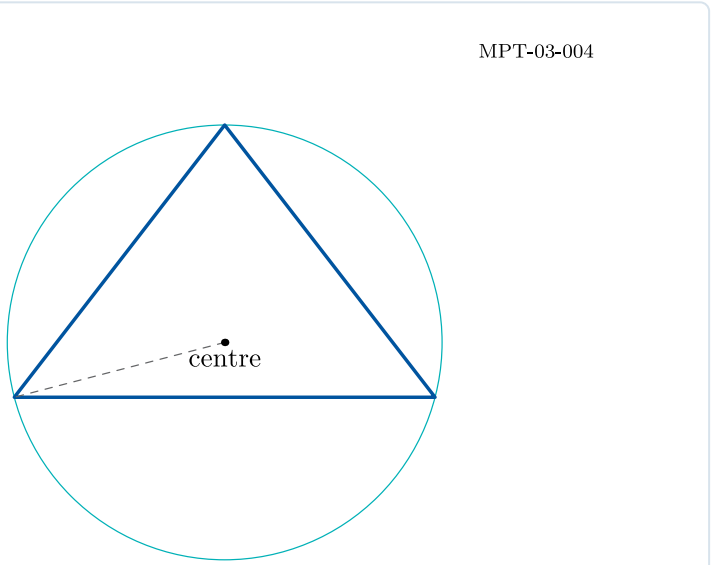
Solve perpendicular bisectors

The unknown centre lies on two perpendicular bisectors. Each occurrence of whatever introduces a fresh scalar unknown for the solver.

MetaPost drawing

```
pair a,b,c,o;
a:=(75,70); b:=(245,70); c:=(160,180);
o=.5[a,b]+whatever*((b-a) rotated 90);
o=.5[a,c]+whatever*((c-a) rotated 90);
draw fullcircle scaled (2*abs(o-a)) shifted o withcolor Teal;
draw a--b--c--cycle withpen pencircle scaled 1.4 withcolor Blue;
draw o--a dashed evenly withcolor Grey;
dotlabel.bot("centre",o);
```

Actual output



MPT-03-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`whatever`	Introduce an independent unknown scalar in each equation.
`rotated 90`	A perpendicular direction to a side.
`abs(o-a)`	The distance from the solved centre to a vertex.

MPT-03-005

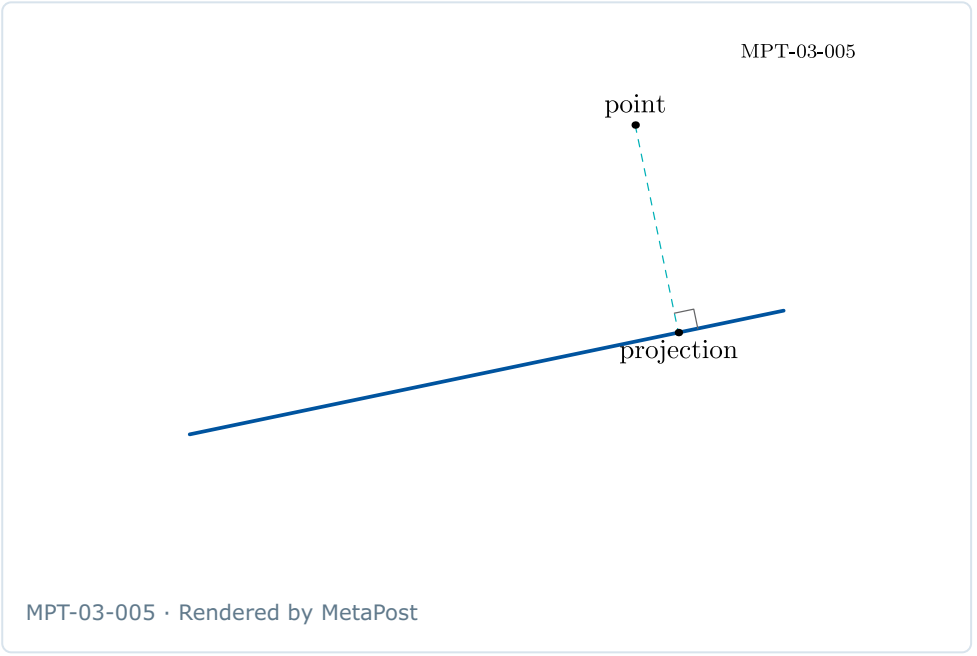
Project onto a line

The foot lies on the line and the connecting segment is perpendicular to it. Those two constraints determine the projection.

MetaPost drawing

```
pair a,b,p,foot,u,v;
a:=(40,55); b:=(280,105); p:=(220,180);
foot=a+whatever*(b-a);
p-foot=whatever*((b-a) rotated 90);
draw a--b withpen pencircle scaled 1.5 withcolor Blue;
draw p--foot dashed evenly withcolor Teal;
u:=8*unitvector(b-a); v:=8*unitvector(p-foot);
draw (foot+u)--(foot+u+v)--(foot+v) withcolor Grey;
dotlabel.top("point",p); dotlabel.bot("projection",foot);
```

Actual output



MPT-03-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Two pair equations`	Constrain the foot to the line and its displacement to the normal.
`unitvector`	Build an eight-unit right-angle marker independent of line length.

MPT-03-006

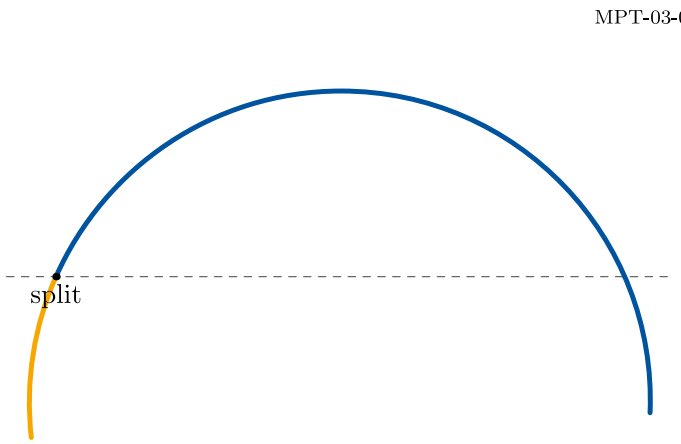
Split a curve at its intersection time

An intersection returns a time on each path. Use the first component as a subpath boundary; it is not an x coordinate.

MetaPost drawing

```
path p,line; pair times,q;
p:=(35,40)..(160,180)..(285,50); line:=(25,105)--(295,105);
times:=p intersectiontimes line;
draw line dashed evenly withcolor Grey;
draw subpath (0,xpart times) of p withpen pencircle scaled 2 withcolor Orange;
draw subpath (xpart times,length p) of p withpen pencircle scaled 2 withcolor Blue;
q:=point (xpart times) of p;
dotlabel.bot("split",q);
```

Actual output



MPT-03-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`intersectiontimes`	Return the two traversal parameters at a crossing.
`subpath`	Extract a part of the original curve without fitting a replacement.

04 · Pens, dashes and colour

MPT-04-001

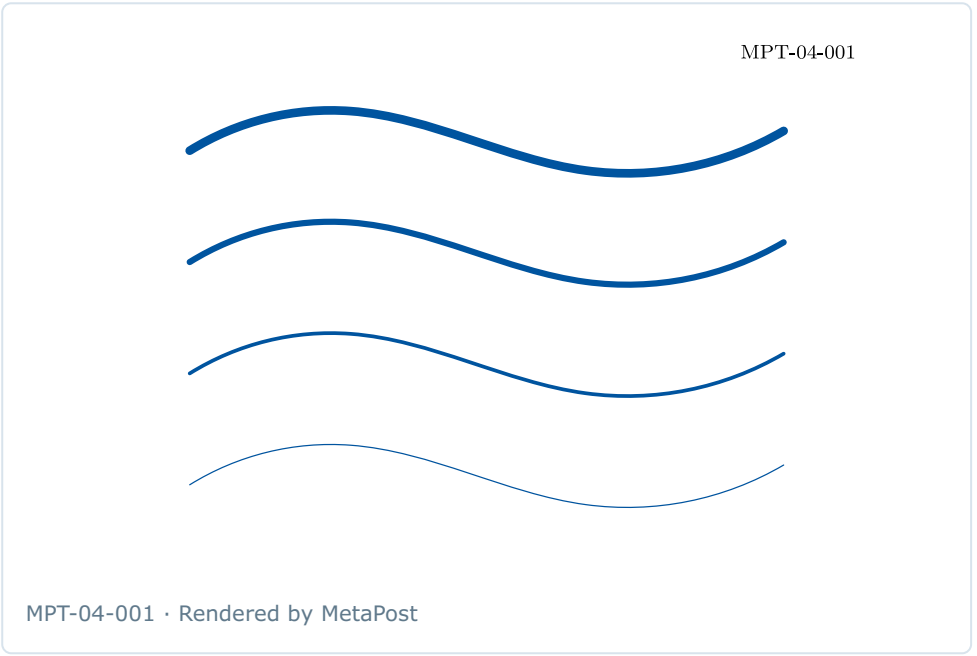
Change pen width

The same path is repeated at four heights. Only the round pen width changes.

MetaPost drawing

```
path p; p:=(40,0)..(105,16)..(200,-8)..(280,8);
for k=0 upto 3:
 draw p shifted (0,35+45k) withpen pencircle scaled (.5+k)
  withcolor Blue;
endfor
```

Actual output



MPT-04-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`pencircle scaled w`	Choose a round pen of width w.
`shifted`	Translate copies without changing their geometry.

MPT-04-002

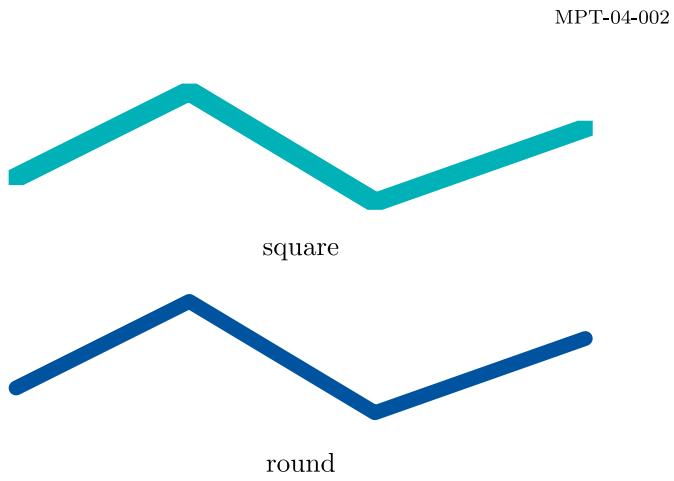
Compare round and square pens

Pen shape changes the stroke even when the centre line is identical.

MetaPost drawing

```
path p; p:=(45,0)--(115,35)--(190,-10)--(275,20);
draw p shifted (0,60) withpen pencircle scaled 6 withcolor Blue;
draw p shifted (0,145) withpen pensquare scaled 6 withcolor Teal;
label("round",(160,30)); label("square",(160,115));
```

Actual output



MPT-04-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`pencircle / pensquare`	Choose the shape swept along the path.
`Same centre line`	The difference comes from the pen rather than different coordinates.

MPT-04-003

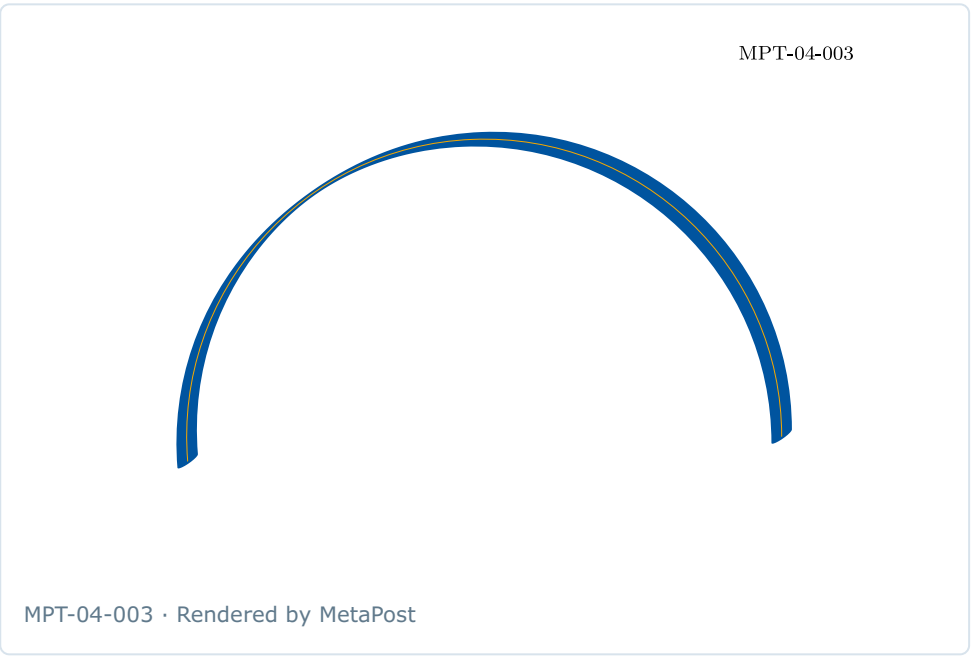
Use a broad elliptical pen

An anisotropic, rotated pen creates a broad-nib stroke. A fine centre line makes its construction visible.

MetaPost drawing

```
path p; p:=(40,45)..(155,175)..(280,55);
draw p withpen pencircle xscaled 10 yscaled 2 rotated 35 withcolor Blue;
draw p withpen pencircle scaled .4 withcolor Orange;
```

Actual output



MPT-04-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`xscaled / yscaled`	Give the pen unequal dimensions.
`rotated 35`	Rotate the nib, not the centre line.

MPT-04-004

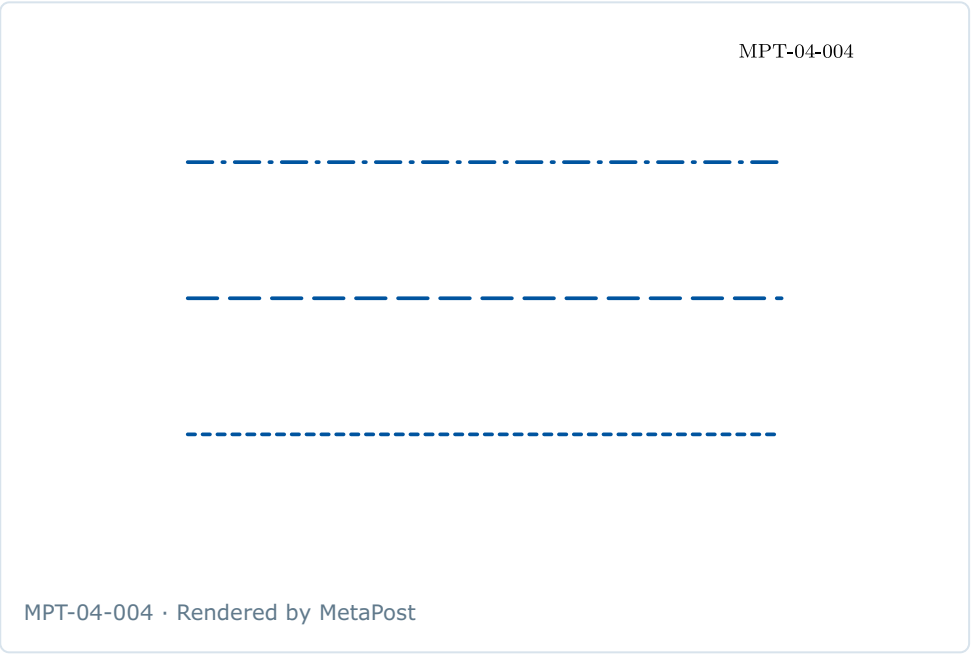
Define dash patterns

Dash lengths are explicit geometry. Compare the default pattern, long dashes and a dash-dot pattern.

MetaPost drawing

```
draw (40,55)--(280,55) dashed evenly withpen pencircle scaled 1.5 withcolor Blue;
draw (40,110)--(280,110) dashed dashpattern(on 12 off 5)
 withpen pencircle scaled 1.5 withcolor Blue;
draw (40,165)--(280,165) dashed dashpattern(on 10 off 4 on 1 off 4)
 withpen pencircle scaled 1.5 withcolor Blue;
```

Actual output



MPT-04-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`evenly`	Use the standard evenly spaced pattern.
`dashpattern`	Alternate specified on and off lengths.

MPT-04-005

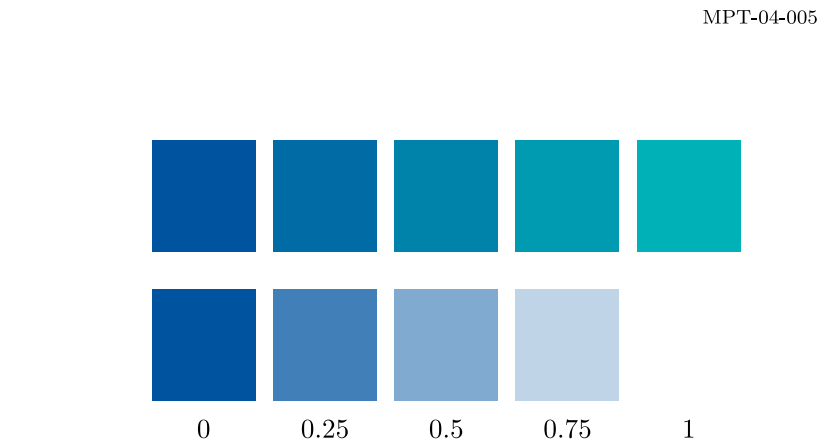
Interpolate colours and tints

The upper row mixes blue and turquoise; the lower row mixes blue and white. These are opaque RGB mixtures, not alpha transparency.

MetaPost drawing

```
for i=0 upto 4:
 fill unitsquare xscaled 42 yscaled 45 shifted (40+49i,115)
  withcolor (i/4)[Blue,Teal];
 fill unitsquare xscaled 42 yscaled 45 shifted (40+49i,55)
  withcolor (i/4)[Blue,white];
 label.bot(decimal(i/4),(61+49i,50));
endfor
```

Actual output



MPT-04-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`t[colourA,colourB]`	Linearly interpolate the colour components.
`White mixture`	Create a tint that remains an opaque fill.

MPT-04-006

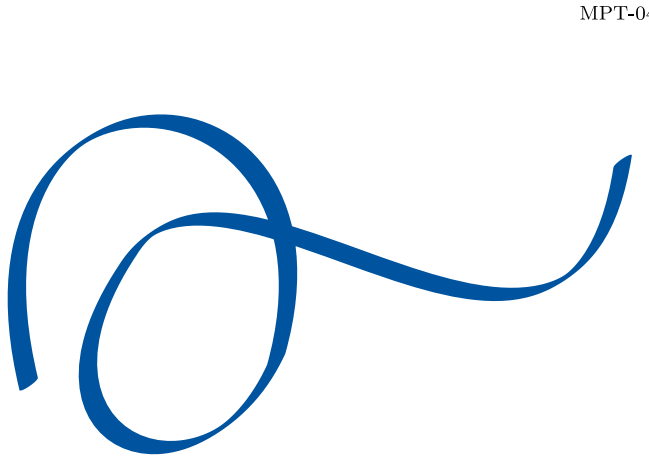
Draw a broad-nib flourish

A compound cubic path becomes a calligraphic mark when swept by a thin, wide nib.

MetaPost drawing

```
path p;
p:=(40,60)..controls (5,205) and (175,195)..(140,70)
 ..controls (110,5) and (30,35)..(80,110)
 ..controls (120,175) and (260,20)..(280,150);
draw p withpen pencircle xscaled 9 yscaled 1.8 rotated 35 withcolor Blue;
```

Actual output



MPT-04-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Compound cubic path`	Join several explicit Bezier segments.
`Elliptical pen`	Keep one nib orientation while the path turns.

05 · Transformations and reusable pictures

MPT-05-001

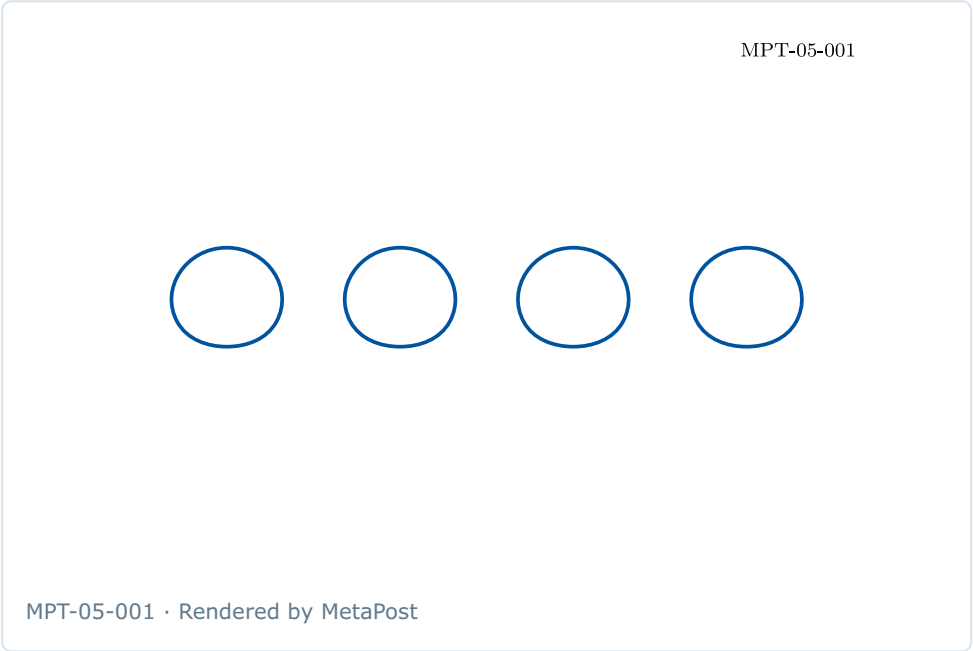
Translate a stored motif

One stored path can be reused at different positions. Its local coordinates remain centred on the origin.

MetaPost drawing

```
path p; p:=(-20,0)..(0,30)..(20,0)..(0,-10)..cycle;
for i=0 upto 3:
 draw p shifted (55+70i,100) withpen pencircle scaled 1.5 withcolor Blue;
endfor
```

Actual output



MPT-05-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`shifted (dx,dy)`	Translate a path by a pair.
`Stored path`	Reuse the same geometry rather than writing four copies.

MPT-05-002

Rotate copies

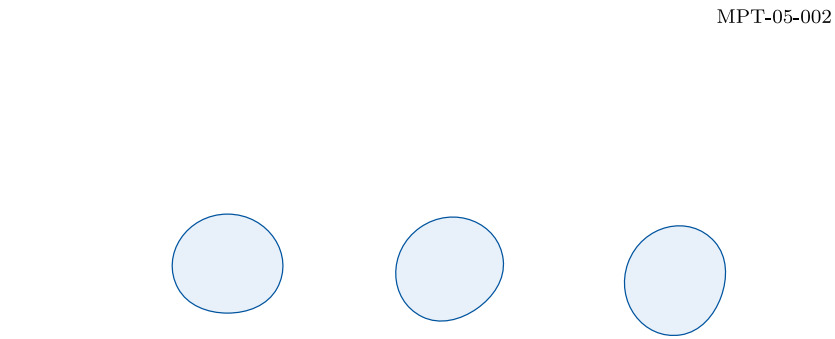
A rotation acts around the local origin. Translate after rotation to put each copy in its own panel.

Compare with MPT-05-001 · Translate a stored motif.

MetaPost drawing

```
path p; p:=(-20,0)..(0,30)..(20,0)..(0,-10)..cycle;
for i=0 upto 2:
 fill p rotated (35i) shifted (65+95i,100) withcolor Wash;
 draw p rotated (35i) shifted (65+95i,100) withcolor Blue;
endfor
```

Actual output



MPT-05-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`rotated angle`	Angles are in degrees.
`Order`	Rotate locally, then translate to the desired centre.

MPT-05-003

Scale each coordinate independently

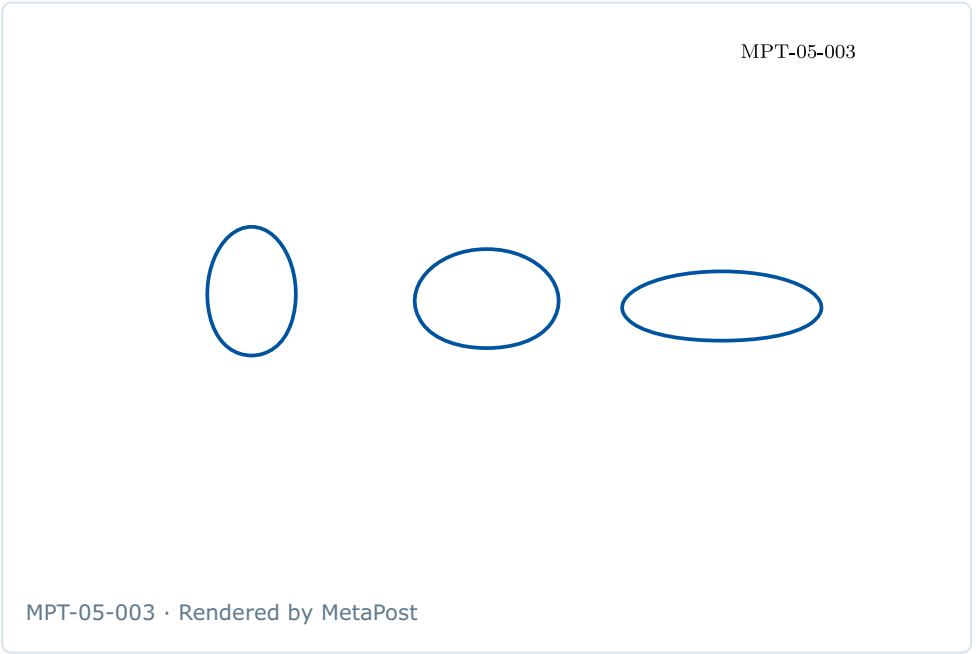
Anisotropic scaling changes proportions. Each copy uses its own horizontal and vertical factors.

Compare with MPT-05-001 · Translate a stored motif.

MetaPost drawing

```
path p; p:=(-20,0)..(0,30)..(20,0)..(0,-10)..cycle;
for i=0 upto 2:
 draw p xscaled (.8+.5i) yscaled (1.3-.3i) shifted (65+95i,100)
  withpen pencircle scaled 1.5 withcolor Blue;
endfor
```

Actual output



MPT-05-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`xscaled / yscaled`	Apply different factors to horizontal and vertical components.
`Pen placement`	withpen chooses the final stroke independently of the transformed path.

MPT-05-004

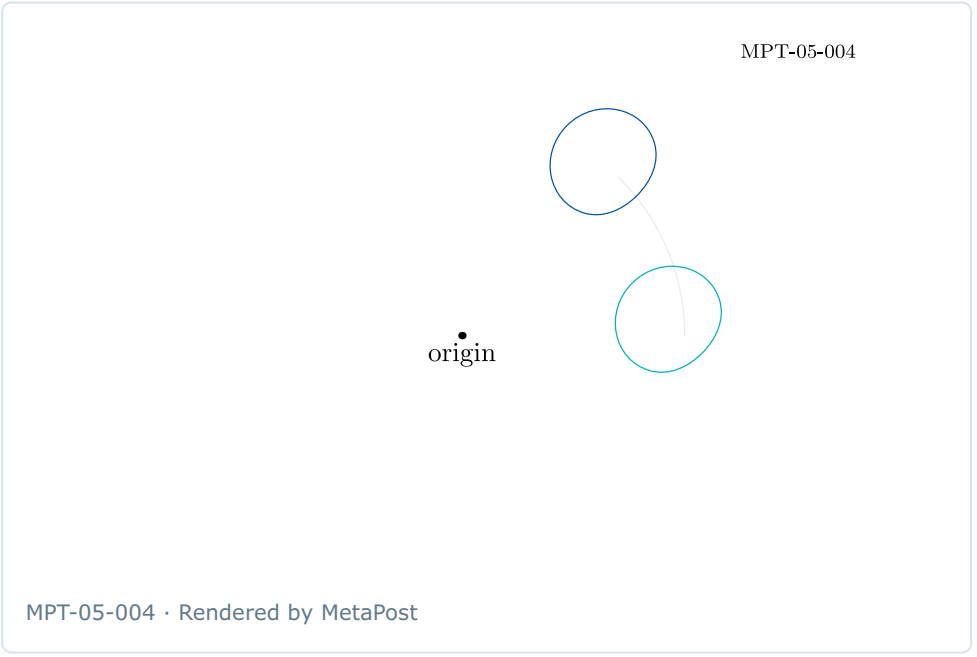
See why transformation order matters

The blue path is translated and then rotated; the turquoise one is rotated and then translated. Their final positions differ even though the same operations appear.

MetaPost drawing

```
path p; p:=(-20,0)..(0,30)..(20,0)..(0,-10)..cycle;
pair o; o:=(150,95);
draw subpath (0,1) of fullcircle scaled 180 shifted o withcolor Light;
draw p shifted (90,0) rotated 45 shifted o withcolor Blue;
draw p rotated 45 shifted (90,0) shifted o withcolor Teal;
dotlabel.bot("origin",o);
```

Actual output



MPT-05-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Operation order`	Transformations are applied from left to right.
`Rotation about the origin`	A preceding translation is rotated along with the path.

MPT-05-005

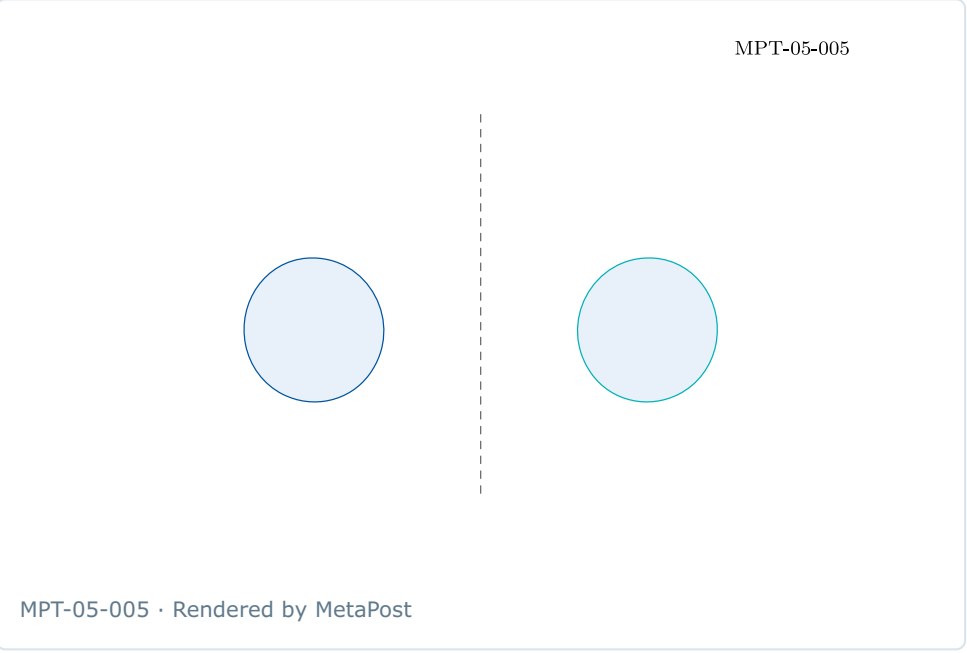
Reflect an asymmetric shape

A negative horizontal scale mirrors the motif. An asymmetric source makes the reflection visible.

MetaPost drawing

```
path p; p:=(0,0)..(35,8)..(10,55)..(-15,20)..cycle;
fill p shifted (80,70) withcolor Wash;
draw p shifted (80,70) withcolor Blue;
fill p xscaled -1 shifted (240,70) withcolor Wash;
draw p xscaled -1 shifted (240,70) withcolor Teal;
draw (160,30)--(160,185) dashed evenly withcolor Grey;
```

Actual output



MPT-05-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`xscaled -1`	Reflect across the local vertical axis.
`Asymmetry`	A symmetric shape could conceal the effect.

MPT-05-006

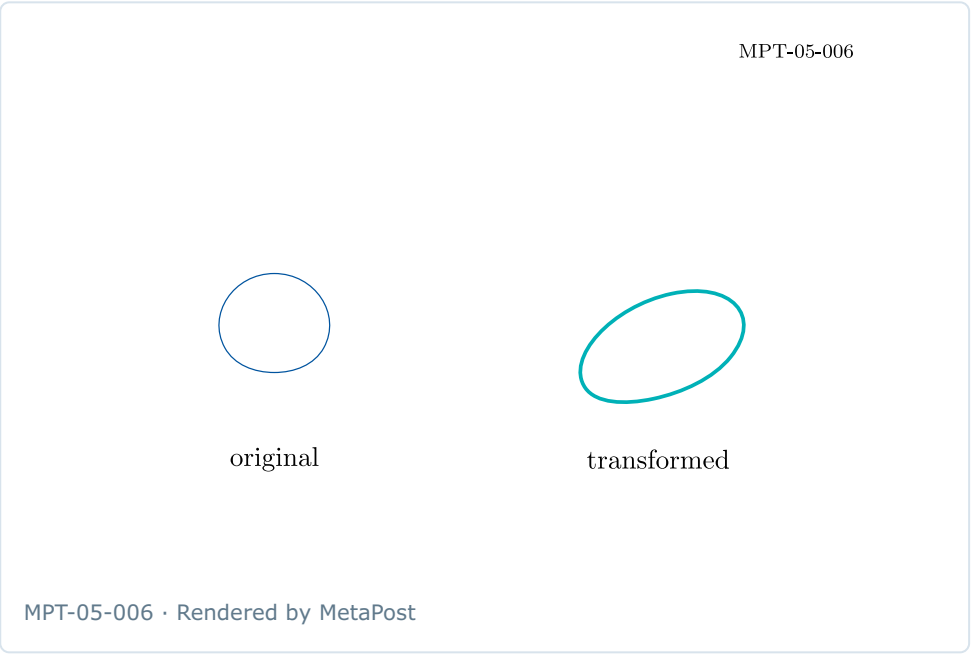
Name and reuse an affine transform

A transform packages scaling, shearing, rotation and translation into one reusable value.

MetaPost drawing

```
path p; p:=(-20,0)..(0,30)..(20,0)..(0,-10)..cycle;
transform T;
T:=identity xscaled 1.5 slanted .4 rotated 12 shifted (230,80);
draw p shifted (75,90) withcolor Blue;
draw p transformed T withpen pencircle scaled 1.5 withcolor Teal;
label("original",(75,45)); label("transformed",(230,45));
```

Actual output



MPT-05-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`identity`	Start from an unchanged coordinate system.
`slanted`	Apply a horizontal shear.
`transformed T`	Apply the named transformation in one operation.

06 · Clipping, layers and cut-outs

MPT-06-001

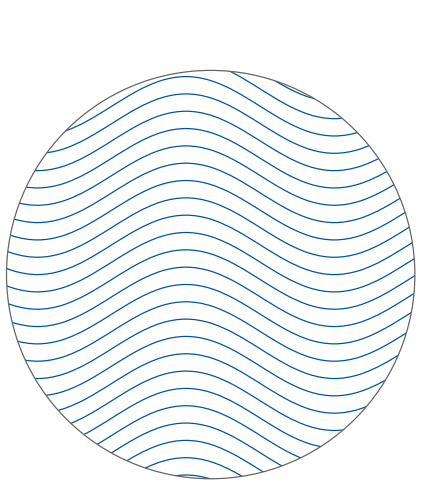
Clip a wave field to a circle

Create the wave field once, then clip its picture. Clipping changes visibility without rewriting the constituent curves.

MetaPost drawing

```
picture waves;
waves:=image(
 for j=-30 step 7 until 230:
  draw for i=0 step 10 until 320:
   if i>0: .. fi (i,j+12sind(3i))
  endfor withcolor Blue;
 endfor
);
path boundary; boundary:=fullcircle scaled 165 shifted (160,105);
clip waves to boundary;
draw waves; draw boundary withcolor Grey;
```

Actual output



MPT-06-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`image(...)`	Capture a group of drawing operations as a picture.
`clip picture to path`	Keep only the portion inside a closed boundary.

MPT-06-002

Reuse the field with another mask

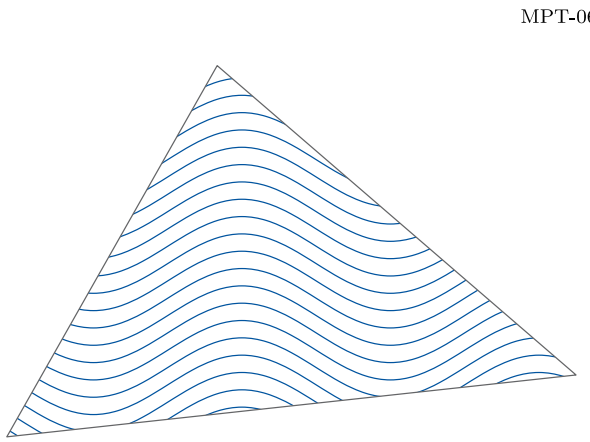
Keep the same wave-generation code and replace only the clipping path.

Compare with MPT-06-001 · Clip a wave field to a circle.

MetaPost drawing

```
picture waves;
waves:=image(
 for j=-30 step 7 until 230:
  draw for i=0 step 10 until 320:
   if i>0: .. fi (i,j+12sind(3i))
  endfor withcolor Blue;
 endfor
);
path boundary; boundary:=(55,40)--(285,65)--(140,190)--cycle;
clip waves to boundary;
draw waves; draw boundary withcolor Grey;
```

Actual output



MPT-06-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Different boundary`	The drawing procedure is independent of the mask.
`Closed triangle`	Any suitable closed path can act as a clipping boundary.

MPT-06-003

Build an emblem with opaque layers

Drawing order makes a compact emblem. The central cut-out is white overpainting on the white canvas, not a transparent hole.

MetaPost drawing

```
fill fullcircle scaled 130 shifted (125,105) withcolor Blue;
fill fullcircle scaled 130 shifted (195,105) withcolor Teal;
fill fullcircle scaled 78 shifted (160,105) withcolor white;
draw fullcircle scaled 42 shifted (160,105)
 withpen pencircle scaled 4 withcolor Orange;
```

Actual output



MPT-06-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Later fills`	Paint over earlier objects.
`White cut-out`	Matches this opaque canvas; it does not create transparency.

MPT-06-004

Make an annular clipping region

Opposite traversal directions make an annular mask. The hole reveals the pale rectangle behind the clipped stripes.

MetaPost drawing

```
picture stripes; path ring;
fill unitsquare xscaled 230 yscaled 165 shifted (45,25) withcolor Wash;
stripes:=image(
 for x=30 step 9 until 290:
  draw (x,20)--(x+50,190) withpen pencircle scaled 3 withcolor Teal;
 endfor
);
ring:=((subpath (0,8) of fullcircle scaled 160)
 --reverse (subpath (0,8) of fullcircle scaled 85)--cycle) shifted (160,105);
clip stripes to ring; draw stripes;
```

Actual output



MPT-06-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`reverse inner contour`	Give the inner boundary the opposite winding.
`subpath (0,8)`	Use an open traversal of the complete circle before joining the boundaries.

MPT-06-005

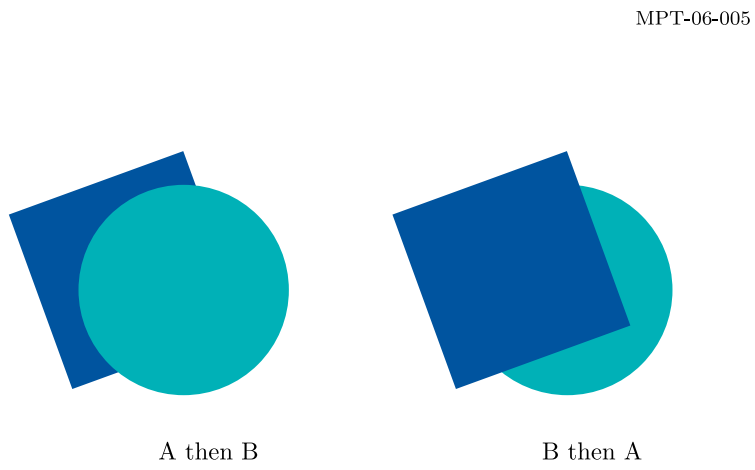
Change the stacking order of pictures

The same two coloured objects appear in both panels. Swapping draw order changes which object covers the overlap.

MetaPost drawing

```
picture a,b;
a:=image(fill unitsquare scaled 75 rotated 20 withcolor Blue;);
b:=image(fill fullcircle scaled 85 shifted (45,40) withcolor Teal;);
draw a shifted (35,60); draw b shifted (35,60);
draw b shifted (190,60); draw a shifted (190,60);
label(btex A then B etex,(90,35)); label(btex B then A etex,(245,35));
```

Actual output



MPT-06-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`picture`	Store several drawings or fills as one reusable object.
`draw order`	The second picture covers the first in the overlapping area.

MPT-06-006

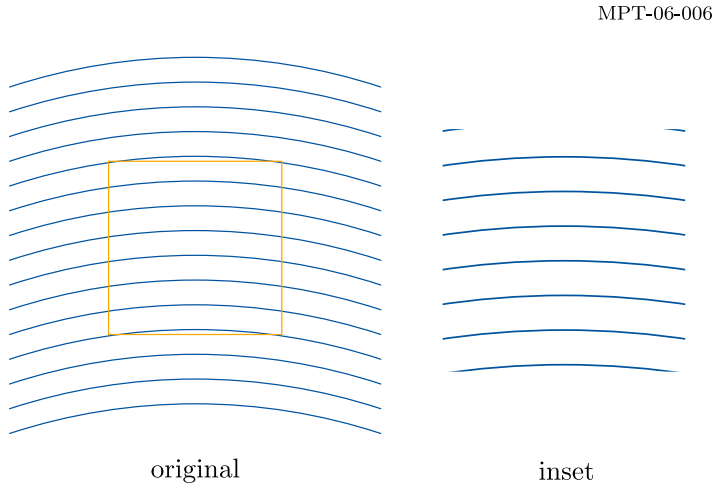
Crop a copy for a magnified inset

Copy a picture before clipping it. The original view remains intact while a selected region is magnified.

MetaPost drawing

```
picture full,crop;
full:=image(
 for j=0 step 10 until 140:
  draw (0,j)..(75,j+12)..(150,j) withcolor Blue;
 endfor
);
crop:=full;
clip crop to unitsquare scaled 70 shifted (40,40);
draw full shifted (25,40);
draw crop shifted (-40,-40) scaled 1.4 shifted (200,65);
draw unitsquare scaled 70 shifted (65,80) withcolor Orange;
label("original",(100,25)); label("inset",(250,25));
```

Actual output



MPT-06-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`crop:=full`	Copy a picture before changing its clipping state.
`Translate before scaling`	Move the crop to the origin, enlarge it, then place the inset.

07 · Loops, macros and generative patterns

MPT-07-001

Repeat rotated quarter-circle tiles

A small motif becomes a Truchet-style field. Rotations are around each tile centre, so neighbouring arcs meet at edge midpoints.

MetaPost drawing

```
picture tile;
tile:=image(
 draw quartercircle scaled 36;
 draw quartercircle scaled 36 rotated 180 shifted (36,36);
);
for i=0 upto 5:
 for j=0 upto 3:
  draw tile shifted (-18,-18) rotated (90*((i+j+i*j) mod 2))
   shifted (58+36i,53+36j) withcolor ((i mod 2)[Blue,Teal]);
 endfor
endfor
```

Actual output



MPT-07-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`quartercircle`	Use a circular arc as a tile motif.
`mod 2`	Choose one of two orientations deterministically.
`Centre before rotation`	Keep each tile within its cell.

MPT-07-002

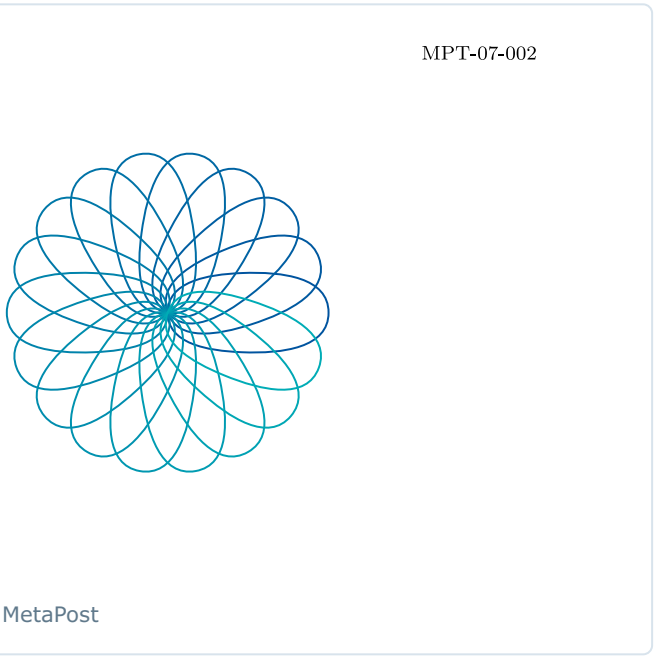
Generate a rosette from one petal

Repeat a closed petal at equally spaced angles. The negative space emerges from the overlaps of identical curves.

MetaPost drawing

```
path petal;
petal:=(0,0)..(30,16)..(65,0)..(30,-16)..cycle;
for a=0 step 20 until 340:
 draw petal rotated a shifted (160,105)
  withpen pencircle scaled .8 withcolor (a/360)[Blue,Teal];
endfor
```

Actual output



MPT-07-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`rotated a`	A full turn divided into eighteen copies.
`One path`	All petals share exactly the same local construction.

MPT-07-003

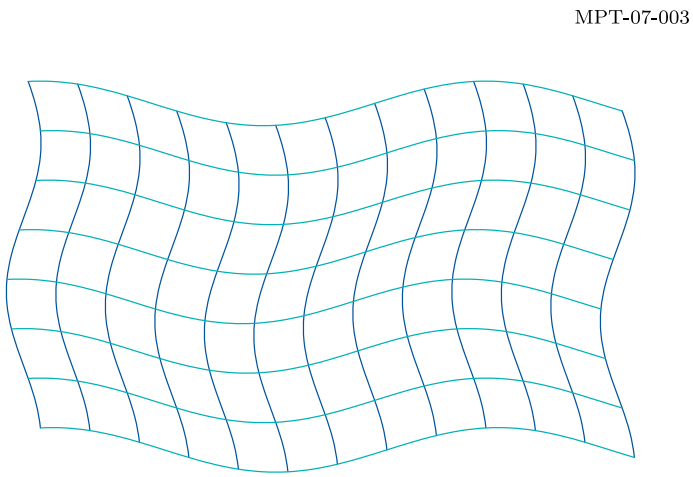
Warp a connected mesh

A small function displaces lattice points. Smooth paths connect the transformed positions while preserving the original grid connectivity.

MetaPost drawing

```
vardef warp(expr x,y)=(x+7sind(3y),y+9sind(2x)) enddef;
for i=0 upto 12:
 draw for j=0 upto 14:
  if j>0: .. fi warp(40+20i,35+10j)
 endfor withcolor Blue;
endfor
for j=0 upto 7:
 draw for i=0 upto 24:
  if i>0: .. fi warp(40+10i,35+20j)
 endfor withcolor Teal;
endfor
```

Actual output



MPT-07-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`vardef`	Define a function returning a pair.
`sind`	Use degree-based trigonometry for the displacement.
`Connectivity`	Transform vertices while retaining the same row and column relationships.

MPT-07-004

Reveal a moire field

Two offset families of circles create an interference-like pattern. Fine strokes keep the emergent bands visible.

MetaPost drawing

```
for r=8 step 4 until 76:
 draw fullcircle scaled (2r) shifted (140,105)
  withpen pencircle scaled .35 withcolor Blue;
 draw fullcircle scaled (2r) shifted (180,105)
  withpen pencircle scaled .35 withcolor Teal;
endfor
```

Actual output



MPT-07-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Two centres`	Offset identical families by forty drawing units.
`Thin pens`	Avoid filling the small spaces that form the visible bands.

MPT-07-005

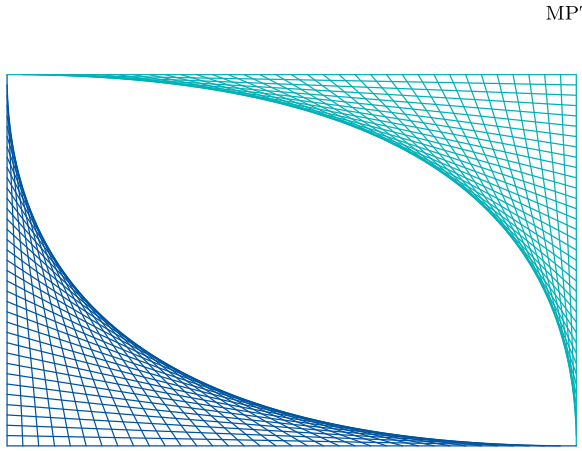
Make curved envelopes from straight segments

The apparent curves are envelopes of straight lines. Every endpoint comes from one interpolation parameter.

MetaPost drawing

```
for k=0 upto 36:
 t:=k/36;
 draw (45+230t,35)--(45,185-150t) withpen pencircle scaled .5 withcolor Blue;
 draw (275-230t,185)--(275,35+150t) withpen pencircle scaled .5 withcolor Teal;
endfor
```

Actual output



MPT-07-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`One parameter`	Move two endpoints at the same rate along different edges.
`Straight segments`	The curved appearance comes from the family, not a spline primitive.

MPT-07-006

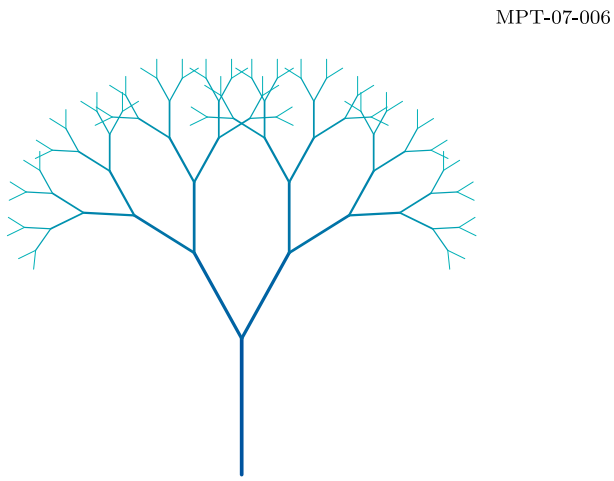
Grow a bounded recursive tree

A branch creates two smaller rotated branches. Depth is explicit and finite; stroke width decreases toward the tips.

MetaPost drawing

```
def branch(expr base,ang,len,depth)=
 begingroup
 save tip; pair tip; tip:=base+len*dir ang;
 draw base--tip withpen pencircle scaled (.2depth+.3)
  withcolor (depth/6)[Teal,Blue];
 if depth>0:
  branch(tip,ang+29,.72len,depth-1);
  branch(tip,ang-29,.72len,depth-1);
 fi
 endgroup
enddef;
branch((160,25),90,55,6);
```

Actual output



MPT-07-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`begingroup / save`	Give each recursive call its own temporary point.
`depth-1`	Guarantee termination.
`Scaled length`	The geometric decay controls the overall size of the tree.

08 · Labels, mathematics and diagrams

MPT-08-001

Position labels around an anchor

Suffixes choose a label side. The central point is shared, while the text is placed relative to its bounds.

MetaPost drawing

```
fill fullcircle scaled 5 shifted (160,110) withcolor Orange;
label.top("above",(160,110));
label.bot("below",(160,110));
label.lft("left",(160,110));
label.rt("right",(160,110));
```

Actual output



MPT-08-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`label suffix`	Select top, bottom, left or right placement.
`Shared anchor`	Text alignment can change without moving the anchor.

MPT-08-002

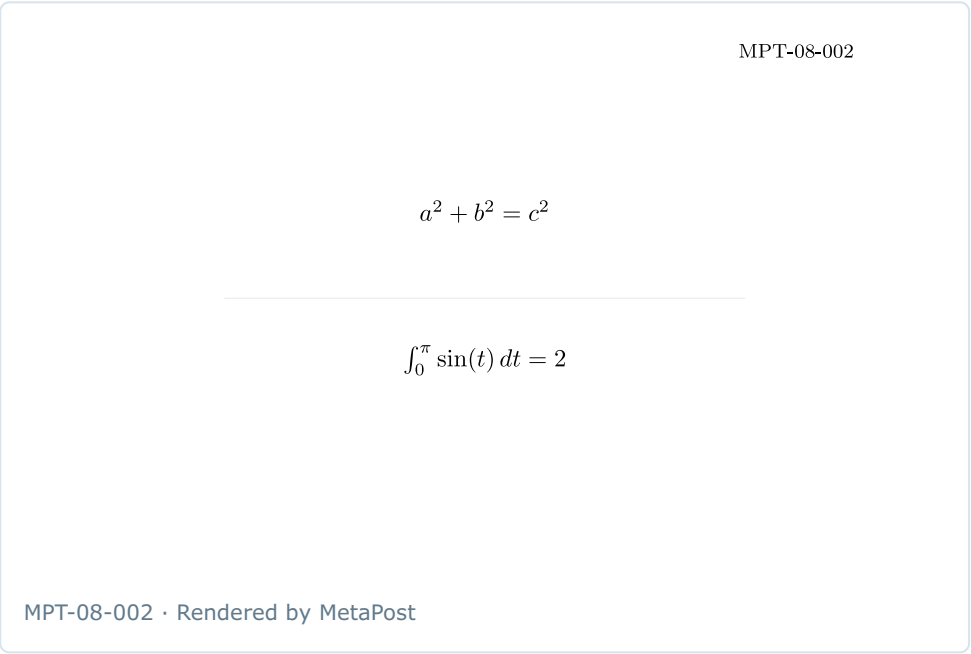
Typeset mathematics with TeX

btex delegates mathematical typesetting to TeX. The SVG contains outlined glyphs, so readers do not need those fonts installed.

MetaPost drawing

```
label(btex $a^2+b^2=c^2$ etex,(160,145));
label(btex $\int_0^\pi \sin(t)\,dt=2$ etex,(160,85));
draw (55,110)--(265,110) withcolor Light;
```

Actual output



MetaPost · exit 0

Command or parameter	Meaning
`btex ... etex`	Create a TeX typeset picture.
`Outlined glyphs`	prologues=3 preserves the native letterforms in the SVG.

MPT-08-003

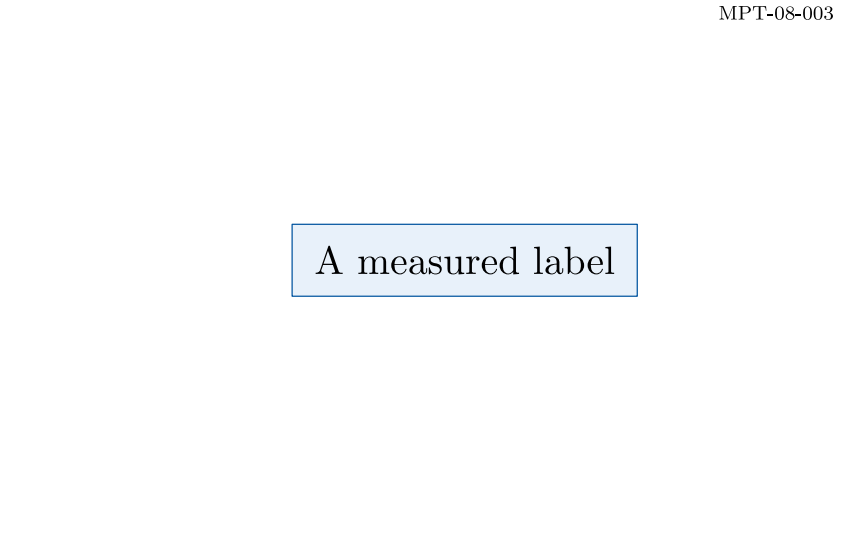
Measure and frame a label

The box is derived from the actual text picture. Changing the words changes the frame automatically.

MetaPost drawing

```
picture caption;
caption:=btex A measured label etex scaled 1.6;
caption:=caption shifted ((160,110)-.5[llcorner caption,urcorner caption]);
bboxmargin:=9;
fill bbox caption withcolor Wash;
draw bbox caption withcolor Blue;
draw caption;
```

Actual output



MetaPost · exit 0

Command or parameter	Meaning
`infont`	Turn text into a picture with the selected font.
`llcorner / urcorner`	Read the actual picture bounds.
`bboxmargin`	Add a consistent margin around its bounding box.

MPT-08-004

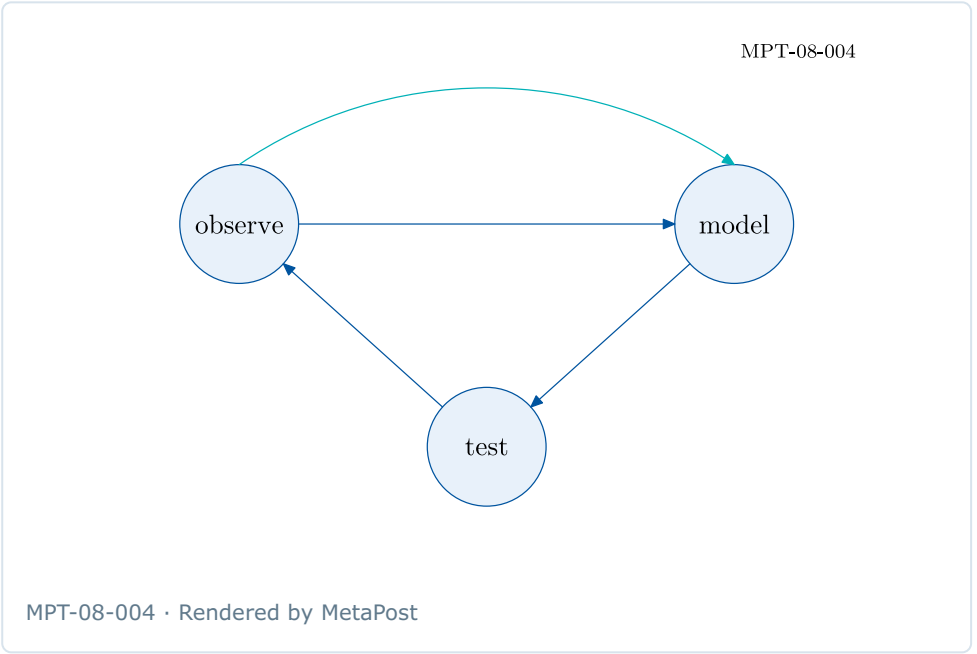
Connect nodes at their boundaries

The arrows are cut against circular node paths, so their endpoints follow the geometry rather than guessed offsets.

MetaPost drawing

```
pair a,b,c; path na,nb,nc;
a:=(60,140); b:=(260,140); c:=(160,50);
na:=fullcircle scaled 48 shifted a;
nb:=fullcircle scaled 48 shifted b;
nc:=fullcircle scaled 48 shifted c;
for p=na,nb,nc: fill p withcolor Wash; draw p withcolor Blue; endfor
ahlength:=5;
drawarrow ((a--b) cutbefore na cutafter nb) withcolor Blue;
drawarrow ((b--c) cutbefore nb cutafter nc) withcolor Blue;
drawarrow ((c--a) cutbefore nc cutafter na) withcolor Blue;
drawarrow (60,164)..(160,195)..(260,164) withcolor Teal;
label("observe",a); label("model",b); label("test",c);
```

Actual output



MPT-08-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`cutbefore / cutafter`	Trim an open path where it meets the node boundaries.
`ahlength`	Choose arrowhead length.
`Node paths`	Reuse the boundaries for both drawing and intersection calculations.

MPT-08-005

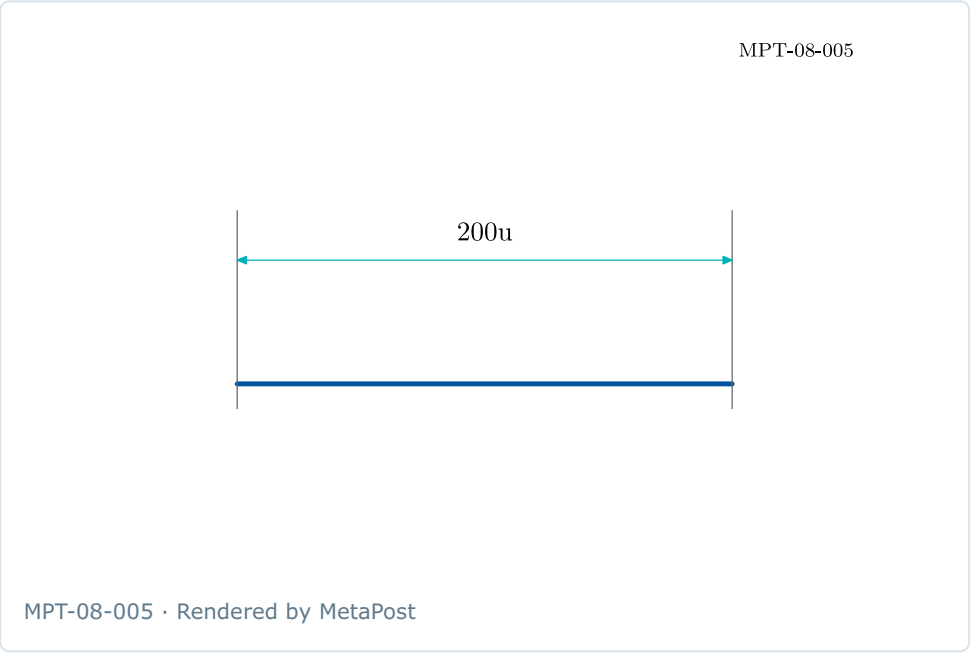
Annotate a measured dimension

The dimension label is calculated from the same endpoint pair as the drawing.

MetaPost drawing

```
pair a,b;
a:=(60,75); b:=(260,75);
draw a--b withpen pencircle scaled 2 withcolor Blue;
draw (a-(0,10))--(a+(0,70)) withcolor Grey;
draw (b-(0,10))--(b+(0,70)) withcolor Grey;
drawdblarrow (a+(0,50))--(b+(0,50)) withcolor Teal;
label.top(decimal(abs(b-a))&"u",.5[a,b]+(0,55));
```

Actual output



MPT-08-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`drawdblarrow`	Put an arrowhead at both ends of a dimension line.
`abs(b-a)`	Calculate the displayed length from the original geometry.

MPT-08-006

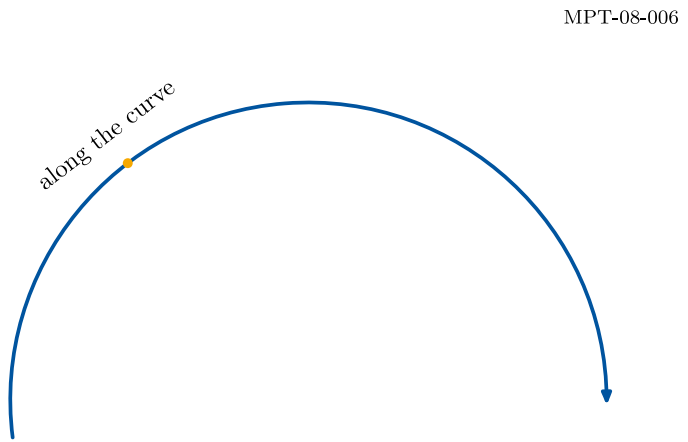
Rotate a label along a curve

Choose a location by arc distance and rotate its caption to the local tangent. A normal offset separates the text from the stroke.

MetaPost drawing

```
path p; numeric t; pair anchor,tangent;
p:=(40,40)..(150,175)..(280,55);
t:=arctime (.32arclength p) of p;
anchor:=point t of p; tangent:=unitvector(direction t of p);
drawarrow p withpen pencircle scaled 1.5 withcolor Blue;
label(btex along the curve etex rotated (angle tangent),
 anchor+14*(tangent rotated 90));
fill fullcircle scaled 4 shifted anchor withcolor Orange;
```

Actual output



MPT-08-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`angle tangent`	Obtain the tangent angle in degrees.
`Normal offset`	Rotate the unit tangent by ninety degrees to place the caption beside the path.

09 · Functions, envelopes and parametric curves

MPT-09-001

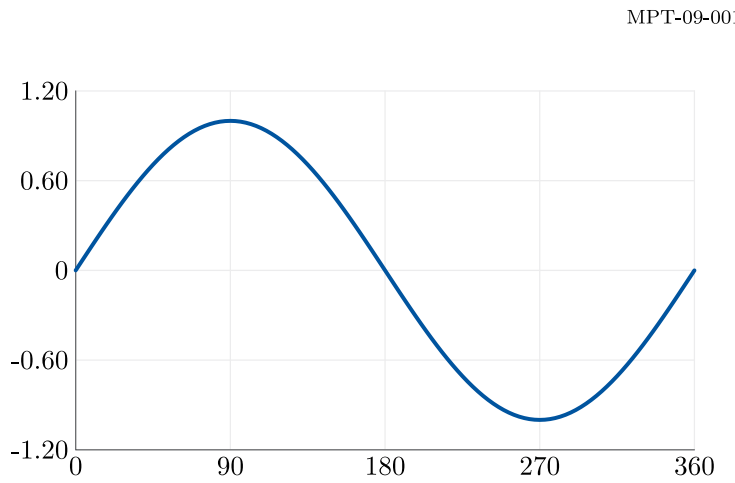
Sample a sine function

A function plot is a sampled path in an explicit coordinate map. The input angles are degrees because sind is degree-based.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=360; ymin:=-1.2; ymax:=1.2;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
draw for a=0 step 3 until 360:
 if a>0: -- fi P(a,sind a)
endfor withpen pencircle scaled 1.6 withcolor Blue;
```

Actual output



MPT-09-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`P(x,y)`	Map data values onto the plot rectangle.
`Sampling step`	Three-degree samples approximate the analytic sine curve.
`sind`	Evaluate sine in degrees.

MPT-09-002

Compare two functions

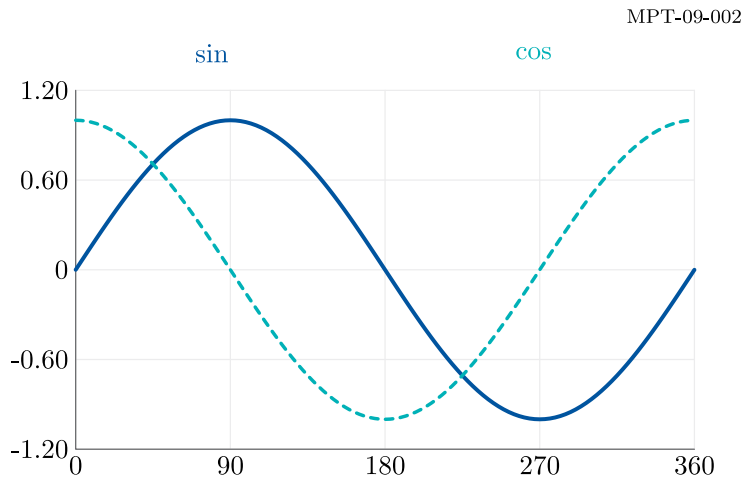
Add a cosine curve to the same data ranges. A second colour and dash pattern distinguish the two functions.

Compare with MPT-09-001 · Sample a sine function.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=360; ymin:=-1.2; ymax:=1.2;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
draw for a=0 step 3 until 360:
 if a>0: -- fi P(a,sind a)
endfor withpen pencircle scaled 1.6 withcolor Blue;
draw for a=0 step 3 until 360:
 if a>0: -- fi P(a,cosd a)
endfor dashed evenly withpen pencircle scaled 1.4 withcolor Teal;
label("sin",(95,195)) withcolor Blue;
label("cos",(225,195)) withcolor Teal;
```

Actual output



MPT-09-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Shared ranges`	Make both curves use the same coordinate map.
`cosd`	The cosine counterpart of sind.
`Colour and dashes`	Use redundant visual distinctions for the two series.

MPT-09-003

Compare knots with a smooth interpolation

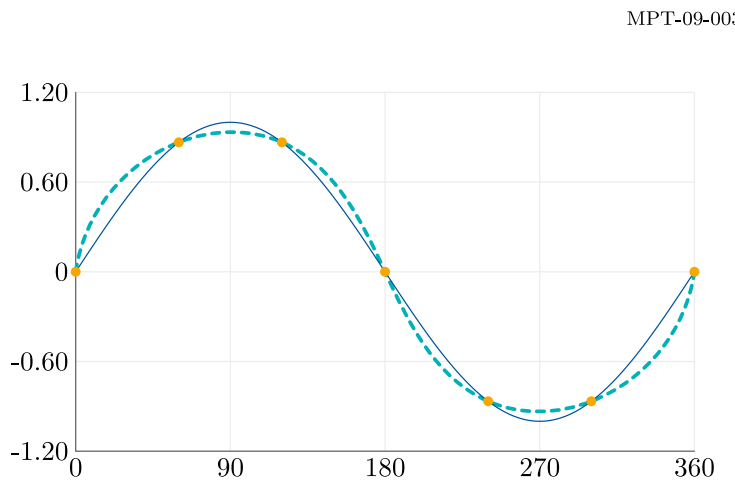
Six coarse intervals are interpolated with MetaPost smooth joins. The dense blue curve is the underlying sine function; the two need not coincide.

Compare with MPT-09-001 · Sample a sine function.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=360; ymin:=-1.2; ymax:=1.2;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
draw for a=0 step 60 until 360:
 if a>0: .. fi P(a,sind a)
endfor dashed evenly withpen pencircle scaled 1.5 withcolor Teal;
draw for a=0 step 3 until 360:
 if a>0: -- fi P(a,sind a)
endfor withcolor Blue;
for a=0 step 60 until 360:
 fill fullcircle scaled 4 shifted P(a,sind a) withcolor Orange;
endfor
```

Actual output



MPT-09-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`.. between samples`	Interpolate the knots with MetaPost cubic curves.
`Dense analytic samples`	Provide a reference rather than claiming interpolation recovers the original function.

MPT-09-004

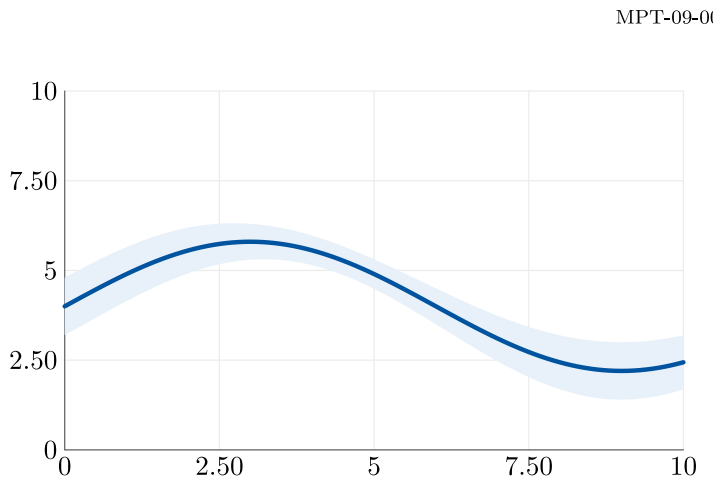
Fill between two envelopes

This synthetic band is an illustrative pair of bounds, not a statistical confidence interval. Construct its upper edge forward and its lower edge backwards.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=10; ymin:=0; ymax:=10;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
vardef m(expr x)=4+1.8sind(30x) enddef;
vardef w(expr x)=.6+.2cosd(40x) enddef;
path upper,lower;
upper:=for i=0 upto 100: if i>0: -- fi P(i/10,m(i/10)+w(i/10)) endfor;
lower:=for i=100 downto 0: if i<100: -- fi P(i/10,m(i/10)-w(i/10)) endfor;
fill upper--lower--cycle withcolor Wash;
draw for i=0 upto 100: if i>0: -- fi P(i/10,m(i/10)) endfor
 withpen pencircle scaled 1.8 withcolor Blue;
```

Actual output



MPT-09-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Reversed lower edge`	Close a polygon around the band without crossing diagonally through it.
`m / w`	Keep the centre and half-width as independent functions.

MPT-09-005

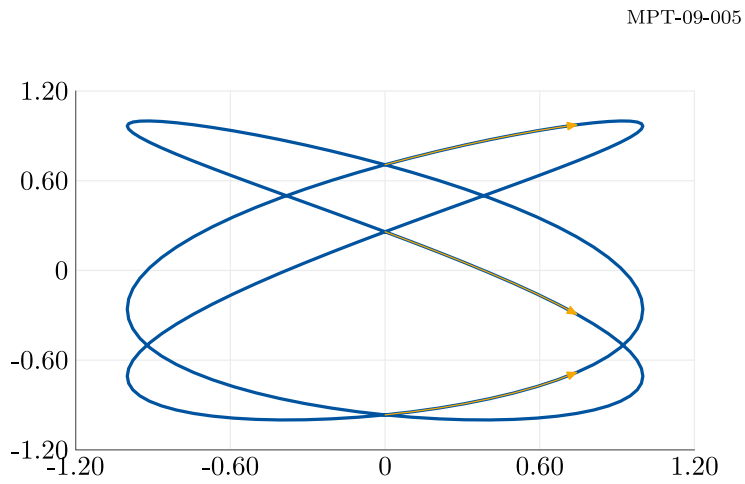
Draw a parametric Lissajous curve

Both coordinates depend on one parameter. Arrowheads on selected subpaths indicate traversal direction.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=-1.2; xmax:=1.2; ymin:=-1.2; ymax:=1.2;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
path p;
p:=for a=0 step 2 until 358:
 if a>0: -- fi P(sind(3a),sind(2a+45))
endfor--cycle;
draw p withpen pencircle scaled 1.3 withcolor Blue;
for k=0 step 60 until 120:
 drawarrow subpath (k,k+8) of p withcolor Orange;
endfor
```

Actual output



MPT-09-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Parametric coordinates`	x and y are separate periodic functions of a.
`Phase shift`	The second coordinate is offset by forty-five degrees.

MPT-09-006

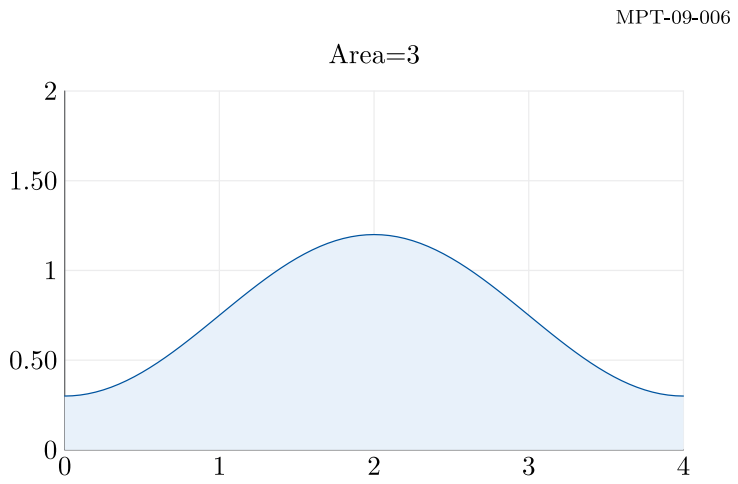
Fill an area and estimate its integral

The shaded area belongs to a positive synthetic function. Eighty trapezoids estimate its integral; the analytic value over this interval is three.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=4; ymin:=0; ymax:=2;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
vardef f(expr x)=.3+.9*(sind(45x))*(sind(45x)) enddef;
path p; numeric area;
p:=P(0,0)--for i=0 upto 80: if i>0: -- fi P(i/20,f(i/20)) endfor
 --P(4,0)--cycle;
fill p withcolor Wash;
draw for i=0 upto 80: if i>0: -- fi P(i/20,f(i/20)) endfor withcolor Blue;
area:=0;
for i=1 upto 80: area:=area+(f((i-1)/20)+f(i/20))/40; endfor
label("Area="&numlabel(area),(165,195));
message "check: area="&decimal area;
```

Actual output



MPT-09-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Closed area path`	Join the sampled function to its zero baseline.
`Trapezoidal rule`	Add interval width times the mean of each pair of endpoint heights.

10 · Data and statistical graphics

MPT-10-001

Build a labelled bar chart

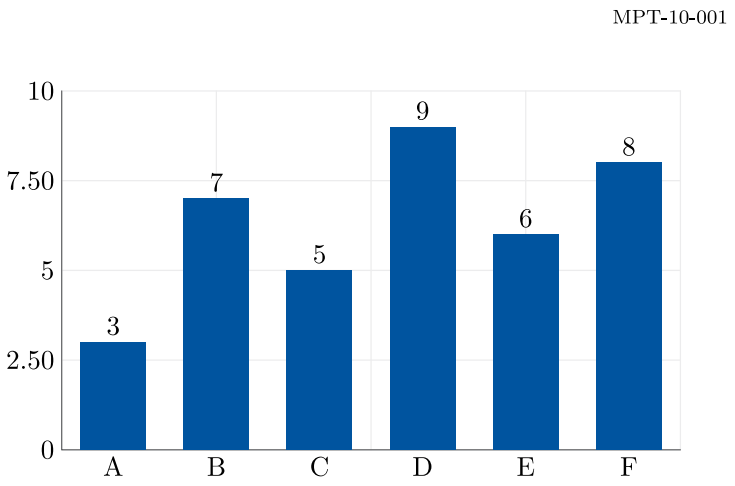
Six inline values determine the rectangles and their labels. Category names are independent of the numeric axis.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0.5; xmax:=6.5; ymin:=0; ymax:=10;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;

 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
numeric v[];
v[1]:=3;
v[2]:=7;
v[3]:=5;
v[4]:=9;
v[5]:=6;
v[6]:=8;
for i=1 upto 6:
 fill P(i-.32,0)--P(i+.32,0)--P(i+.32,v[i])--P(i-.32,v[i])--cycle
  withcolor Blue;
 label.bot(substring (i-1,i) of "ABCDEF",P(i,0));
 label.top(decimal v[i],P(i,v[i]));
endfor
```

Actual output



MPT-10-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Rectangle bounds`	Compute bar width in data units before mapping coordinates.
`substring`	Choose the category letter from a short string.

MPT-10-002

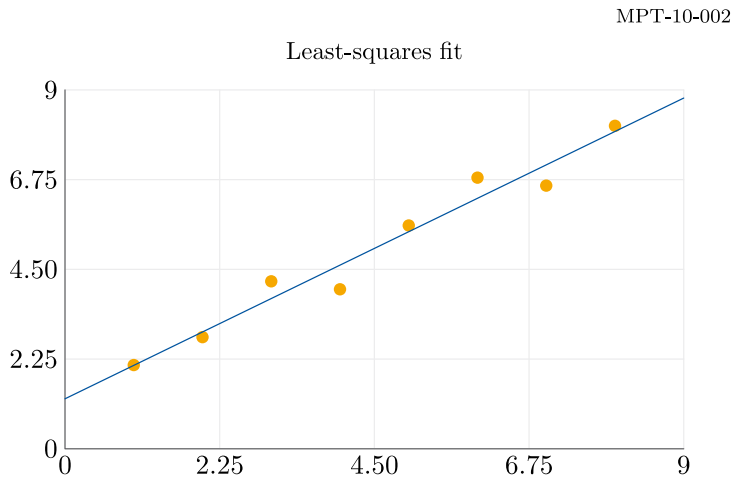
Fit a line to a scatter plot

The least-squares coefficients are calculated from these eight observations. The plotted trend is not a hand-picked line.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=9; ymin:=0; ymax:=9;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
numeric v[];
v[1]:=2.1;
v[2]:=2.8;
v[3]:=4.2;
v[4]:=4.0;
v[5]:=5.6;
v[6]:=6.8;
v[7]:=6.6;
v[8]:=8.1;
numeric N,sx,sy,sxx,sxy,slope,intercept;
N:=8; sx:=sy:=sxx:=sxy:=0;
for i=1 upto N:
 sx:=sx+i; sy:=sy+v[i]; sxx:=sxx+i*i; sxy:=sxy+i*v[i];
 fill fullcircle scaled 5 shifted P(i,v[i]) withcolor Orange;
endfor
slope:=(N*sxy-sx*sy)/(N*sxx-sx*sx);
intercept:=(sy-slope*sx)/N;
draw P(0,intercept)--P(9,intercept+9slope) withcolor Blue;
label(btex Least-squares fit etex,(165,195));
message "check: slope="&decimal slope;
```

Actual output



MPT-10-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Summary sums`	Accumulate x, y, x squared and x times y from the observations.
`Slope / intercept`	Use the standard straight-line least-squares formulas.

MPT-10-003

Bin raw observations into a histogram

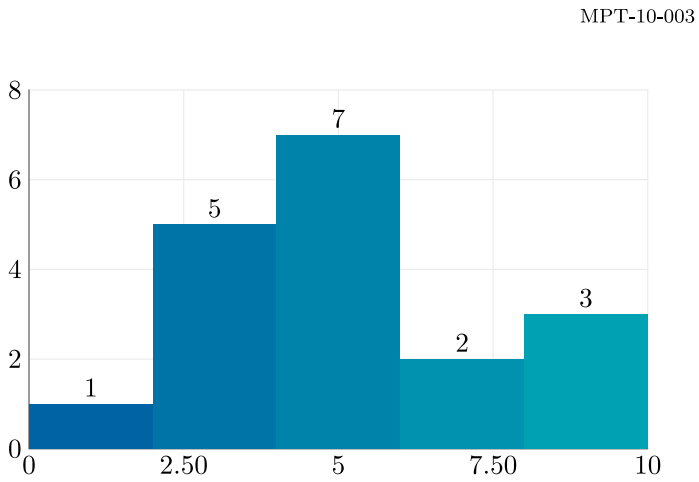
The source contains observations, not precomputed counts. Five width-two bins cover zero to ten; all eighteen observations are assigned exactly once.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=10; ymin:=0; ymax:=8;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
numeric v[];
v[1]:=1;
v[2]:=2;
v[3]:=2;
v[4]:=2;
v[5]:=3;
v[6]:=3;
v[7]:=4;
v[8]:=4;
v[9]:=4;
v[10]:=5;
v[11]:=5;
v[12]:=5;
v[13]:=5;
v[14]:=6;
v[15]:=7;
v[16]:=8;
v[17]:=8;
v[18]:=9;
numeric count[],total;
for b=1 upto 5: count[b]:=0; endfor
for i=1 upto 18: b:=floor(v[i]/2)+1; count[b]:=count[b]+1; endfor
total:=0;
for b=1 upto 5:
 total:=total+count[b];
 fill P(2b-2,0)--P(2b,0)--P(2b,count[b])--P(2b-2,count[b])--cycle
```

```
  withcolor (b/6)[Blue,Teal];
 label.top(decimal count[b],P(2b-1,count[b]));
endfor
message "check: histogram-total="&decimal total;
```

Actual output



MPT-10-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`floor(value/2)+1`	Assign these values to left-closed width-two bins.
`Counts`	The five frequencies sum to eighteen.

MPT-10-004

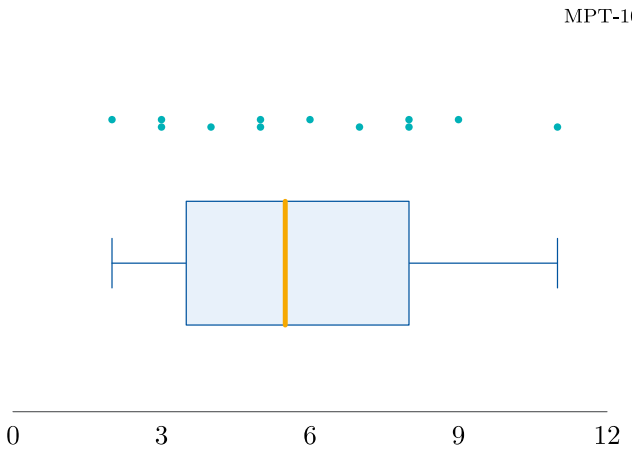
Calculate a box plot from observations

Sort the twelve observations, then use median-of-halves quartiles. Whiskers show the sample minimum and maximum, not an automatic outlier rule.

MetaPost drawing

```
numeric v[];
v[1]:=8;
v[2]:=2;
v[3]:=5;
v[4]:=11;
v[5]:=3;
v[6]:=7;
v[7]:=9;
v[8]:=4;
v[9]:=8;
v[10]:=6;
v[11]:=3;
v[12]:=5;
numeric N,temp,qone,median,qthree;
N:=12;
for pass=1 upto N-1:
 for i=1 upto N-pass:
  if v[i]>v[i+1]:
   temp:=v[i]; v[i]:=v[i+1]; v[i+1]:=temp;
  fi
 endfor
endfor
qone:=(v[3]+v[4])/2; median:=(v[6]+v[7])/2; qthree:=(v[9]+v[10])/2;
vardef X(expr x)=40+20x enddef;
draw (X(0),50)--(X(12),50) withcolor Grey;
for x=0 step 3 until 12: label.bot(decimal x,(X(x),47)); endfor
draw (X(v[1]),110)--(X(v[N]),110) withcolor Blue;
fill (X(qone),85)--(X(qthree),85)--(X(qthree),135)--(X(qone),135)--cycle withcolor Wash;
draw (X(qone),85)--(X(qthree),85)--(X(qthree),135)--(X(qone),135)--cycle withcolor Blue;
draw (X(median),85)--(X(median),135) withpen pencircle scaled 2 withcolor Orange;
draw (X(v[1]),100)--(X(v[1]),120) withcolor Blue;
draw (X(v[N]),100)--(X(v[N]),120) withcolor Blue;
for i=1 upto N: fill fullcircle scaled 3 shifted (X(v[i]),165+3*(i mod 2)) withcolor Teal; endfor
message "check: quartiles="&decimal qone&","&decimal median&","&decimal qthree;
```

Actual output



MPT-10-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Sorting`	Exchange neighbouring values until the sample is ordered.
`Quartiles`	This convention gives 3.5, 5.5 and 8.
`Whiskers`	Explicitly use minimum 2 and maximum 11.

MPT-10-005

Draw an empirical cumulative distribution

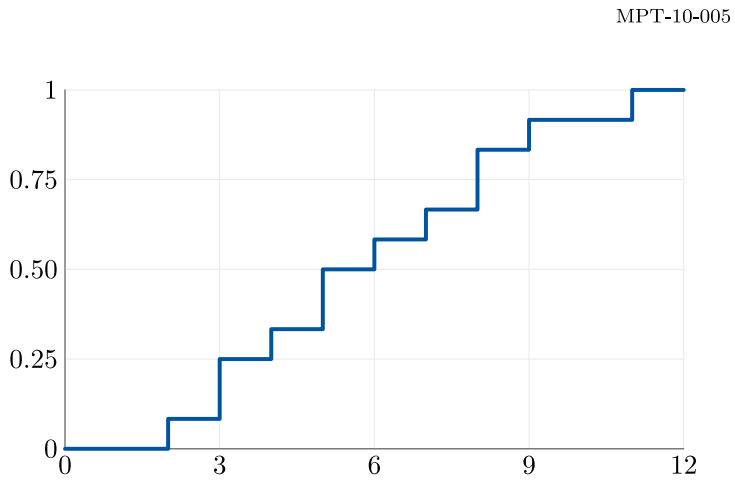
A staircase accumulates the sorted observations. Repeated values produce a combined vertical jump and the curve reaches one after the last observation.

MetaPost drawing

```
numeric xmin,xmax,ymin,ymax,xx,yy;
xmin:=0; xmax:=12; ymin:=0; ymax:=1;
vardef P(expr x,y)=
 (40+250*(x-xmin)/(xmax-xmin),35+145*(y-ymin)/(ymax-ymin))
enddef;
for i=0 upto 4:
 xx:=xmin+i*(xmax-xmin)/4; yy:=ymin+i*(ymax-ymin)/4;
 draw P(xx,ymin)--P(xx,ymax) withcolor Light;
 draw P(xmin,yy)--P(xmax,yy) withcolor Light;
 label.bot(numlabel(xx),P(xx,ymin));
 label.lft(numlabel(yy),P(xmin,yy));
endfor
draw P(xmin,ymax)--P(xmin,ymin)--P(xmax,ymin) withcolor Grey;
numeric v[];
v[1]:=8;
v[2]:=2;
v[3]:=5;
v[4]:=11;
v[5]:=3;
v[6]:=7;
v[7]:=9;
v[8]:=4;
v[9]:=8;
v[10]:=6;
v[11]:=3;
v[12]:=5;
numeric N,temp; N:=12;
for pass=1 upto N-1:
 for i=1 upto N-pass:
  if v[i]>v[i+1]:
   temp:=v[i]; v[i]:=v[i+1]; v[i+1]:=temp;
  fi
 endfor
endfor
path p;
p:=P(0,0);
for i=1 upto N:
 p:=p--P(v[i],(i-1)/N)--P(v[i],i/N);
endfor
```

```
p:=p--P(12,1);
draw p withpen pencircle scaled 1.6 withcolor Blue;
message "check: ecdf-final="&decimal(N/N);
```

Actual output



MPT-10-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`No bins`	Each record contributes mass 1/N at its own value.
`Vertical then horizontal`	Keep the cumulative fraction constant between observations.

MPT-10-006

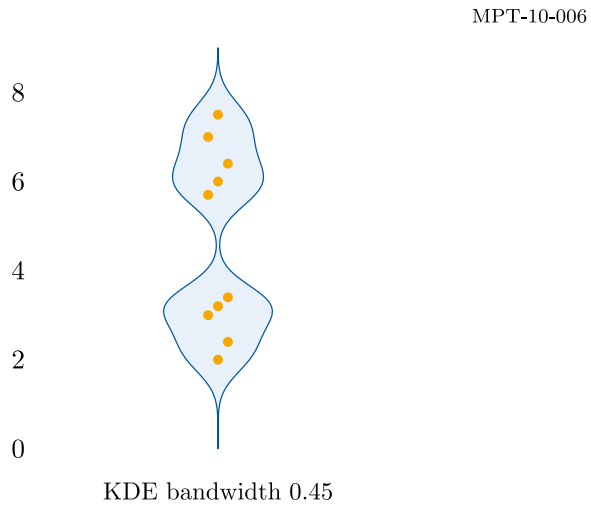
Construct an empirical violin

This violin sums Gaussian kernels at the ten actual observations. Width represents estimated density; the orange dots show the original sample.

MetaPost drawing

```
numeric v[];
v[1]:=2;
v[2]:=2.4;
v[3]:=3;
v[4]:=3.2;
v[5]:=3.4;
v[6]:=5.7;
v[7]:=6;
v[8]:=6.4;
v[9]:=7;
v[10]:=7.5;
numeric N,h; N:=10; h:=.45;
vardef density(expr x)=
 save s,z; numeric s,z; s:=0;
 for j=1 upto N:
  z:=(x-v[j])/h;
  if abs z<6: s:=s+mexp(-128*z*z); fi
 endfor
 s/(N*h*sqrt(2*3.141592654))
enddef;
path p; numeric value;
p:=(160,35);
for k=0 upto 90:
 value:=k/10; p:=p--(160+80density(value),35+18value);
endfor
for k=90 downto 0:
 value:=k/10; p:=p--(160-80density(value),35+18value);
endfor
p:=p--cycle; fill p withcolor Wash; draw p withcolor Blue;
for i=1 upto N: fill fullcircle scaled 4 shifted (160+4*((i mod 3)-1),35+18v[i]) withcolor Orange; endfor
for value=0 step 2 until 8: label.lft(decimal value,(85,35+18value)); endfor
label(btex KDE bandwidth 0.45 etex,(160,18));
```

Actual output



MPT-10-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`mexp(-128*z*z)`	MetaPost mexp(t) means exp(t/256), giving the Gaussian kernel.
`h=0.45`	A common bandwidth in data units.
`Mirrored boundaries`	Trace one density edge upwards and the other downwards before closing.

11 · Circular, polar and projected plots

MPT-11-001

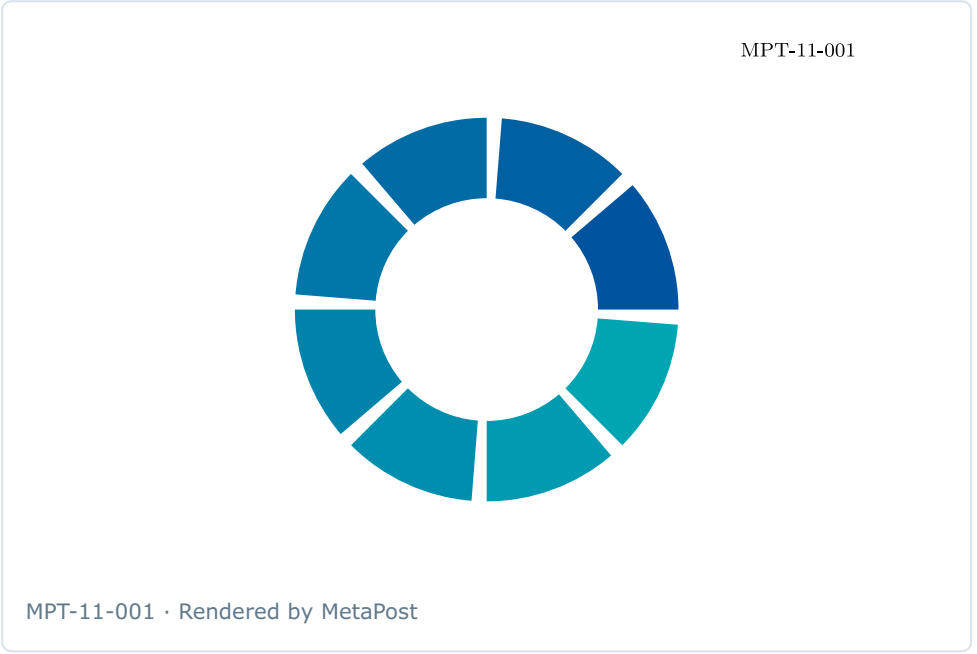
Build a segmented annulus

Subpaths of a circle form ring segments. Rotate a single segment to create a repeated annular composition with visible gaps.

MetaPost drawing

```
path segment;
segment:=(subpath (0,.9) of fullcircle scaled 155)
 --reverse (subpath (0,.9) of fullcircle scaled 90)--cycle;
for i=0 upto 7:
 fill segment rotated (45i) shifted (160,105) withcolor (i/8)[Blue,Teal];
endfor
```

Actual output



MPT-11-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Circular subpath`	Use less than one eighth of the circle to leave gaps.
`Reverse the inner arc`	Join the inner boundary back in the opposite direction.

MPT-11-002

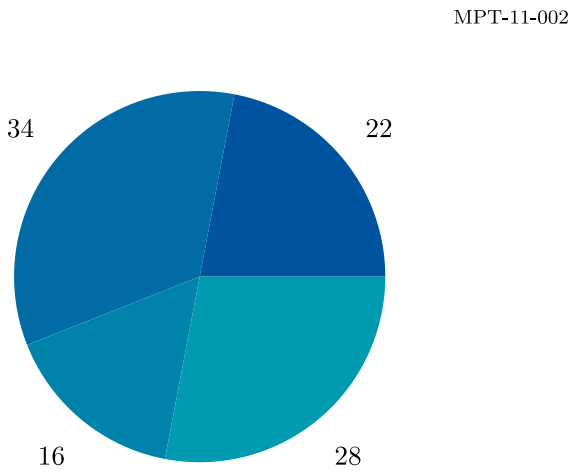
Turn values into pie-sector angles

Four values total one hundred. Their angles are computed from that total; a fine polygonal sampling approximates each circular arc.

MetaPost drawing

```
numeric v[];
v[1]:=22;
v[2]:=34;
v[3]:=16;
v[4]:=28;
numeric total,start,finish; total:=0; start:=0;
for i=1 upto 4: total:=total+v[i]; endfor
path sector;
for i=1 upto 4:
 finish:=start+360v[i]/total;
 sector:=origin--75dir start;
 for a=start+2 step 2 until finish: sector:=sector--75dir a; endfor
 sector:=sector--75dir finish--cycle;
 fill sector shifted (160,105) withcolor ((i-1)/4)[Blue,Teal];
 label(decimal v[i],(160,105)+94dir(.5[start,finish]));
 start:=finish;
endfor
```

Actual output



MPT-11-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`360*value/total`	Compute angular share from data.
`Two-degree arc sampling`	Use explicit points rather than assuming Bezier path time is proportional to angle.

MPT-11-003

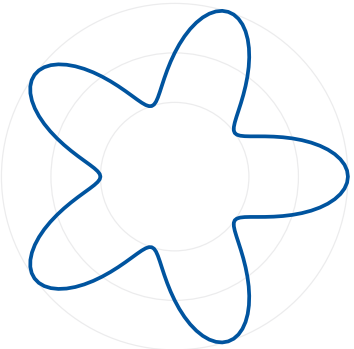
Draw a polar flower

The radius varies with angle. MetaPost trigonometric functions use degrees, so the full turn is zero to 360.

MetaPost drawing

```
path p; numeric r;
for radius=30 step 20 until 70:
 draw fullcircle scaled (2radius) shifted (160,105) withcolor Light;
endfor
p:=for a=0 step 2 until 358:
 if a>0: -- fi
 (160,105)+(50+20cosd(5a))*dir a
endfor--cycle;
draw p withpen pencircle scaled 1.5 withcolor Blue;
```

Actual output



MPT-11-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`r(angle)`	Radius is 50+20*cos(5*angle), always positive here.
`dir a`	Convert polar radius and angle into a pair.

MPT-11-004

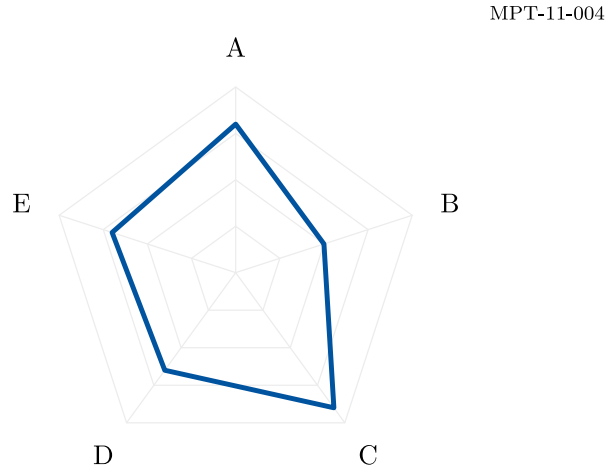
Compare values on a radar polygon

Five normalized values share a zero-to-one radial scale. The grid and the data polygon use the same spoke directions.

MetaPost drawing

```
numeric v[];
v[1]:=0.8;
v[2]:=0.5;
v[3]:=0.9;
v[4]:=0.65;
v[5]:=0.7;
path p;
pair o; o:=(160,105);
for level=.25 step .25 until 1:
 draw for i=1 upto 5:
  if i>1: -- fi o+75level*dir(90-72*(i-1))
 endfor--cycle withcolor Light;
endfor
for i=1 upto 5:
 draw o--(o+75dir(90-72*(i-1))) withcolor Light;
 label(substring(i-1,i) of "ABCDE",o+91dir(90-72*(i-1)));
endfor
p:=for i=1 upto 5:
 if i>1: -- fi o+75v[i]*dir(90-72*(i-1))
endfor--cycle;
draw p withpen pencircle scaled 2 withcolor Blue;
```

Actual output



MPT-11-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Shared normalization`	All values are fractions of the same maximum.
`90-72*(i-1)`	Place five spokes clockwise, starting at the top.

MPT-11-005

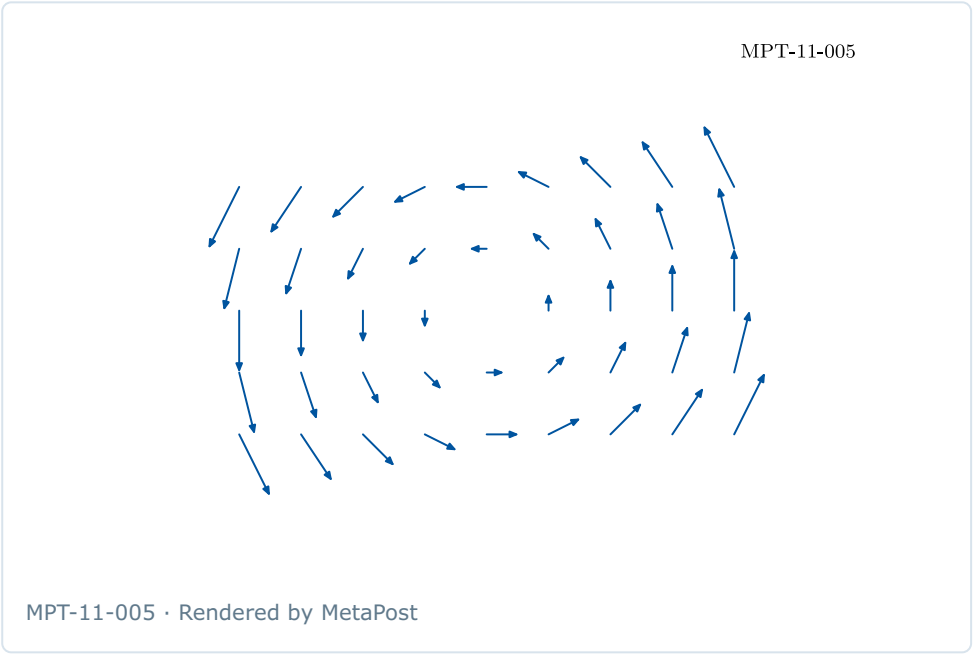
Render a rotational vector field

Each arrow represents F(x,y)=(-y,x). Length is proportional to field magnitude, with one common scale; the zero vector at the centre is omitted.

MetaPost drawing

```
pair base,delta;
ahlength:=3;
for i=-4 upto 4:
 for j=-2 upto 2:
  base:=(160+25i,105+25j); delta:=6*(-j,i);
  if abs delta>0:
   drawarrow base--(base+delta) withpen pencircle scaled .8 withcolor Blue;
  fi
 endfor
endfor
```

Actual output



MPT-11-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Vector displacement`	Rotate the position vector ninety degrees.
`One scale`	Use the same conversion for every vector magnitude.

MPT-11-006

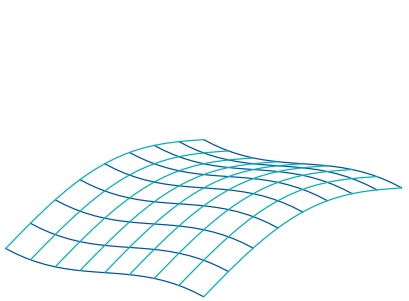
Project a wireframe surface

An explicit affine projection turns three coordinates into a drawing pair. This is a transparent wireframe illustration, without automatic hidden-line removal.

MetaPost drawing

```
vardef project(expr x,y,z)=(160+40*(x-y),55+22*(x+y)+35z) enddef;
vardef height(expr x,y)=.8+.35*(sind(90x)+cosd(90y)) enddef;
for i=0 upto 8:
 draw for j=0 upto 20:
  if j>0: -- fi project(i/4,j/10,height(i/4,j/10))
 endfor withcolor Blue;
endfor
for j=0 upto 8:
 draw for i=0 upto 20:
  if i>0: -- fi project(i/10,j/4,height(i/10,j/4))
 endfor withcolor Teal;
endfor
```

Actual output



MPT-11-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`project(x,y,z)`	A pair-valued function implements the chosen view.
`Two line families`	Sample rows and columns of the same surface.
`No visibility solver`	All grid strokes remain visible.

12 · The graph library and publication

MPT-12-001

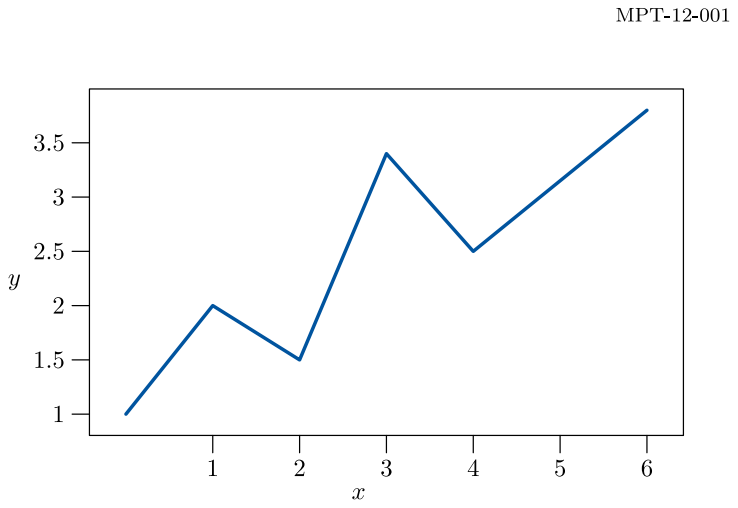
Let graph.mp choose coordinate ranges

The standard graph library maps data coordinates and draws axes. The data remain an inline path, so no external file is required.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 gdraw (0,1)--(1,2)--(2,1.5)--(3,3.4)--(4,2.5)--(6,3.8) withpen pencircle scaled 1.4
withcolor Blue;
 glabel.lft(btex $y$ etex,OUT);
 glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
```

Actual output



MPT-12-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`input graph`	Load MetaPost standard plotting macros.
`begingraph(width,height)`	Specify the plotting rectangle, not data ranges.
`gdraw`	Map the supplied data path into the current graph coordinates.

MPT-12-002

Set explicit graph ranges

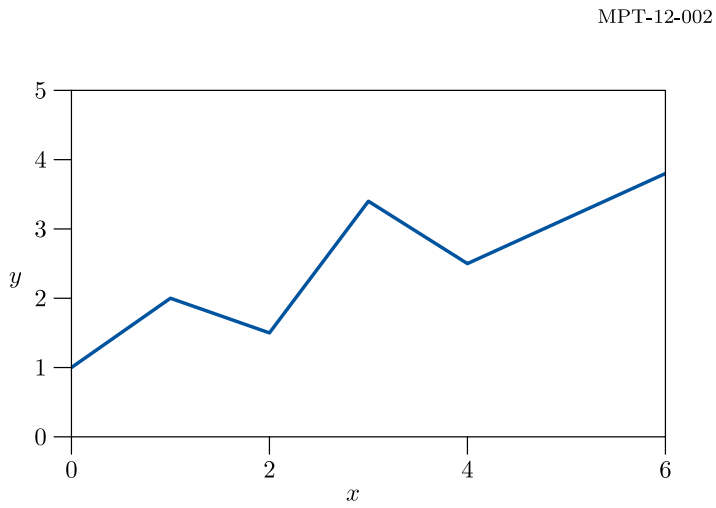
Keep the same observations but request a zero baseline and a fixed upper bound.

Compare with MPT-12-001 · Let graph.mp choose coordinate ranges.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 setrange(0,0,6,5);
 gdraw (0,1)--(1,2)--(2,1.5)--(3,3.4)--(4,2.5)--(6,3.8) withpen pencircle scaled 1.4
withcolor Blue;
 glabel.lft(btex $y$ etex,OUT);
 glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
```

Actual output



MPT-12-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`setrange(xmin,ymin,xmax,ymax)`	Choose explicit ranges before the data are drawn.
`OUT labels`	Place axis labels relative to the entire graph.

MPT-12-003

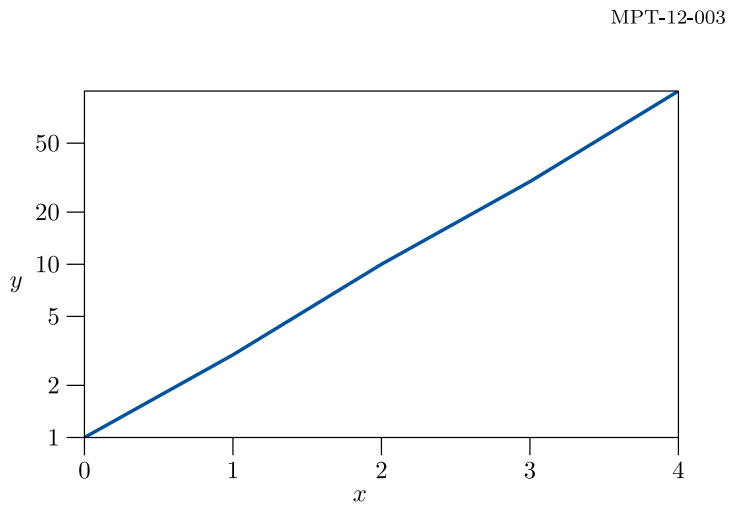
Use a logarithmic graph axis

The y values span two decades. A logarithmic coordinate system makes their ratios, rather than differences, equally spaced.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 setcoords(linear,log);
 setrange(0,1,4,100);
 gdraw (0,1)--(1,3)--(2,10)--(3,30)--(4,100) withpen pencircle scaled 1.4 withcolor
Blue;
 glabel.lft(btex $y$ etex,OUT);
 glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
```

Actual output



MPT-12-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`setcoords(linear,log)`	Keep x linear and transform the positive y values logarithmically.
`Positive range`	The requested logarithmic axis begins at one.

MPT-12-004

Fill data in graph coordinates

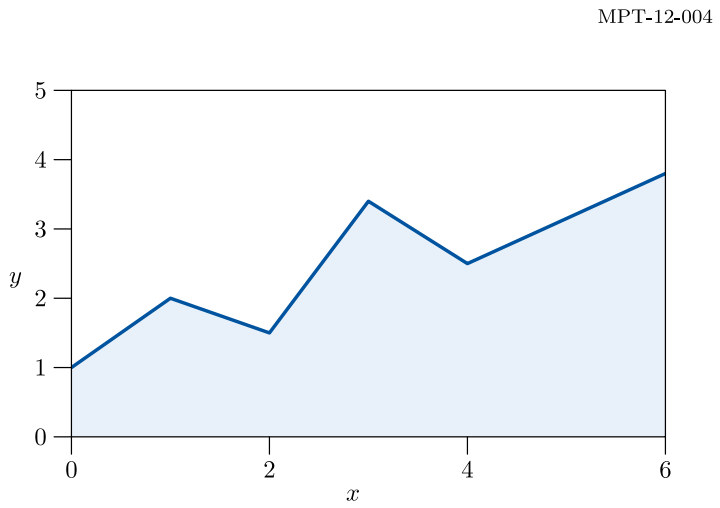
gfill applies the same data-to-picture mapping as gdraw. Close the area against zero before filling it.

Compare with MPT-12-002 · Set explicit graph ranges.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 setrange(0,0,6,5);
 gfill (0,0)--(0,1)--(1,2)--(2,1.5)--(3,3.4)--(4,2.5)--(6,3.8)--(6,0)--cycle
withcolor Wash;
 gdraw (0,1)--(1,2)--(2,1.5)--(3,3.4)--(4,2.5)--(6,3.8) withpen pencircle scaled 1.4
withcolor Blue;
 glabel.lft(btex $y$ etex,OUT);
 glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
```

Actual output



MPT-12-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`gfill`	Fill a closed path expressed in data coordinates.
`Same mapping`	Use the same graph range for the fill and the line.

MPT-12-005

Add an automatic graph grid

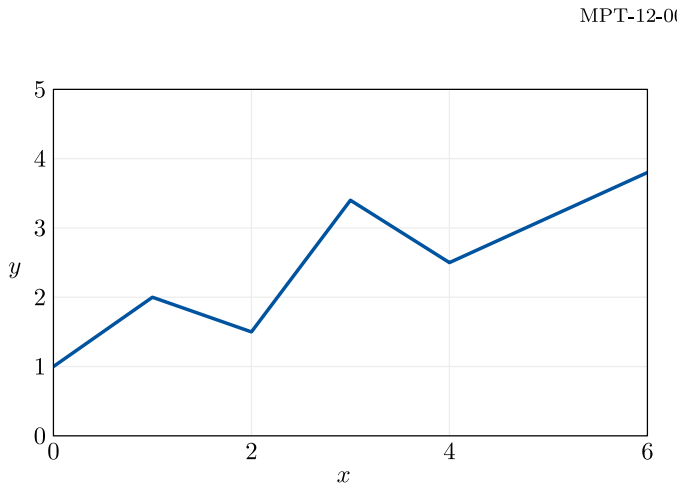
Let the library choose labelled grid positions from the current ranges. Keep grid strokes light so that the data remain dominant.

Compare with MPT-12-002 · Set explicit graph ranges.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 setrange(0,0,6,5);
 autogrid(grid.bot,grid.lft) withcolor Light;
 gdraw (0,1)--(1,2)--(2,1.5)--(3,3.4)--(4,2.5)--(6,3.8) withpen pencircle scaled 1.4
withcolor Blue;
 glabel.lft(btex $y$ etex,OUT);
 glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
```

Actual output



MPT-12-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`autogrid`	Iterate the automatically selected axis positions.
`grid.bot / grid.lft`	Draw grid lines and labels for the bottom and left axes.

MPT-12-006

Compose two data series and a legend

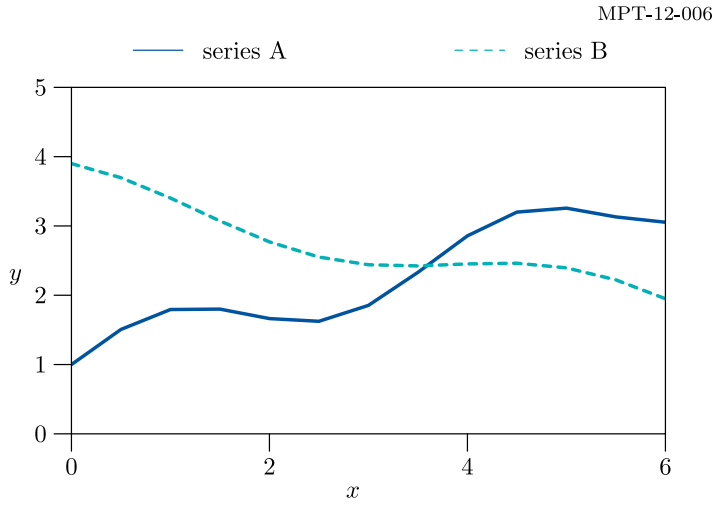
Prepare two paths, let graph.mp map them together, then add a compact legend as ordinary MetaPost drawing.

Compare with MPT-12-002 · Set explicit graph ranges.

MetaPost drawing

```
input graph;
path a,b;
a:=for i=0 upto 12: if i>0: -- fi (i/2,1+.4*(i/2)+.4sind(50i)) endfor;
b:=for i=0 upto 12: if i>0: -- fi (i/2,3.6-.3*(i/2)+.3cosd(35i)) endfor;
draw begingraph(240,140);
 setrange(0,0,6,5);
 gdraw a withpen pencircle scaled 1.4 withcolor Blue;
 gdraw b dashed evenly withpen pencircle scaled 1.4 withcolor Teal;
 glabel.lft(btex $y$ etex,OUT); glabel.bot(btex $x$ etex,OUT);
endgraph shifted (50,40);
draw (75,195)--(95,195) withcolor Blue;
label.rt(btex series A etex,(100,195));
draw (205,195)--(225,195) dashed evenly withcolor Teal;
label.rt(btex series B etex,(230,195));
```

Actual output



MPT-12-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Two paths`	Keep series identity separate from the shared coordinate system.
`Manual legend`	Compose the graph with the same text and geometry tools used elsewhere.

13 · External data and incomplete observations

Keep measurements separate from drawing instructions. The source-and-data ZIP preserves relative filenames; extract the whole archive before running its .mp program. The bundled data/README.md defines columns, units and provenance. All measurements here are synthetic. Bounds are illustrative display intervals, not confidence intervals. The validity flag distinguishes an absent observation from a measured zero. graph.mp expects numeric whitespace-separated fields and stops at a blank line; remove headers and separators deliberately before use. The explicit readfrom example instead shows how to skip an empty line.

MPT-13-001

Plot a two-column file with graph.mp

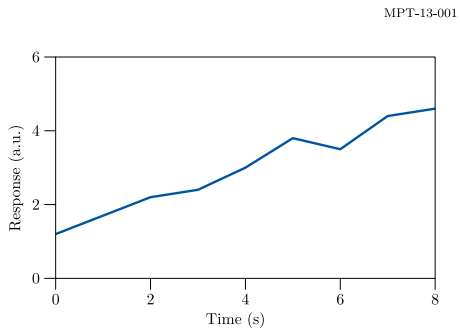
Pass a filename directly to gdraw. The first column is time in seconds; the second is response in arbitrary units. Explicit ranges make subsequent comparisons honest.

Compare with MPT-12-002 · Set explicit graph ranges.

MetaPost drawing

```
input graph;
draw begingraph(240,140);
 setrange(0,0,8,6);
 gdraw "data/calibration.dat" withcolor Blue
   withpen pencircle scaled 1.5;
 glabel.bot(btex Time (s) etex,OUT);
 glabel.lft(btex Response (a.u.) etex rotated 90,OUT);
endgraph shifted (55,42);
```

Actual output



MPT-13-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`gdraw "filename"`	Read consecutive x/y pairs from a two-column numeric file.
`setrange(0,0,8,6)`	Fix both axes rather than letting each dataset choose a range.

MPT-13-002

Select columns from a wider table

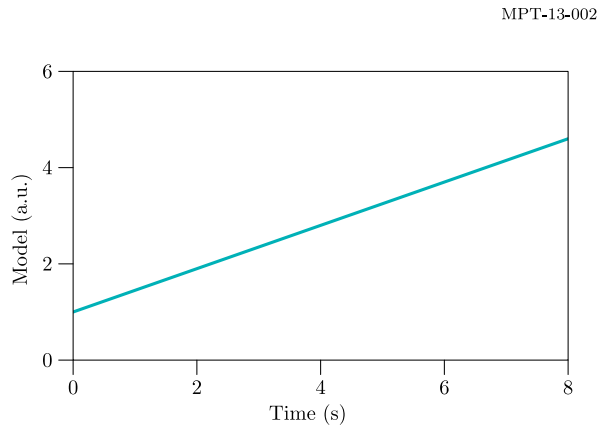
readings.dat has six fields per row: time, observation, model, lower bound, upper bound and validity. Select time and model here; the model is defined even where the observation is absent.

Compare with MPT-13-001 · Plot a two-column file with graph.mp.

MetaPost drawing

```
input graph;
path modelpath;
gdata("data/readings.dat", row,
 if i=1: modelpath:=(scantokens row1,scantokens row3);
 else: augment.modelpath(scantokens row1,scantokens row3); fi
);
draw begingraph(240,140);
 setrange(0,0,8,6);
 gdraw modelpath withcolor Teal withpen pencircle scaled 1.5;
 glabel.bot(btex Time (s) etex,OUT);
 glabel.lft(btex Model (a.u.) etex rotated 90,OUT);
endgraph shifted (55,42);
```

Actual output



MPT-13-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`gdata(file,row,commands)`	Execute commands for each row; row1, row2, etc. are strings.
`scantokens row3`	Convert a reviewed numeric field into a number.
`augment.modelpath`	Append a selected coordinate pair to the existing path.

MPT-13-003

Read pairs and skip an empty line

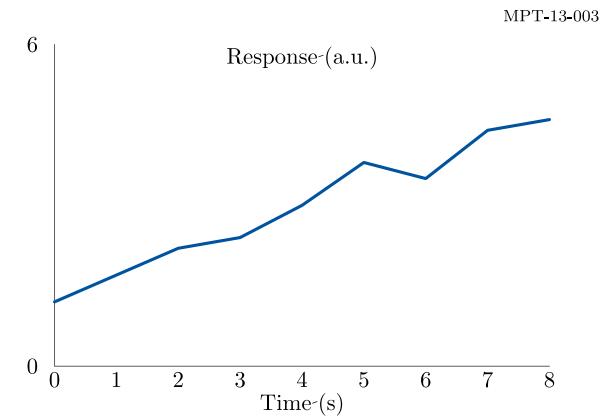
readfrom returns one string per physical line. Stop at EOF and skip the deliberately empty line in points.dat. scantokens parses the reviewed pair notation; it is not a general CSV reader.

Compare with MPT-13-001 · Plot a two-column file with graph.mp.

MetaPost drawing

```
string line; pair p; path curve; boolean started;
started:=false;
forever:
 line:=readfrom "data/points.dat";
 exitif line=EOF;
 if line<>"":
  p:=scantokens line;
  p:=(45+30*xpart p,40+26*ypart p);
  if started: curve:=curve--p; else: curve:=p; fi
  started:=true;
 fi
endfor;
draw (45,196)--(45,40)--(285,40) withcolor Grey;
draw curve withpen pencircle scaled 1.5 withcolor Blue;
label.bot("Time (s)",(165,30));
label.lft("0",(40,40)); label.lft("6",(40,196));
for x=0 upto 8: label.bot(decimal x,(45+30x,40)); endfor;
label.top("Response (a.u.)",(165,181));
```

Actual output



MPT-13-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`readfrom / EOF`	Read each physical line once; stop before another read restarts the file.
`line<>""`	Skip an empty line explicitly.
`scantokens line`	Parse trusted numeric pairs; arbitrary text may contain executable tokens.

MPT-13-004

Leave a gap for a missing observation

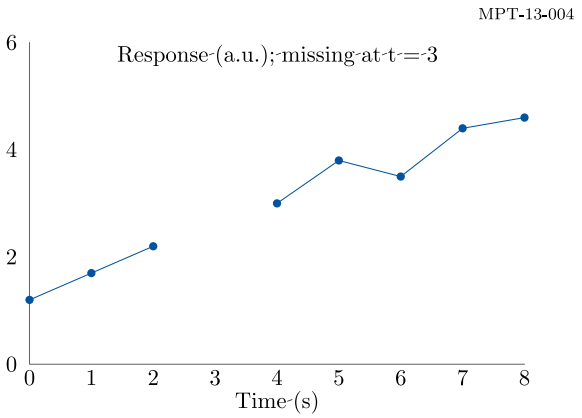
Ignore the placeholder at t = 3 and reset the connection flag. A measured zero would still be drawn when validity = 1. Joining every valid point would misleadingly bridge the missing observation.

Compare with MPT-13-002 · Select columns from a wider table.

MetaPost drawing

```
input graph;
numeric x,y; pair p,previous; boolean connected;
connected:=false;
draw (45,196)--(45,40)--(285,40) withcolor Grey;
gdata("data/readings.dat", row,
 if scantokens row6=1:
  x:=scantokens row1; y:=scantokens row2;
  p:=(45+30x,40+26y);
  if connected: draw previous--p withcolor Blue; fi
  fill fullcircle scaled 4 shifted p withcolor Blue;
  previous:=p; connected:=true;
 else: connected:=false; fi
);
label.bot("Time (s)",(165,30));
label.top("Response (a.u.); missing at t = 3",(165,181));
for x=0 upto 8: label.bot(decimal x,(45+30x,40)); endfor;
for y=0 step 2 until 6: label.lft(decimal y,(42,40+26y)); endfor;
```

Actual output



MPT-13-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`row6=1`	Use the explicit validity flag, never the placeholder value.
`connected:=false`	Break the line at a missing row.
`previous--p`	Draw segments only between consecutive observed rows.

MPT-13-005

Draw asymmetric intervals from columns

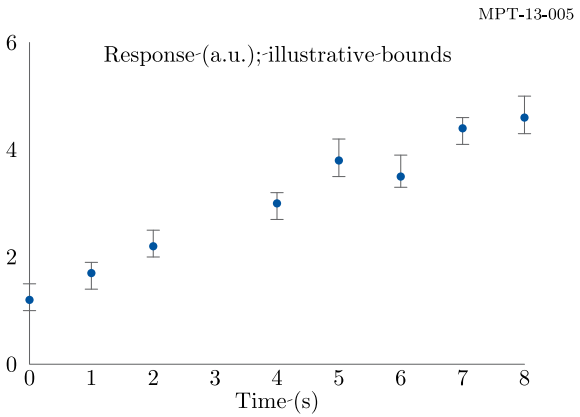
Read the lower and upper endpoints independently. The observation need not sit at their midpoint. Skip the absent row and use a short horizontal cap at each endpoint; these synthetic bounds have no confidence-level interpretation.

Compare with MPT-13-004 · Leave a gap for a missing observation.

MetaPost drawing

```
input graph;
numeric x,y,lo,hi;
draw (45,196)--(45,40)--(285,40) withcolor Grey;
gdata("data/readings.dat", row,
 if scantokens row6=1:
  x:=45+30*(scantokens row1); y:=40+26*(scantokens row2);
  lo:=40+26*(scantokens row4); hi:=40+26*(scantokens row5);
  draw (x,lo)--(x,hi) withcolor Grey;
  draw (x-3,lo)--(x+3,lo) withcolor Grey;
  draw (x-3,hi)--(x+3,hi) withcolor Grey;
  fill fullcircle scaled 4 shifted (x,y) withcolor Blue;
 fi
);
label.bot("Time (s)",(165,30));
label.top("Response (a.u.); illustrative bounds",(165,181));
for x=0 upto 8: label.bot(decimal x,(45+30x,40)); endfor;
for y=0 step 2 until 6: label.lft(decimal y,(42,40+26y)); endfor;
```

Actual output



MPT-13-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`row4 / row5`	Store actual interval endpoints, not a symmetric plus/minus width.
`Short caps`	Keep interval ends visible without obscuring the points.
`row6`	Apply the same validity rule to the point and both endpoints.

MPT-13-006

Compute a centred moving mean

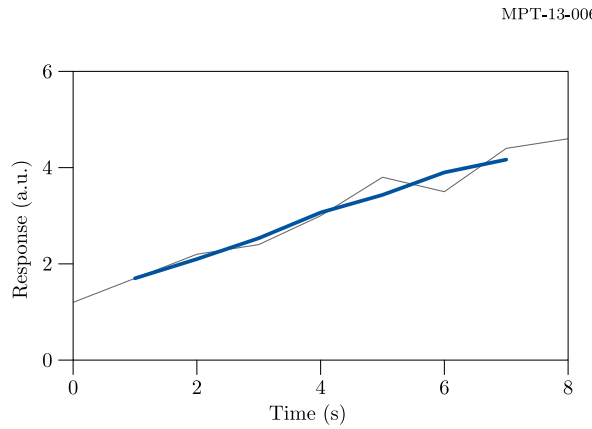
Read the complete calibration series into arrays. Average each interior triple and plot the result at its middle time. Omit the two endpoints instead of silently padding them or changing the window width.

Compare with MPT-13-001 · Plot a two-column file with graph.mp.

MetaPost drawing

```
input graph;
numeric n,t[],v[]; path raw,smoothed;
n:=0;
gdata("data/calibration.dat", row,
 n:=n+1; t[n]:=scantokens row1; v[n]:=scantokens row2;
);
raw:=for j=1 upto n: if j>1: -- fi (t[j],v[j]) endfor;
smoothed:=for j=2 upto n-1: if j>2: -- fi
 (t[j],(v[j-1]+v[j]+v[j+1])/3) endfor;
draw begingraph(240,140);
 setrange(0,0,8,6);
 gdraw raw withcolor Grey;
 gdraw smoothed withpen pencircle scaled 1.8 withcolor Blue;
 glabel.bot(btex Time (s) etex,OUT);
 glabel.lft(btex Response (a.u.) etex rotated 90,OUT);
endgraph shifted (55,42);
message "MPT-13-006: rows=" & decimal n & "; centres=" & decimal(n-2);
```

Actual output



MPT-13-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`v[j-1]+v[j]+v[j+1]`	A three-observation centred mean; appropriate here for equally spaced times.
`2 upto n-1`	Draw only locations with a complete window.
`message`	Record nine input rows and seven computed centres in the compiler log.

14 · Complete publication figures

Use the same synthetic readings in several compositions. Each complete program loads the six columns into t, observed, model, low, high and valid arrays before drawing; expand its source to see that reader. All panels use explicit scales, shared units and the same Computer Modern typography. Ordinary MetaPost pairs, pictures and transformations are sufficient for layout. Keep the published figure's physical size in mind: resizing a complete picture also resizes its lettering and pen widths. The capstone specifies a 420 by 280 bp canvas.

MPT-14-001

Give two panels the same scale

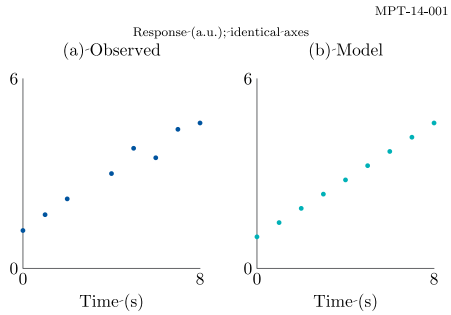
Draw observation and model in equal panels. Both share x = 0..8 and y = 0..6; the observed panel omits the missing observation. Different automatic ranges would exaggerate or suppress their visual difference.

Compare with MPT-13-004 · Leave a gap for a missing observation.

MetaPost drawing

```
vardef panel(expr origin,which)=
 draw (origin+(0,120))--origin--(origin+(112,0)) withcolor Grey;
 for j=1 upto n:
  if (which=2) or (valid[j]=1):
   fill fullcircle scaled 3 shifted
    (origin+(14t[j],20*(if which=1: observed[j] else: model[j] fi)))
    withcolor if which=1: Blue else: Teal fi;
  fi
 endfor;
 label.lft("0",origin); label.lft("6",origin+(0,120));
 label.bot("0",origin); label.bot("8",origin+(112,0));
 label.bot("Time (s)",origin+(56,-13));
enddef;
panel((40,47),1); panel((188,47),2);
label.top("(a) Observed",(96,176));
label.top("(b) Model",(244,176));
label("Response (a.u.); identical axes" infont "cmr8",(165,196));
```

Actual output



MPT-14-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`panel(origin,which)`	Reuse one panel geometry; select only the displayed data.
`14t[j],20y`	Use identical units per point in both panels.
`valid[j]`	Omit the missing observation in panel (a).

MPT-14-002

Align observations and residuals

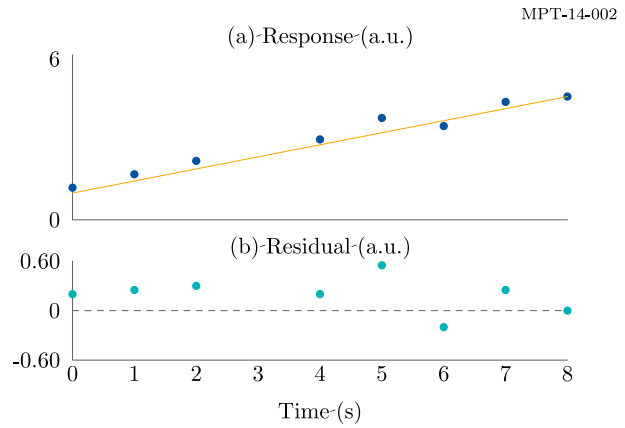
Use one x mapping for an upper response panel and a lower residual panel. Residual means observed minus model, so positive residuals appear above zero. The missing observation contributes to neither panel.

Compare with MPT-14-001 · Give two panels the same scale.

MetaPost drawing

```
vardef xx(expr x)=45+30x enddef;
draw (45,190)--(45,110)--(285,110) withcolor Grey;
draw (45,90)--(45,42)--(285,42) withcolor Grey;
draw (45,66)--(285,66) dashed evenly withcolor Grey;
for j=1 upto n:
 if valid[j]=1:
  fill fullcircle scaled 4 shifted (xx(t[j]),110+13observed[j]) withcolor Blue;
  fill fullcircle scaled 4 shifted
   (xx(t[j]),66+40*(observed[j]-model[j])) withcolor Teal;
 fi
endfor;
draw (xx(0),123)--(xx(8),110+13model[n]) withcolor Orange;
label.lft("0",(42,110)); label.lft("6",(42,188));
for y=-.6 step .6 until .6: label.lft(numlabel(y),(42,66+40y)); endfor;
label.top("(a) Response (a.u.)",(165,190));
label.top("(b) Residual (a.u.)",(165,88));
for j=0 upto 8: label.bot(decimal j,(xx(j),42)); endfor;
label.bot("Time (s)",(165,25));
```

Actual output



MPT-14-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`xx(x)`	Share exactly the same horizontal transform between panels.
`observed[j]-model[j]`	Define the residual and its sign explicitly.
`Dashed zero line`	Provide a visual reference for positive and negative residuals.

MPT-14-003

Reuse panel pictures without rescaling text

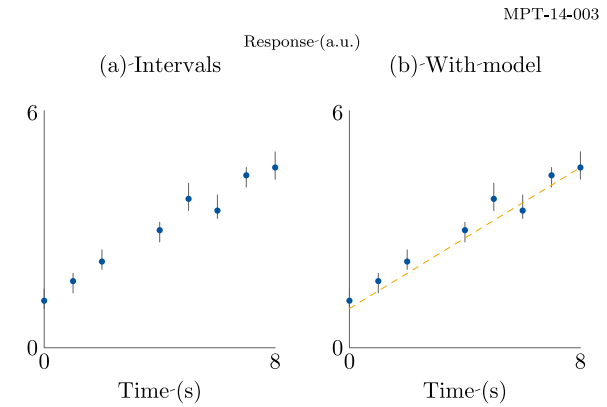
Capture a complete panel in a picture and move two copies. Shifting preserves font size and stroke width. A shared y label and matching axes keep the composition compact; panel letters distinguish the two views.

Compare with MPT-14-001 · Give two panels the same scale.

MetaPost drawing

```
picture panel;
panel:=image(
 draw (0,115)--(0,0)--(112,0) withcolor Grey;
 for j=1 upto n:
  if valid[j]=1:
   draw (14t[j],19low[j])--(14t[j],19high[j]) withcolor Grey;
   fill fullcircle scaled 3 shifted (14t[j],19observed[j]) withcolor Blue;
  fi
 endfor;
 label.lft("0",(0,0)); label.lft("6",(0,114));
 label.bot("0",(0,0)); label.bot("8",(112,0));
 label.bot("Time (s)",(56,-13));
);
draw panel shifted (40,48);
draw panel shifted (188,48);
label.top("(a) Intervals",(96,176));
label.top("(b) With model",(244,176));
draw (188,67)--(300,48+19model[n])
 dashed evenly withcolor Orange;
label("Response (a.u.)" infont "cmr8",(165,196));
```

Actual output



MPT-14-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`image(...)`	Collect axes, marks and labels in one reusable picture.
`shifted`	Translate a panel while preserving physical type and pen sizes.
`Overlay after placement`	Add the model only to the second translated panel.

MPT-14-004

Normalise two series to a shared baseline

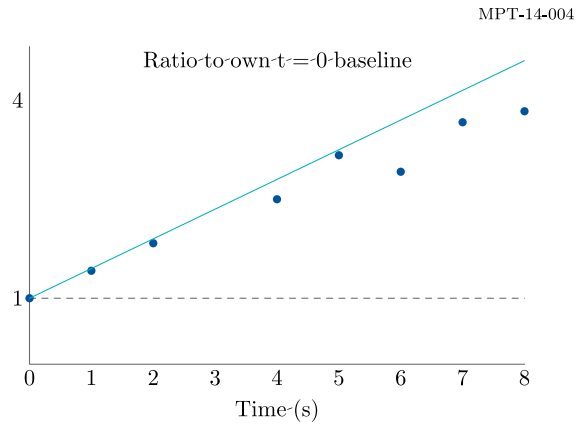
Divide each observed and model response by its own value at t = 0. Both start at one. The dimensionless ratios compare relative change, rather than making different starting values disappear without explanation.

Compare with MPT-14-002 · Align observations and residuals.

MetaPost drawing

```
vardef xy(expr x,y)=(45+30x,40+32y) enddef;
numeric a,b;
a:=observed[1]; b:=model[1];
draw (45,194)--(45,40)--(285,40) withcolor Grey;
draw xy(0,1)--xy(8,1) dashed evenly withcolor Grey;
for j=1 upto n:
 if valid[j]=1:
  fill fullcircle scaled 4 shifted xy(t[j],observed[j]/a) withcolor Blue;
 fi
endfor;
draw xy(0,model[1]/b)--xy(8,model[n]/b) withcolor Teal;
label.lft("1",xy(0,1)); label.lft("4",xy(0,4));
for j=0 upto 8: label.bot(decimal j,xy(j,0)); endfor;
label.bot("Time (s)",(165,26));
label.top("Ratio to own t = 0 baseline",(165,181));
message "MPT-14-004: observed baseline=" & numlabel(a)
 & "; model baseline=" & numlabel(b);
```

Actual output



MPT-14-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`observed[1] / model[1]`	Use each series' first valid, nonzero baseline; these fixtures provide both at t = 0.
`Dimensionless ratio`	State the transformation so the reader can recover its meaning.
`Baseline reference`	A ratio of one means no change from that series' starting value.

MPT-14-005

Make the legend survive greyscale printing

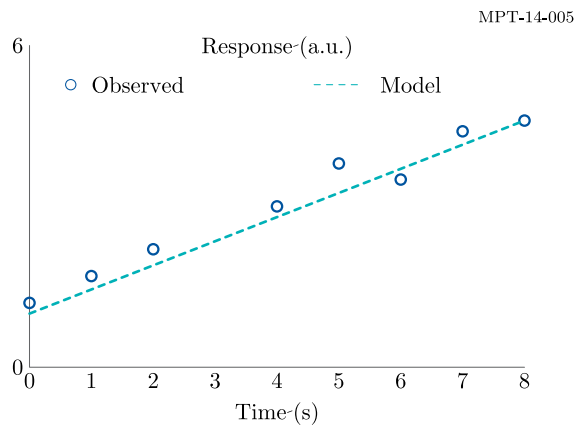
Use hollow circles for observed values and a dashed line for the model. Shape and stroke carry series identity alongside colour. Draw the legend with those exact marks and keep it inside the canvas.

Compare with MPT-14-002 · Align observations and residuals.

MetaPost drawing

```
vardef xy(expr x,y)=(45+30x,40+26y) enddef;
draw (45,196)--(45,40)--(285,40) withcolor Grey;
for j=1 upto n:
 if valid[j]=1:
  draw fullcircle scaled 5 shifted xy(t[j],observed[j])
   withpen pencircle scaled 1.1 withcolor Blue;
 fi
endfor;
draw xy(0,model[1])--xy(8,model[n]) dashed evenly
 withpen pencircle scaled 1.3 withcolor Teal;
draw fullcircle scaled 5 shifted (65,177) withcolor Blue;
label.rt("Observed",(72,177));
draw (183,177)--(205,177) dashed evenly withcolor Teal;
label.rt("Model",(212,177));
for j=0 upto 8: label.bot(decimal j,xy(j,0)); endfor;
label.bot("Time (s)",(165,26));
label.lft("0",xy(0,0)); label.lft("6",xy(0,6));
label("Response (a.u.)",(165,195));
```

Actual output



MPT-14-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Hollow circle`	Identify observations by shape as well as colour.
`dashed evenly`	Give the model a distinct stroke in colour and greyscale.
`Matching legend marks`	Use the same shape, dash and colour as the plotted series.

MPT-14-006

Assemble a titled figure with an audit trail

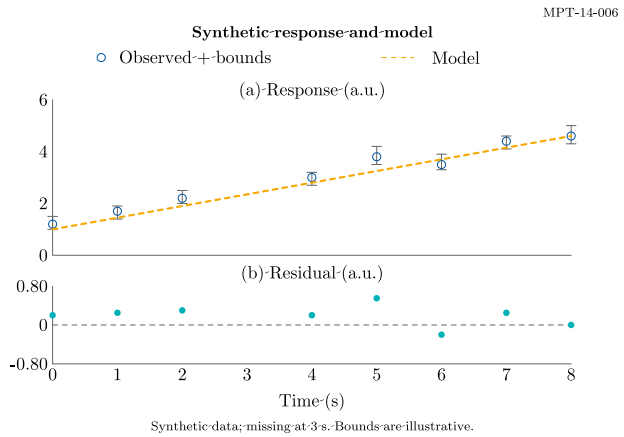
Combine asymmetric bounds, a dashed model, aligned residuals, physical units, panel letters and a concise caption. The expanded program contains the file reader and panel helpers. Its log records the eight valid observations and mean residual; the downloadable PDF embeds the exact source and inputs.

Compare with MPT-14-005 · Make the legend survive greyscale printing.

MetaPost drawing

```
publication_axes;
for j=1 upto n:
 if valid[j]=1:
  interval_at(t[j],low[j],high[j]);
  draw fullcircle scaled 5 shifted mainpoint(t[j],observed[j]) withcolor Blue;
  fill fullcircle scaled 4 shifted residualpoint(t[j],observed[j]-model[j])
   withcolor Teal;
 fi
endfor;
draw mainpoint(0,model[1])--mainpoint(8,model[n])
 dashed evenly withpen pencircle scaled 1.3 withcolor Orange;
label("Synthetic response and model" infont "cmbx10",(220,257));
draw fullcircle scaled 5 shifted (90,242) withcolor Blue;
label.rt("Observed + bounds",(98,242));
draw (263,242)--(283,242) dashed evenly withcolor Orange;
label.rt("Model",(291,242));
label("Synthetic data; missing at 3 s. Bounds are illustrative." infont "cmr8",(220,14));
message "MPT-14-006: valid=" & decimal count
 & "; residual sum=" & numlabel(total)
 & "; mean=" & decimal(total/count);
```

Actual output



MPT-14-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`420 by 280 bp`	Choose the final canvas before setting typography; bp means 1/72 inch.
`mainpoint / residualpoint`	Use shared x coordinates and separate explicit y ranges.
`interval_at`	Draw the stored lower and upper endpoints, with caps.
`message`	Capture numerical checks without drawing rounded values back into calculations.

15 · Drawing recognisable objects

Build original illustrations from silhouettes, local tangents, clipped textures, ellipses, depth layers and projections.

MPT-15-001

Draw a leaf from a silhouette and veins

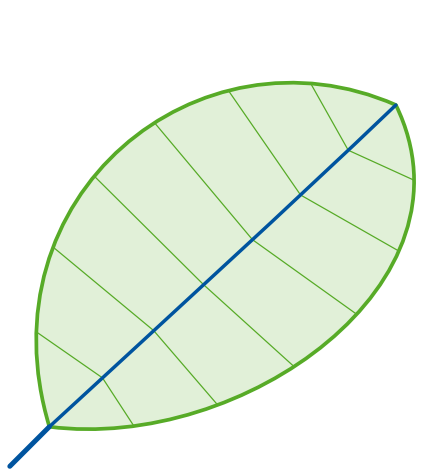
Two cubic segments form the silhouette. Sample its boundary and a separate midrib to anchor the veins; draw the pale fill first.

Compare with MPT-02-003 · Expose cubic control handles.

MetaPost drawing

```
path leaf,midrib;
leaf:=(92,45)..controls (64,138) and (152,210)..(232,175)
  ..controls (267,103) and (170,35)..cycle;
midrib:=(92,45)..controls (143,92) and (187,132)..(232,175);
fill leaf withcolor .82[Green,white];
draw leaf withpen pencircle scaled 1.5 withcolor Green;
for i=1 upto 6:
  draw point (i/7) of midrib--point (i/7) of leaf withcolor Green;
  draw point (i/7) of midrib--point (2-i/7) of leaf withcolor Green;
endfor;
draw midrib withpen pencircle scaled 1.4 withcolor Blue;
draw (76,29)--(92,45) withpen pencircle scaled 2 withcolor Blue;
```

Actual output



MPT-15-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`controls`	Shape each side with its own cubic handles.
`point t of leaf`	Use times 0-1 for one side and 1-2 for the other.

MPT-15-002

Grow a feather along a curved spine

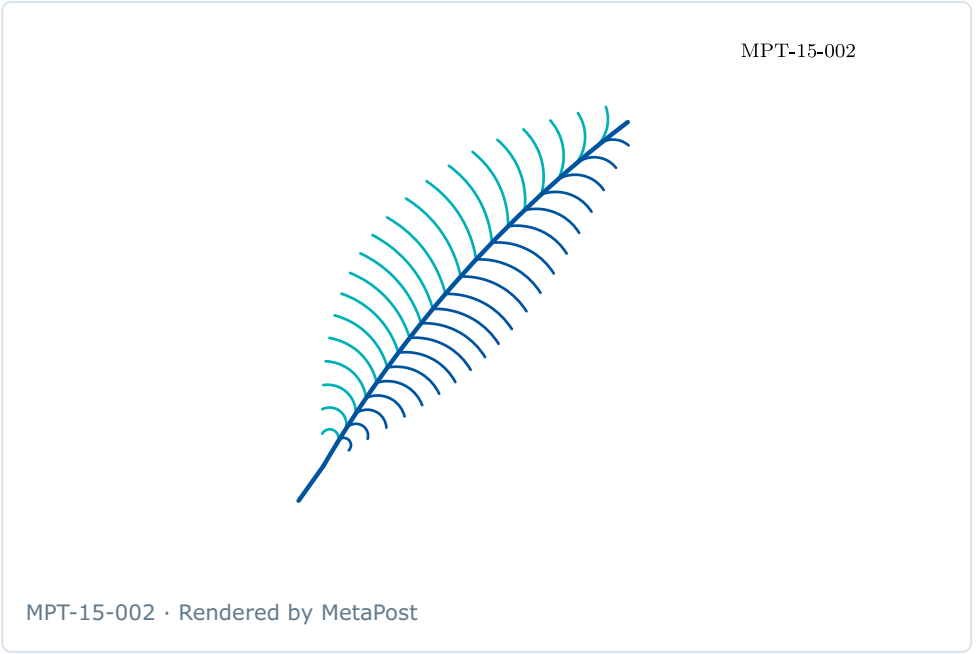
Replace the filled silhouette with individual barbs. At each sample, the spine tangent supplies a local direction and its rotated unit vector supplies a normal. A sine envelope narrows the feather at both ends.

Compare with MPT-15-001 · Draw a leaf from a silhouette and veins.

MetaPost drawing

```
path spine;
pair c,tangent,normal,lefttip,righttip;
numeric w;
spine:=(94,42)..(142,110)..(217,181);
for t=.06 step .045 until .96:
  c:=point (2t) of spine;
  tangent:=unitvector direction (2t) of spine;
  normal:=tangent rotated 90;
  w:=38sind(180t);
  lefttip:=c+w*normal+16t*tangent;
  righttip:=c-.78w*normal+13t*tangent;
  draw c..(.55[c,lefttip]+4tangent)..lefttip
    withpen pencircle scaled 1.1 withcolor Teal;
  draw c..(.55[c,righttip]+4tangent)..righttip
    withpen pencircle scaled 1.1 withcolor Blue;
endfor;
draw spine withpen pencircle scaled 1.8 withcolor Blue;
draw (84,28)--(94,42) withpen pencircle scaled 1.8 withcolor Blue;
```

Actual output



MPT-15-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`unitvector direction t of spine`	Normalise the tangent so barb widths use drawing units.
`rotated 90`	Construct a perpendicular local direction at each sample.
`38sind(180t)`	Use a bounded width envelope along the feather.

MPT-15-003

Clip a scale texture inside a fish

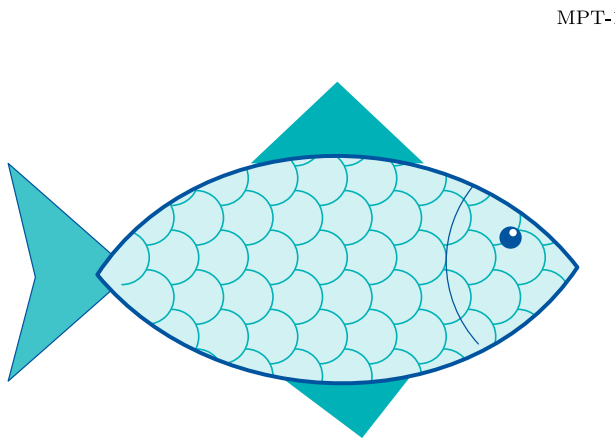
Layer tail and fins behind a smooth body. Repeated half circles make the scales in a separate picture, then a clip confines them to the body. Add the gill and eye last so they remain readable.

Compare with MPT-06-002 · Reuse the field with another mask.

MetaPost drawing

```
path body,tail;
picture scales;
body:=(77,106)..controls (119,170) and (225,169)..(271,109)
  ..controls (229,46) and (125,48)..cycle;
tail:=(91,107)--(41,151)--(52,105)--(41,63)--cycle;
fill tail withcolor .25[Teal,white];
draw tail withcolor Blue;
fill (139,151)--(174,184)--(209,151)--cycle withcolor Teal;
fill (148,67)--(184,40)--(206,68)--cycle withcolor Teal;
fill body withcolor .82[Teal,white];
scales:=image(
  for row=0 upto 7:
    for col=0 upto 8:
      draw halfcircle scaled 22 rotated -90
        shifted (87+20col+10(row mod 2),49+16row)
        withpen pencircle scaled .65 withcolor Teal;
    endfor;
  endfor;
);
clip scales to body; draw scales;
draw body withpen pencircle scaled 1.6 withcolor Blue;
draw (229,142)..(218,111)..(231,78) withcolor Blue;
fill fullcircle scaled 9 shifted (244,121) withcolor Blue;
fill fullcircle scaled 3 shifted (245,123) withcolor white;
```

Actual output



MPT-15-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`image(...)`	Collect texture strokes without clipping the fins or eye.
`row mod 2`	Offset alternate rows of scales.
`clip scales to body`	Retain only the texture inside the body silhouette.

MPT-15-004

Assemble a cup with a handle and steam

Ellipses establish the rim and saucer. The handle is a broad curve drawn behind the cup, while a smaller ellipse inside the rim suggests coffee. Open curves above it read as steam without a filled outline.

Compare with MPT-15-003 · Clip a scale texture inside a fish.

MetaPost drawing

```
path cup,handle;
fill fullcircle xscaled 218 yscaled 24 shifted (155,44) withcolor Light;
draw fullcircle xscaled 218 yscaled 24 shifted (155,44) withcolor Grey;
handle:=(221,128)..(270,134)..(269,86)..(219,79);
draw handle withpen pencircle scaled 13 withcolor Blue;
draw handle withpen pencircle scaled 7 withcolor white;
cup:=(88,133)--(101,63)..(155,47)..(210,63)--(224,133)--cycle;
fill cup withcolor .87[Blue,white];
draw cup withpen pencircle scaled 1.5 withcolor Blue;
fill fullcircle xscaled 136 yscaled 33 shifted (156,133) withcolor white;
draw fullcircle xscaled 136 yscaled 33 shifted (156,133) withcolor Blue;
fill fullcircle xscaled 119 yscaled 21 shifted (156,131) withcolor .35[Orange,Grey];
draw (110,119)--(116,77) withpen pencircle scaled 4 withcolor white;
for x=125 step 31 until 187:
  draw (x,157)..(x-6,170)..(x+5,184)..(x,196)
    withpen pencircle scaled 1.5 withcolor Teal;
endfor;
```

Actual output



MPT-15-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`xscaled / yscaled`	Flatten a circle into a rim, coffee surface or saucer.
`Two handle strokes`	Place a narrow white stroke over a broad coloured stroke.
`Layer order`	Put the handle behind the cup and the rim above the body.

MPT-15-005

Build a coastal scene in depth layers

Compose an illustration from back to front: sky, beacon, water, distant headland, cliff and lighthouse. Clip the coloured bands to the tapering tower so one silhouette controls its entire facade.

Compare with MPT-06-005 · Change the stacking order of pictures.

MetaPost drawing

```
path tower,cliff;
picture bands;
fill unitsquare xscaled 280 yscaled 164 shifted (20,27) withcolor Wash;
fill (219,162)--(31,184)--(31,128)--cycle withcolor .83[Orange,white];
fill unitsquare xscaled 280 yscaled 67 shifted (20,27) withcolor .7[Teal,white];
fill (20,94)--(58,114)--(102,101)--(146,94)--cycle withcolor .7[Blue,white];
for y=39 step 14 until 83:
  draw (32,y)--(116,y) withpen pencircle scaled 1.2 withcolor white;
endfor;
cliff:=(145,27)--(157,63)--(190,75)--(230,69)--(253,46)--(300,27)--cycle;
fill cliff withcolor Grey;
tower:=(193,68)--(203,148)--(232,148)--(242,68)--cycle;
fill tower withcolor white;
bands:=image(
  for y=87 step 30 until 117:
    fill unitsquare xscaled 75 yscaled 14 shifted (180,y) withcolor Blue;
  endfor;
);
clip bands to tower; draw bands; draw tower withcolor Blue;
fill unitsquare xscaled 39 yscaled 23 shifted (198,148) withcolor .35[Orange,white];
draw unitsquare xscaled 39 yscaled 23 shifted (198,148) withcolor Blue;
draw (217.5,148)--(217.5,171) withcolor Blue;
fill (193,171)--(217.5,190)--(242,171)--cycle withcolor Blue;
draw (191,147)--(244,147) withpen pencircle scaled 2 withcolor Blue;
fill unitsquare xscaled 10 yscaled 18 shifted (213,68) withcolor Blue;
```

Actual output



MPT-15-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Back-to-front painting`	Let later opaque shapes cover the layers behind them.
`Tapered tower`	Use a polygon as the shared boundary for facade and stripes.
`clip bands to tower`	Keep facade decoration aligned with the tower edges.

MPT-15-006

Project a small isometric house

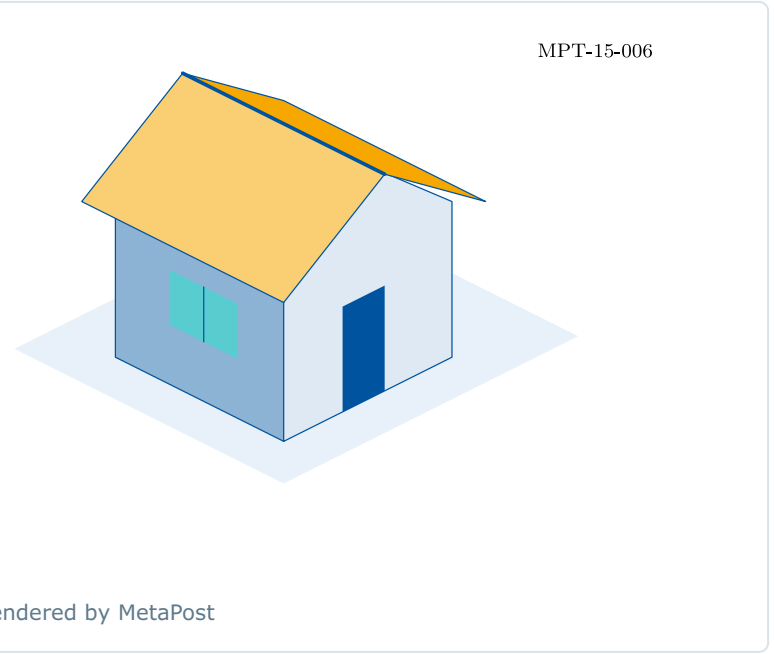
Map three model coordinates to the drawing plane with a reusable projection. The two walls and roof planes share projected vertices. Different opaque tints suggest light; the construction uses no perspective foreshortening or hidden-surface solver.

Compare with MPT-11-006 · Project a wireframe surface.

MetaPost drawing

```
vardef iso(expr x,y,z)=(160+34(x-y),52+17(x+y)+37z) enddef;
path front,side,roofleft,roofright;
front:=iso(0,0,0)--iso(2,0,0)--iso(2,0,1.7)
  --iso(1,0,2.55)--iso(0,0,1.7)--cycle;
side:=iso(0,0,0)--iso(0,2,0)--iso(0,2,1.7)--iso(0,0,1.7)--cycle;
roofleft:=iso(-.2,-.2,1.7)--iso(-.2,2.2,1.7)
  --iso(1,2.2,2.55)--iso(1,-.2,2.55)--cycle;
roofright:=iso(1,-.2,2.55)--iso(1,2.2,2.55)
  --iso(2.2,2.2,1.7)--iso(2.2,-.2,1.7)--cycle;
fill iso(-.5,-.5,0)--iso(3,-.5,0)--iso(3,2.7,0)
  --iso(-.5,2.7,0)--cycle withcolor Wash;
fill side withcolor .55[Blue,white]; draw side withcolor Blue;
fill front withcolor .87[Blue,white]; draw front withcolor Blue;
fill iso(.7,0,0)--iso(1.2,0,0)--iso(1.2,0,1.15)
  --iso(.7,0,1.15)--cycle withcolor Blue;
fill iso(0,.55,.65)--iso(0,1.35,.65)--iso(0,1.35,1.25)
  --iso(0,.55,1.25)--cycle withcolor .35[Teal,white];
draw iso(0,.95,.65)--iso(0,.95,1.25) withcolor Blue;
fill roofleft withcolor .45[Orange,white]; draw roofleft withcolor Blue;
fill roofright withcolor Orange; draw roofright withcolor Blue;
draw iso(1,-.2,2.55)--iso(1,2.2,2.55)
  withpen pencircle scaled 1.5 withcolor Blue;
```

Actual output



MPT-15-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`iso(x,y,z)`	Use equal projected x and y scales with a separate vertical coordinate.
`Shared projected vertices`	Keep adjacent wall and roof faces joined without gaps.
`Face tints`	Assign a fixed light direction by colouring each plane consistently.

16 · Stroke styles and visual systems

Style the same geometry with caps, joins, variable widths, ink treatments, shadows, palette roles and road casings.

MPT-16-001

Compare line caps and corner joins

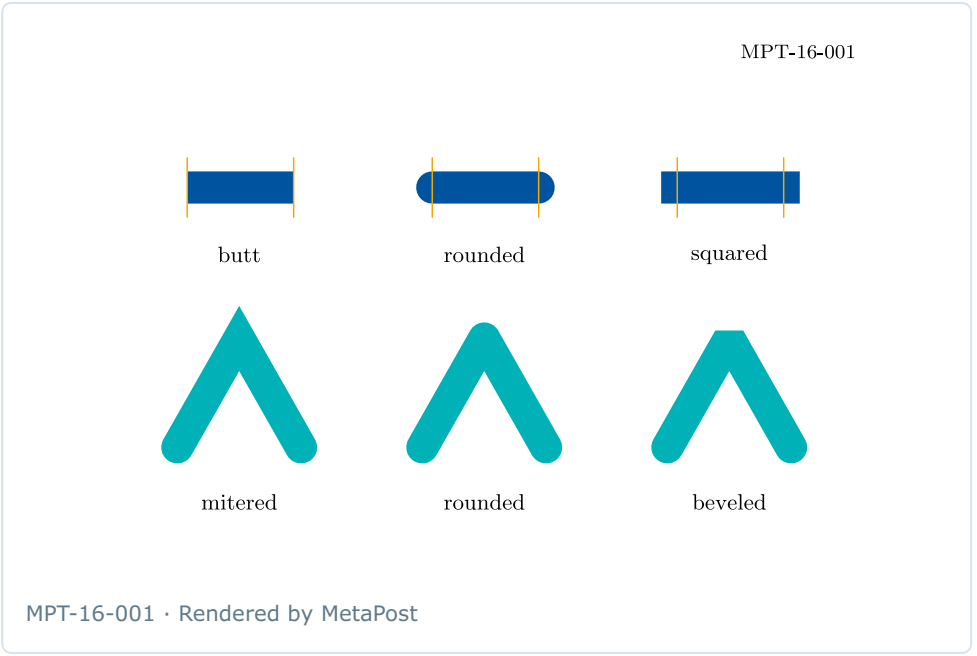
Compare open stroke ends above and corner joins below. Orange ticks mark the endpoints; local groups isolate each setting.

Compare with MPT-04-002 · Compare round and square pens.

MetaPost drawing

```
string capname[],joinname[];
capname[0]:="butt"; capname[1]:="rounded"; capname[2]:="squared";
joinname[0]:="mitered"; joinname[1]:="rounded"; joinname[2]:="beveled";
for k=0 upto 2:
  begingroup
    interim linecap:=k;
    draw (39+99k,155)--(82+99k,155)
      withpen pencircle scaled 13 withcolor Blue;
  endgroup;
  for x=39+99k,82+99k:
    draw (x,143)--(x,167) withpen pencircle scaled .6 withcolor Orange;
  endfor;
  label(capname[k] infont "cmr9",(60+99k,128));
  begingroup
    interim linejoin:=k;
    draw (35+99k,50)--(60+99k,94)--(85+99k,50)
      withpen pencircle scaled 13 withcolor Teal;
  endgroup;
  label(joinname[k] infont "cmr9",(60+99k,28));
endfor;
```

Actual output



MPT-16-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`linecap / linejoin`	Choose 0, 1 or 2 in the labelled left-to-right order.
`interim`	Limit the setting to the enclosing group.

MPT-16-002

Construct a tapered ribbon around a path

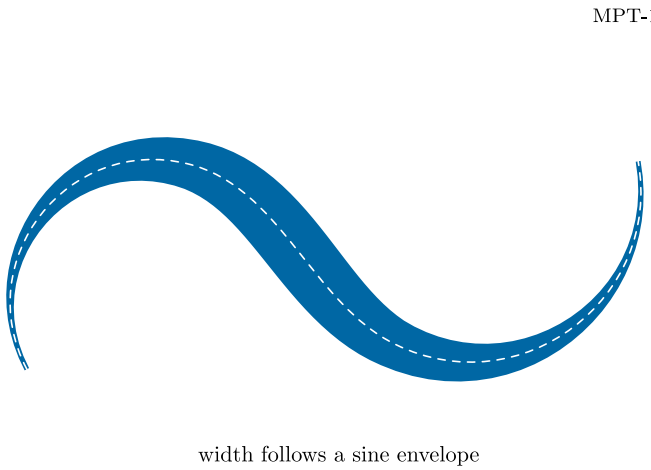
A fixed pen cannot change width along a stroke. Sample a centre line, offset it on each side along its normal and join the resulting polylines into a filled ribbon. This controlled example uses gentle curvature and narrow offsets to avoid self-intersections.

Compare with MPT-15-002 · Grow a feather along a curved spine.

MetaPost drawing

```
path spine,leftedge,rightedge,ribbon;
pair c,n;
numeric w;
spine:=(34,67)..(99,150)..(183,77)..(281,151);
for i=0 upto 80:
  c:=point (3i/80) of spine;
  n:=(unitvector direction (3i/80) of spine) rotated 90;
  w:=1+12*(sind(180i/80)**2);
  if i=0:
    leftedge:=c+w*n; rightedge:=c-w*n;
  else:
    leftedge:=leftedge--(c+w*n);
    rightedge:=rightedge--(c-w*n);
  fi;
endfor;
ribbon:=leftedge--reverse rightedge--cycle;
fill ribbon withcolor .2[Blue,Teal];
draw spine dashed evenly withpen pencircle scaled .65 withcolor white;
label(btex width follows a sine envelope etex scaled .9,(160,32));
```

Actual output



MPT-16-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`point / direction`	Sample location and tangent at the same path time.
`c + w*n / c - w*n`	Offset to opposite sides by a variable half width.
`reverse rightedge`	Walk back along the second edge to close the ribbon.

MPT-16-003

Give one motif three ink treatments

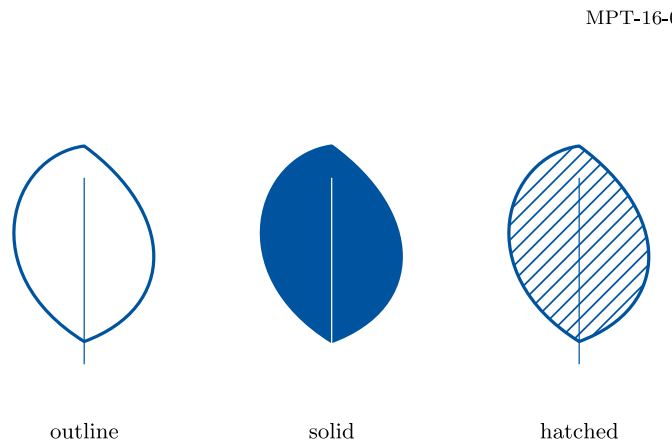
Keep the leaf geometry identical while switching between outline, solid silhouette and clipped hatching. A macro owns the style decision; its local variables let the caller repeat it without leaking temporary paths.

Compare with MPT-15-001 · Draw a leaf from a silhouette and veins.

MetaPost drawing

```
def leafmark(expr anchor,style)=
  begingroup
    save p,h; path p; picture h;
    p:=((0,-36)..controls (-45,-8) and (-30,39)..(0,43)
      ..controls (37,16) and (38,-22)..cycle) shifted anchor;
    if style=1:
      fill p withcolor Blue;
    elseif style=2:
      h:=image(
        for x=-90 step 6 until 60:
          draw ((x,-45)--(x+95,50)) shifted anchor
            withpen pencircle scaled .7 withcolor Blue;
        endfor;
      );
      clip h to p; draw h;
    fi;
    draw p withpen pencircle scaled 1.3 withcolor Blue;
    draw (anchor+(0,-45))--(anchor+(0,30))
      withcolor if style=1: white else: Blue fi;
  endgroup;
enddef;
leafmark((60,115),0); leafmark((160,115),1); leafmark((260,115),2);
label("outline" infont "cmr9",(60,43));
label("solid" infont "cmr9",(160,43));
label("hatched" infont "cmr9",(260,43));
```

Actual output



MPT-16-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`style`	Select a treatment while keeping one shared boundary.
`save p,h`	Make the macro paths and pictures local to each call.
`clip`	Confine parallel ink strokes to the silhouette.

MPT-16-004

Compare flat, offset and long shadows

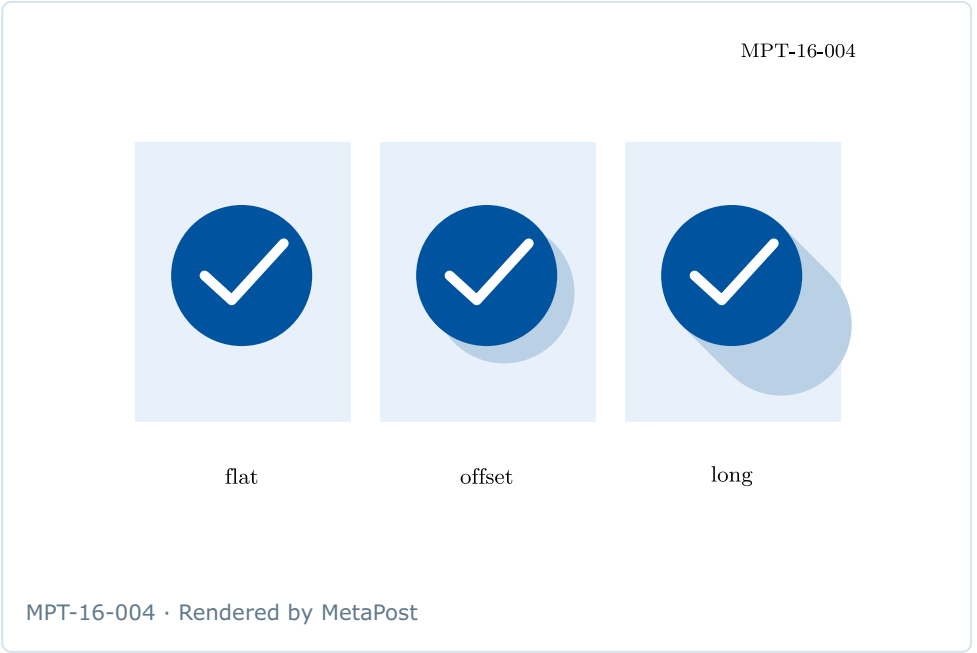
Use the same round badge in three treatments. An offset shadow is one translated silhouette; a long shadow is a finite stack of small translations. Draw the badge and its white mark above the shadow. All colours are opaque.

Compare with MPT-06-005 · Change the stacking order of pictures.

MetaPost drawing

```
path badge,mark;
pair c;
badge:=fullcircle scaled 57;
mark:=(-15,0)--(-4,-10)--(17,13);
for k=0 upto 2:
  c:=(61+99k,119);
  fill unitsquare xscaled 87 yscaled 113 shifted (xpart c-43,60)
    withcolor Wash;
  if k=1:
    fill badge shifted (c+(7,-7)) withcolor .73[Blue,white];
  elseif k=2:
    for d=20 step -1 until 1:
      fill badge shifted (c+(d,-d)) withcolor .73[Blue,white];
    endfor;
  fi;
  fill badge shifted c withcolor Blue;
  draw mark shifted c withpen pencircle scaled 4 withcolor white;
endfor;
label("flat" infont "cmr9",(61,38));
label("offset" infont "cmr9",(160,38));
label("long" infont "cmr9",(259,38));
```

Actual output



MPT-16-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`shifted (7,-7)`	Give a silhouette an explicit offset from its object.
`for d=20 step -1 until 1`	Build a bounded extrusion from back to front.
`Opaque shadow tint`	Mix a colour with white before painting the badge on top.

MPT-16-005

Assign colour roles for light and dark scenes

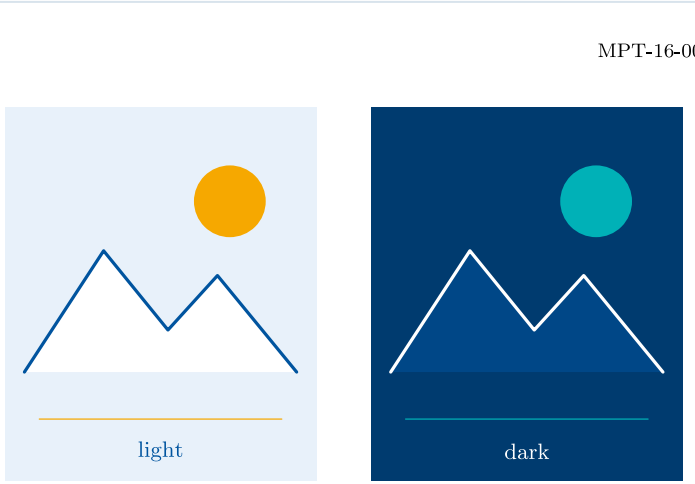
Give background, surface, ink and accent separate roles. The same small landscape uses two explicit palettes. Change the role values at the panel boundary; geometry and line widths remain identical.

Compare with MPT-04-005 · Interpolate colours and tints.

MetaPost drawing

```
color background,surface,ink,accent;
pair c;
for k=0 upto 1:
  c:=(23+148k,36);
  if k=0:
    background:=Wash; surface:=white; ink:=Blue; accent:=Orange;
  else:
    background:=.7Blue; surface:=.85Blue; ink:=white; accent:=Teal;
  fi;
  fill unitsquare xscaled 126 yscaled 151 shifted c withcolor background;
  fill fullcircle scaled 29 shifted (c+(91,113)) withcolor accent;
  fill ((8,44)--(40,93)--(66,61)--(86,83)--(118,44)--cycle)
    shifted c withcolor surface;
  draw ((8,44)--(40,93)--(66,61)--(86,83)--(118,44))
    shifted c withpen pencircle scaled 1.3 withcolor ink;
  draw ((14,25)--(112,25)) shifted c withcolor accent;
  label(if k=0: "light" else: "dark" fi infont "cmr9",c+(63,12))
    withcolor ink;
endfor;
```

Actual output



MPT-16-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`background / surface / ink / accent`	Name visual roles independently of particular RGB values.
`Palette branch`	Change all role assignments together for each scene.
`Same geometry`	Compare palette effects without changing positions or scale.

MPT-16-006

Draw a map with cased roads and a bridge

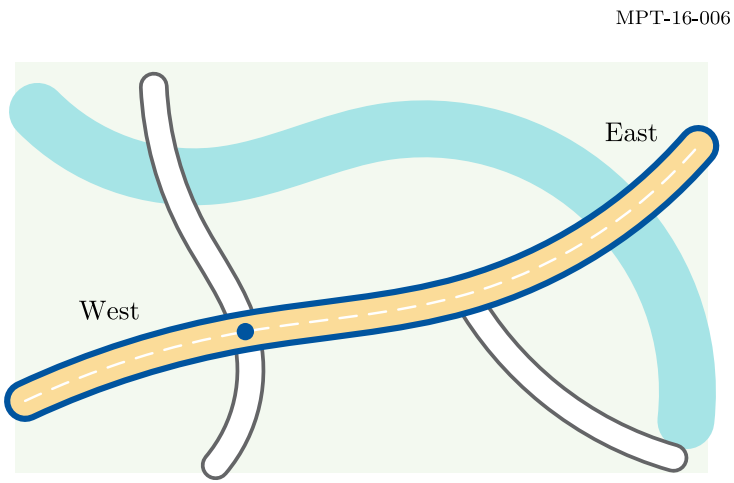
A road casing is a wide outer stroke covered by a narrower inner stroke. Paint all local-road casings before their interiors to join the junction cleanly. The main road crosses the river last, so its casing also reads as a bridge. This is an invented map.

Compare with MPT-16-001 · Compare line caps and corner joins.

MetaPost drawing

```
path river,mainroad,localroad[];
river:=(29,172)..(101,146)..(171,164)..(291,47);
mainroad:=(24,55)..(113,83)..(202,98)..(296,158);
localroad[1]:=(76,182)..(92,124)..(113,83)..(101,29);
localroad[2]:=(202,98)..(236,58)..(286,32);
fill unitsquare xscaled 280 yscaled 166 shifted (20,26) withcolor .93[Green,white];
draw river withpen pencircle scaled 23 withcolor .65[Teal,white];
begingroup
  interim linecap:=rounded; interim linejoin:=rounded;
  for k=1 upto 2:
    draw localroad[k] withpen pencircle scaled 12 withcolor Grey;
  endfor;
  for k=1 upto 2:
    draw localroad[k] withpen pencircle scaled 9 withcolor white;
  endfor;
  draw mainroad withpen pencircle scaled 17 withcolor Blue;
  draw mainroad withpen pencircle scaled 12 withcolor .6[Orange,white];
  draw mainroad dashed dashpattern(on 7 off 5)
    withpen pencircle scaled 1 withcolor white;
endgroup;
fill fullcircle scaled 7 shifted (113,83) withcolor Blue;
label.top("West",(58,85)); label.top("East",(269,157));
```

Actual output



MPT-16-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Casing then interior`	Use two widths on one path to make a bordered road.
`Two passes over local roads`	Cover all casing ends with the joined inner strokes.
`Road after river`	Keep the crossing above the water by drawing order.

17 · Textures, ornament and lettering

Make vector textures, ornament and circular lettering with bounded loops and the installed TeX fonts.

MPT-17-001

Shade three discs with clipped hatching

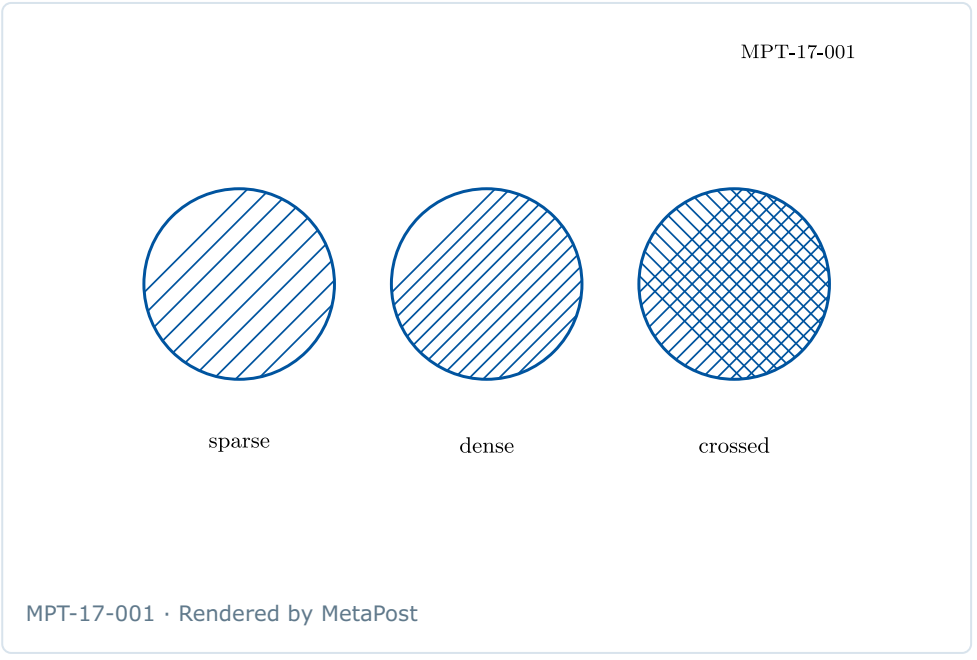
Vary hatch density and direction inside clipped discs. The third disc adds a second line family for crosshatching.

Compare with MPT-16-003 · Give one motif three ink treatments.

MetaPost drawing

```
path disc;
picture hatch;
numeric spacing;
pair c;
for k=0 upto 2:
  c:=(60+100k,116);
  disc:=fullcircle scaled 77 shifted c;
  spacing:=if k=0: 9 else: 6 fi;
  hatch:=image(
    for x=-90 step spacing until 90:
      draw ((x,-55)--(x+110,55)) shifted c
        withpen pencircle scaled .7 withcolor Blue;
      if k=2:
        draw ((x,55)--(x+110,-55)) shifted c
          withpen pencircle scaled .7 withcolor Blue;
      fi;
    endfor;
  );
  clip hatch to disc; draw hatch;
  draw disc withpen pencircle scaled 1.2 withcolor Blue;
endfor;
label("sparse" infont "cmr9",(60,51));
label("dense" infont "cmr9",(160,51));
label("crossed" infont "cmr9",(260,51));
```

Actual output



MPT-17-001 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`spacing`	Set the distance between hatch origins.
`Second line family`	Cross two directions to add ink density.

MPT-17-002

Make a halftone field from variable dots

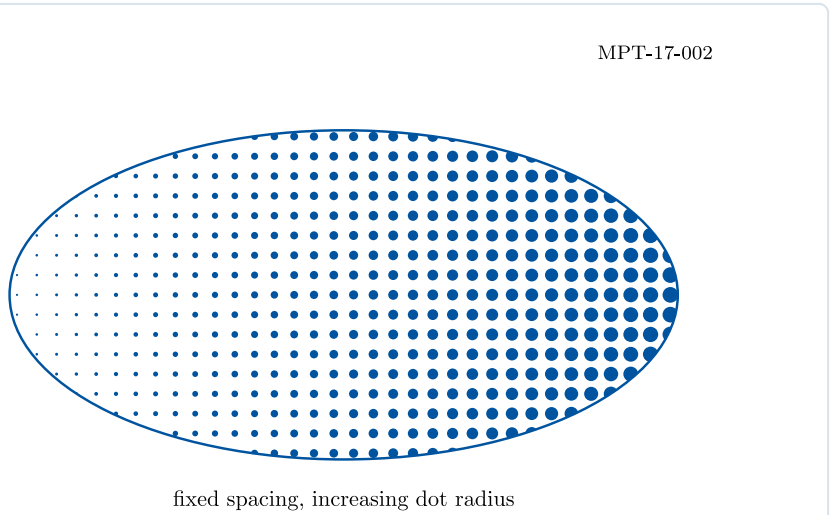
Keep a regular dot lattice and vary only the radius from left to right. Dot radius stays below half the lattice spacing, so dots do not overlap. Clip the field to an oval; no raster image or random sampling is involved.

Compare with MPT-17-001 · Shade three discs with clipped hatching.

MetaPost drawing

```
path oval;
picture dots;
numeric radius;
oval:=fullcircle xscaled 270 yscaled 133 shifted (160,112);
dots:=image(
  for row=0 upto 18:
    for col=0 upto 35:
      radius:=.35+2.9(col/35);
      fill fullcircle scaled (2radius) shifted (20+8col,40+8row)
        withcolor Blue;
    endfor;
  endfor;
);
clip dots to oval; draw dots;
draw oval withpen pencircle scaled 1 withcolor Blue;
label(btex fixed spacing, increasing dot radius etex scaled .9,(160,29));
```

Actual output



MPT-17-002 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`8-unit lattice`	Keep dot centres equally spaced in both directions.
`scaled (2radius)`	MetaPost scales the unit circle by its diameter.
`radius < 4`	Keep neighbouring dots separate at every shade.

MPT-17-003

Engrave a guilloche border with periodic curves

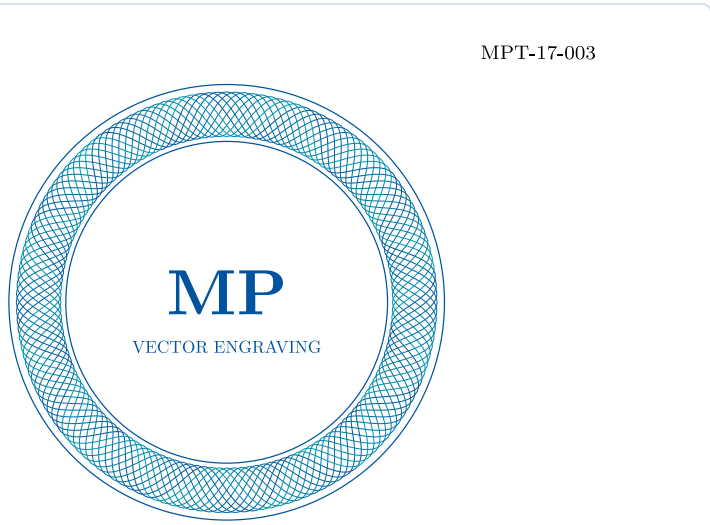
Modulate the radius of a circle with a periodic sine. Twelve phase shifts weave a fine ornamental band around a clear centre. Sampled polylines make the period and amplitude explicit; the first and last samples meet through cycle.

Compare with MPT-07-004 · Reveal a moire field.

MetaPost drawing

```
path p;
numeric a,r;
pair c;
c:=(160,109);
for phase=0 step 30 until 330:
  for j=0 upto 359:
    a:=j; r:=76+9sind(12a+phase);
    if j=0: p:=c+r*dir(a);
    else: p:=p--(c+r*dir(a)); fi;
  endfor;
  draw p--cycle withpen pencircle scaled .35
    withcolor (phase/440)[Blue,Teal];
endfor;
draw fullcircle scaled 130 shifted c withcolor Blue;
draw fullcircle scaled 176 shifted c withcolor Blue;
label("MP" infont "cmbx10" scaled 2.6,c+(0,4)) withcolor Blue;
label(btex VECTOR ENGRAVING etex scaled .7,c+(0,-18)) withcolor Blue;
```

Actual output



MPT-17-003 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`76 + 9sind(12a+phase)`	Set mean radius, radial amplitude and twelve oscillations per turn.
`phase`	Shift the oscillation while leaving the base circle fixed.
`.35-unit pen`	Keep overlapping lines fine enough for a legible ornamental band.

MPT-17-004

Alternate over and under in a woven band

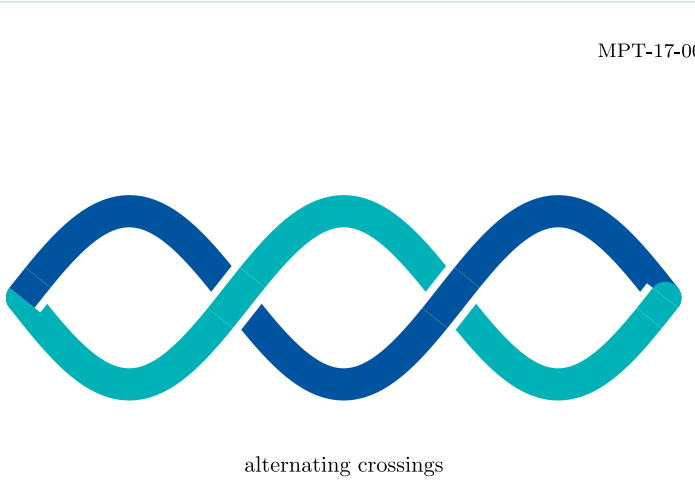
Two sinusoidal strands meet at four known crossings. Draw both strands, then redraw a short section of the selected upper strand at each crossing with a white casing. Alternating that selection creates the woven order without a general intersection solver.

Compare with MPT-16-006 · Draw a map with cased roads and a bridge.

MetaPost drawing

```
path upper,lower,bridge;
upper:=(30,110); lower:=(30,110);
for j=1 upto 120:
  upper:=upper--(30+260j/120,110+35sind(540j/120));
  lower:=lower--(30+260j/120,110-35sind(540j/120));
endfor;
draw upper withpen pencircle scaled 13 withcolor Blue;
draw lower withpen pencircle scaled 13 withcolor Teal;
for k=0 upto 3:
  bridge:=subpath (max(0,40k-3),min(120,40k+3)) of
    if (k mod 2)=0: upper else: lower fi;
  begingroup
    interim linecap:=butt;
    draw bridge withpen pencircle scaled 19 withcolor white;
    draw bridge withpen pencircle scaled 13
      withcolor if (k mod 2)=0: Blue else: Teal fi;
  endgroup;
endfor;
label(btex alternating crossings etex scaled .9,(160,42));
```

Actual output



MPT-17-004 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`Opposite sine offsets`	Construct crossings at sample times 0, 40, 80 and 120.
`k mod 2`	Alternate which strand passes above the other.
`White casing`	Clear the lower strand; butt caps keep the bridge joined.

MPT-17-005

Tile a field with nested wave arcs

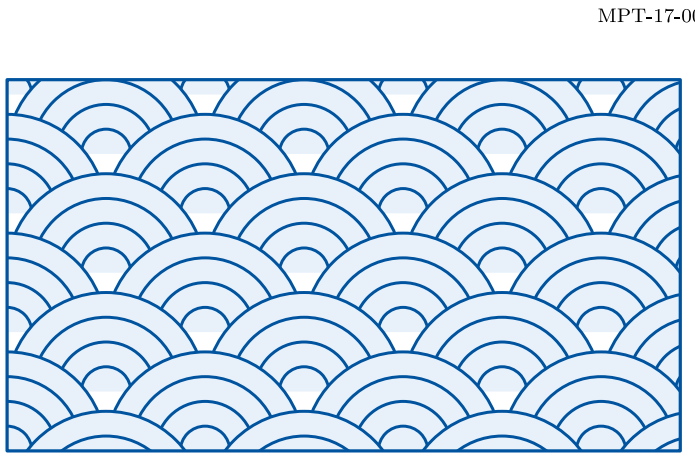
Repeat concentric upper semicircles on staggered rows. Paint a background semicircle before each group so the next row hides the lower parts of the row behind it. Clip the complete pattern to a rectangular swatch and keep the border separate.

Compare with MPT-07-001 · Repeat rotated quarter-circle tiles.

MetaPost drawing

```
picture waves;
path swatch;
pair c;
swatch:=unitsquare xscaled 272 yscaled 150 shifted (24,34);
waves:=image(
  for row=7 downto -1:
    for col=-1 upto 4:
      c:=(24+80col+40(row mod 2),34+24row);
      fill (halfcircle scaled 80--cycle) shifted c withcolor Wash;
      for diameter=20 step 20 until 80:
        draw halfcircle scaled diameter shifted c
          withpen pencircle scaled 1.2 withcolor Blue;
      endfor;
    endfor;
  endfor;
);
clip waves to swatch; draw waves;
draw swatch withpen pencircle scaled 1.3 withcolor Blue;
```

Actual output



MPT-17-005 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`halfcircle`	Reuse the standard upper semicircle at four diameters.
`40(row mod 2)`	Shift alternate rows by half the 80-unit horizontal repeat.
`row=7 downto -1`	Paint the higher rows first so lower waves overlap them.

MPT-17-006

Place lettering around an illustrated seal

Split a heading into individual glyph pictures and place them at equal angular steps. Rotate each glyph to the local tangent of the circular baseline. Leave a blank step for spaces and use TeX for the lower caption. This uses fixed angular spacing rather than automatic optical kerning; short uppercase text suits the method.

Compare with MPT-08-006 · Rotate a label along a curve.

MetaPost drawing

```
string heading;
pair c;
numeric a;
path leaf;
c:=(160,106); heading:="METAPOST STUDIO";
draw fullcircle scaled 173 shifted c withpen pencircle scaled 1.5 withcolor Blue;
draw fullcircle scaled 162 shifted c withpen pencircle scaled .6 withcolor Blue;
for k=0 upto length heading-1:
  a:=160-10k;
  if substring (k,k+1) of heading <> " ":
    label((substring (k,k+1) of heading infont "cmbx10") rotated (a-90),
      c+67dir(a)) withcolor Blue;
  fi;
endfor;
draw (c+(-20,-27))..c..(c+(17,31)) withpen pencircle scaled 1.5 withcolor Blue;
leaf:=(0,0)..controls (-5,16) and (13,24)..(22,22)
  ..controls (24,8) and (9,0)..cycle;
for k=0 upto 2:
  fill leaf rotated (20+12k) shifted (c+(-10+8k,-14+17k)) withcolor Teal;
  fill leaf reflectedabout ((0,0),(0,1)) rotated (-15+12k)
    shifted (c+(-10+8k,-14+17k)) withcolor Blue;
endfor;
label(btex DRAW / REFINE / REPEAT etex scaled .7,c+(0,-47)) withcolor Blue;
fill fullcircle scaled 3 shifted (c+(-63,-31)) withcolor Orange;
fill fullcircle scaled 3 shifted (c+(63,-31)) withcolor Orange;
```

Actual output



MPT-17-006 · Rendered by MetaPost

MetaPost · exit 0

Command or parameter	Meaning
`substring (k,k+1) of heading`	Extract one character as a separate label picture.
`c + 67dir(a)`	Place glyph centres on a circular baseline.
`rotated (a-90)`	Orient each glyph along the clockwise tangent.
`Blank step / btex ... etex`	Reserve a position without drawing a space glyph; let TeX set word spacing in the caption.

Example and command index

- MPT-01-001 · Join two coordinate pairs
- MPT-01-002 · Make a polyline
- MPT-01-003 · Close and fill a shape
- MPT-01-004 · See the coordinate grid
- MPT-01-005 · Use centimetres and millimetres
- MPT-01-006 · Interpolate between pairs
- MPT-02-001 · Let a smooth path pass through knots
- MPT-02-002 · Specify endpoint directions
- MPT-02-003 · Expose cubic control handles
- MPT-02-004 · Compare path tension
- MPT-02-005 · Place dots by arc length
- MPT-02-006 · Inspect tangents on a closed path
- MPT-03-001 · Solve a triangle from relationships
- MPT-03-002 · Construct medians and a centroid
- MPT-03-003 · Compute the chord of two ellipses
- MPT-03-004 · Solve perpendicular bisectors
- MPT-03-005 · Project onto a line
- MPT-03-006 · Split a curve at its intersection time
- MPT-04-001 · Change pen width
- MPT-04-002 · Compare round and square pens
- MPT-04-003 · Use a broad elliptical pen
- MPT-04-004 · Define dash patterns
- MPT-04-005 · Interpolate colours and tints
- MPT-04-006 · Draw a broad-nib flourish
- MPT-05-001 · Translate a stored motif
- MPT-05-002 · Rotate copies
- MPT-05-003 · Scale each coordinate independently
- MPT-05-004 · See why transformation order matters
- MPT-05-005 · Reflect an asymmetric shape
- MPT-05-006 · Name and reuse an affine transform
- MPT-06-001 · Clip a wave field to a circle
- MPT-06-002 · Reuse the field with another mask
- MPT-06-003 · Build an emblem with opaque layers
- MPT-06-004 · Make an annular clipping region
- MPT-06-005 · Change the stacking order of pictures
- MPT-06-006 · Crop a copy for a magnified inset
- MPT-07-001 · Repeat rotated quarter-circle tiles
- MPT-07-002 · Generate a rosette from one petal
- MPT-07-003 · Warp a connected mesh
- MPT-07-004 · Reveal a moire field
- MPT-07-005 · Make curved envelopes from straight segments
- MPT-07-006 · Grow a bounded recursive tree
- MPT-08-001 · Position labels around an anchor
- MPT-08-002 · Typeset mathematics with TeX
- MPT-08-003 · Measure and frame a label
- MPT-08-004 · Connect nodes at their boundaries
- MPT-08-005 · Annotate a measured dimension
- MPT-08-006 · Rotate a label along a curve
- MPT-09-001 · Sample a sine function
- MPT-09-002 · Compare two functions
- MPT-09-003 · Compare knots with a smooth interpolation
- MPT-09-004 · Fill between two envelopes
- MPT-09-005 · Draw a parametric Lissajous curve
- MPT-09-006 · Fill an area and estimate its integral

- MPT-10-001 · Build a labelled bar chart
- MPT-10-002 · Fit a line to a scatter plot
- MPT-10-003 · Bin raw observations into a histogram
- MPT-10-004 · Calculate a box plot from observations
- MPT-10-005 · Draw an empirical cumulative distribution
- MPT-10-006 · Construct an empirical violin
- MPT-11-001 · Build a segmented annulus
- MPT-11-002 · Turn values into pie-sector angles
- MPT-11-003 · Draw a polar flower
- MPT-11-004 · Compare values on a radar polygon
- MPT-11-005 · Render a rotational vector field
- MPT-11-006 · Project a wireframe surface
- MPT-12-001 · Let graph.mp choose coordinate ranges
- MPT-12-002 · Set explicit graph ranges
- MPT-12-003 · Use a logarithmic graph axis
- MPT-12-004 · Fill data in graph coordinates
- MPT-12-005 · Add an automatic graph grid
- MPT-12-006 · Compose two data series and a legend
- MPT-13-001 · Plot a two-column file with graph.mp
- MPT-13-002 · Select columns from a wider table
- MPT-13-003 · Read pairs and skip an empty line
- MPT-13-004 · Leave a gap for a missing observation
- MPT-13-005 · Draw asymmetric intervals from columns
- MPT-13-006 · Compute a centred moving mean
- MPT-14-001 · Give two panels the same scale
- MPT-14-002 · Align observations and residuals
- MPT-14-003 · Reuse panel pictures without rescaling text
- MPT-14-004 · Normalise two series to a shared baseline
- MPT-14-005 · Make the legend survive greyscale printing
- MPT-14-006 · Assemble a titled figure with an audit trail
- MPT-15-001 · Draw a leaf from a silhouette and veins
- MPT-15-002 · Grow a feather along a curved spine
- MPT-15-003 · Clip a scale texture inside a fish
- MPT-15-004 · Assemble a cup with a handle and steam
- MPT-15-005 · Build a coastal scene in depth layers
- MPT-15-006 · Project a small isometric house
- MPT-16-001 · Compare line caps and corner joins
- MPT-16-002 · Construct a tapered ribbon around a path
- MPT-16-003 · Give one motif three ink treatments
- MPT-16-004 · Compare flat, offset and long shadows
- MPT-16-005 · Assign colour roles for light and dark scenes
- MPT-16-006 · Draw a map with cased roads and a bridge
- MPT-17-001 · Shade three discs with clipped hatching
- MPT-17-002 · Make a halftone field from variable dots
- MPT-17-003 · Engrave a guilloche border with periodic curves
- MPT-17-004 · Alternate over and under in a woven band
- MPT-17-005 · Tile a field with nested wave arcs
- MPT-17-006 · Place lettering around an illustrated seal

Content: CC0 1.0. Structure and publishing framework: CC BY 4.0 by Lukas C. Bossert. Repository source · Publishing framework source · License scope and third-party notices.