Folding in R

Origami, Manifolds, and Dimension Reduction

R. Heller

2026-11-10

Contents

Preface **15**
What this book is not . . . 15
Companion package . . . 15

Impressum **17**

Acknowledgments **19**

How to use this book **21**
Standing methodological rules . . . 21
What you need . . . 21
How the chapters are shaped . . . 22
Callouts . . . 22
How to read it . . . 23

I Folding **25**

1 Introduction **27**
1.1 What this chapter answers . . . 27
1.2 The argument this book joins . . . 27
1.3 The answer key . . . 29
1.4 What is claimed, and what is not . . . 31
1.5 How the book is organised . . . 33
1.6 Further reading . . . 33

2 The geometry of folding **35**
2.1 What this chapter answers . . . 35
2.2 Setup . . . 35
2.3 Background . . . 36
2.3.1 What a crease pattern is, and what it is not . . . 36
2.3.2 Rigid folding . . . 36
2.3.3 Two kinds of isometry . . . 36
2.3.4 Curvature, and where it is not . . . 37
2.4 The isometry, in five steps . . . 37
2.4.1 One: the folding is written down, not solved for . . . 37
2.4.2 Two: the isometry is exact, not converged . . . 38
2.4.3 Three: what folding does to distance along the surface . . . 40

2.4.4 Four: what folding does to distance through space 40
2.4.5 Five: the floor under “strictly” 40
2.5 Results 41
2.6 Diagnostics 42
2.6.1 An isometry test cannot tell a fold from a pleat 42
2.6.2 A test that a correct implementation passes is not a test a wrong one fails 43
2.6.3 A first-order flex at the flat state proves nothing 43
2.7 Where it fails 43
2.7.1 The chart is the geodesic only inside a convex region . . . 43
2.7.2 Self-intersection 44
2.7.3 Sampling, and what a finite sample of a zero-reach surface is 45
2.7.4 One family 45
2.7.5 What the chapter does not settle 45
2.8 Reproduce this 45
2.9 Further reading 46

3 Folding as a generative model 47
3.1 What this chapter answers 47
3.2 Setup 47
3.3 Background 48
3.4 The Miura-ori in four scalars 48
3.4.1 The derivation, checked 49
3.5 Sampling and noise 52
3.6 Results: what theta does 54
3.7 Diagnostics 57
3.8 Where it fails 57
3.9 Reproduce this 58
3.10 Further reading 59

II Methods under test 61

4 Linear projections and where they break 63
4.1 What this chapter answers 63
4.2 Setup 64
4.3 Background 64
4.3.1 The projection 64
4.3.2 The distance version 65
4.3.3 What is being scored 65
4.4 A shadow, and two names for it 65
4.4.1 Exact on the flat sheet 65
4.4.2 What the projection discards 66
4.4.3 One computation, four names 67
4.5 Results 67
4.6 Diagnostics 70
4.7 Where it fails 71
4.8 Reproduce this 71
4.9 Further reading 72

5 Geodesic methods **73**
5.1 What this chapter answers . . . 73
5.2 Setup . . . 74
5.3 Background . . . 74
5.3.1 Isomap, in three stages . . . 74
5.3.2 Diffusion maps, and why it is filed here . . . 75
5.4 The geodesic you do not have to estimate . . . 75
5.4.1 The claim, and its one condition . . . 75
5.4.2 The decomposition . . . 76
5.5 Results . . . 77
5.6 When does the graph bridge? . . . 79
5.7 Diagnostics . . . 81
5.8 Where it fails . . . 83
5.9 Reproduce this . . . 85
5.10 Further reading . . . 85

6 Neighbour embeddings **87**
6.1 What this chapter answers . . . 87
6.2 Setup . . . 87
6.3 Background . . . 88
6.4 What the methods produce, and what it costs . . . 89
6.4.1 The deception . . . 90
6.5 Results . . . 91
6.5.1 Neighbourhood size, and where Isomap breaks . . . 93
6.5.2 Do they actually disagree? . . . 94
6.6 Diagnostics . . . 94
6.6.1 Two conventions, and which claim needs which . . . 94
6.6.2 What twenty seeds is, and what it is not . . . 95
6.7 Where it fails . . . 96
6.7.1 `preserve.seed`, and a correction to the correction . . . 96
6.7.2 Laplacian eigenmaps declines a whole noise model . . . 97
6.7.3 LLE under outliers stops being a curve . . . 97
6.7.4 What this chapter does not establish . . . 98
6.8 Reproduce this . . . 98
6.9 Further reading . . . 98

7 Learned folds **99**

III The benchmark **101**

8 Building crease-pattern manifolds **103**
8.1 What this chapter answers . . . 103
8.2 Setup . . . 104
8.3 Background . . . 104
8.3.1 Four objects, four verbs . . . 104
8.4 Constructions that keep the answer key . . . 105
8.4.1 One pattern family, and two honest gaps . . . 105
8.4.2 The isometry certificate . . . 107
8.4.3 Products: raising intrinsic dimension without losing truth 108

8.4.4 Lifting to arbitrary ambient dimension . . . 110
8.5 Results . . . 110
8.5.1 The floor bites, and nothing goes under it . . . 110
8.5.2 Excess as a curve . . . 111
8.5.3 Why a benchmark stuck at intrinsic dimension 2 is not a serious benchmark . . . 112
8.6 Diagnostics . . . 113
8.7 Where it fails . . . 113
8.8 Reproduce this . . . 114
8.9 Further reading . . . 115

9 Ground truth and the evaluators 117
9.1 What "correct" means here . . . 117
9.2 Setup . . . 118
9.3 Background . . . 119
9.3.1 The headline metric: normalised Procrustes RMSE . . . 119
9.3.2 The four metrics under audit . . . 120
9.3.3 Three reference geometries . . . 121
9.4 The known-answer test . . . 122
9.4.1 Why the obvious fix fails . . . 124
9.5 Results . . . 125
9.5.1 How optimistic, and in what units . . . 125
9.5.2 Four metrics, one number . . . 128
9.6 Diagnostics . . . 129
9.6.1 Does k change the answer? . . . 129
9.6.2 What to report when you have no answer key . . . 130
9.7 Where it fails . . . 132
9.8 Reproduce this . . . 132
9.9 Further reading . . . 133

10 Benchmark results 135
10.1 What this chapter answers . . . 135
10.2 Setup . . . 135
10.3 What the grid holds . . . 136
10.4 Where the benchmark has resolving power . . . 138
10.5 Results . . . 139
10.6 Diagnostics . . . 141
10.7 Where it fails . . . 143
10.8 The pre-registered selection rule . . . 144
10.9 Reproduce this . . . 145
10.10Further reading . . . 145

11 Crease patterns vs. the Swiss roll 147
11.1 What this chapter answers . . . 147
11.2 Setup . . . 147
11.3 Background . . . 148
11.3.1 The Swiss roll has an exact chart, and it is usually not the one shipped . . . 148
11.3.2 The severed sphere has no answer key at all . . . 150
11.4 Difficulty has two axes, not one . . . 150

11.4.1 The knob has to move the right thing 150
11.4.2 The fix, and what it costs 152
11.4.3 One geometry against a family 153
11.5 Results 153
11.5.1 The head-to-head, matched on separation alone 153
11.5.2 Why that answer is the wrong one 155
11.5.3 The controlled comparison 155
11.6 Diagnostics 156
11.7 Where it fails 158
11.8 Reproduce this 158
11.9 Further reading 158

IV Application 161

12 From benchmark to decision 163

A Notation 165
Symbols 165

B Datasets and codebook 167
Generated data 167
Precomputed results 167
External data 168
Licence, and what may be redistributed 168
Label provenance, and why the comparison is not circular 168

Glossary 171

Colophon 173

About the author 175

Citing this book 177

References 179

List of Figures

1.1 A flat crease pattern and the same sheet folded. Turning the folded sheet changes which facets you can see; it does not change the pattern. 28

2.1 The flat crease pattern and the same sheet folded. Mountain and valley creases are distinguished by colour and by line style together, so the distinction survives greyscale printing. 39
2.2 One pattern at three points of its one-parameter family. The sheet contracts in both directions as it folds; the flat chart behind it does not change at all. 39
2.3 The ambient-to-geodesic distance ratio across a single crease, against the dihedral angle. The curve is $\sin(\rho/2)$; the points are measured from folded coordinates. A flat sheet sits at the right-hand end and contracts nothing. 42
2.4 Minimum clearance between non-adjacent facets, against the folding parameter. The surface approaches itself as it folds and never touches; the dashed line marks the largest fold angle the benchmark grid uses. 44

3.1 The generative family. Rows are two opening angles α; columns are three values of the folding parameter. The chart differs between rows and is identical along each row: θ moves the sheet and never the answer key. 51
3.2 The four metric constants of the folded placement against the folding parameter, for the pattern of ‘gen-setup‘. Lines are the closed form; points are the same constants read back out of the coordinates ‘fold()‘ returns. 51
3.3 The same triangles, drawn from twice. Panel A is the sampler's draw, uniform over area. Panel B is the naive barycentric draw without the square root, which concentrates points at the fan apex of every triangle — that is, at the crease-pattern vertices. . 53
3.4 Four difficulty measures over the benchmark grid's θ sweep, computed on one seed at $n = 800$. All four are monotone and none is linear: the last quarter of the axis does most of the work. . . . 56

3.5 The ambient k-nearest-neighbour graph at the grid's largest fold, drawn on the chart it should have respected. Panel A holds the edges that are also chart neighbours; panel B holds the false ones. The false edges are not scattered over the sheet — they trace the crease lines, and every one of them straddles a fold. 56

4.1 The two-component PCA of the same sheet at three fold angles, coloured by the exact chart coordinate along the corrugation. The grey paths are level sets of that coordinate: an embedding that recovered the chart draws them nested and non-crossing, and one that folded the chart back onto itself draws two of them on top of each other. 66

4.2 PCA and classical MDS on the same ambient cloud. Panel A overplots one cell: filled points are the PCA embedding, open circles the Procrustes-aligned MDS embedding, and every circle is centred on a point. Panel B is the difference in reconstruction error over every cell of the committed grid, in units of machine epsilon. 68

4.3 The reference curve. Two-component PCA's reconstruction error against the chart, over the fold sweep, for the three noise models. The line is the mean over the twenty benchmark seeds and the band is their full range; the band is narrower than the line except at the last fold angle. The floor is zero everywhere on this grid, so the whole of each curve is recoverable structure that the projection did not recover. 69

4.4 The residual a scree plot shows against the error the answer key shows, over the same sweep. Both are dimensionless and both are monotone, but they are not proxies for each other: the scree residual is measured against the ambient cloud and the reconstruction error against the chart, and their ratio is not constant. 70

5.1 Shortest-path distance on the ambient neighbourhood graph against the exact geodesic distance, as a ratio, for a lightly folded sheet and a deeply folded one. The horizontal line at one is exactness. Points above it are the discretisation bias of walking along graph edges; points below it are distances the fold has contracted. A random ten thousand pairs are drawn in each panel. 77

5.2 Isomap and diffusion maps over the fold sweep, with PCA from Chapter 4 as the reference. Panel A is reconstruction error against the chart, panel B is Q_{NX} at K_{QNX} against the chart. Lines are means over the twenty benchmark seeds and bands in panel A are their full range. Lower is better in A, higher in B. . 78

5.3 The analytic onset against the measured one, over a fold sweep deliberately extended past the benchmark's own ceiling — the vertical dashed line at the largest fold angle the grid uses. Panel A is the exact clearance between non-touching facets as the sheet closes, with the chart k-nearest-neighbour radius for four neighbourhood sizes drawn as horizontal lines; each crossing is a predicted onset, marked with an open circle. Panel B is the measured number of fold-jump edges per sample, averaged over the twenty benchmark seeds, with the same predictions marked on the horizontal axis. 81

5.4 Why the optimum moves. Panel A is Isomap's reconstruction error against neighbourhood size at three fold angles, from the committed sweep; the open circles mark each curve's minimum. Panel B is the median relative departure of the shortest-path distance from the exact geodesic, measured at the same three fold angles: positive is the discretisation bias of walking along edges, negative is the fold contracting them, and each curve crosses zero at a smaller neighbourhood size than the last. 83

5.5 What Isomap returns, coloured by the exact chart coordinate along the corrugation, with grey level sets of that coordinate drawn over the points. Panel A is the lightly folded sheet and panel B the deeply folded one. Both recover the chart; the damage in B is a crowding of the level sets, not a tear. 85

6.1 One folded sample, embedded four ways. Colour is the first chart coordinate and the black paths are level sets of the second, so a faithful embedding shows a smooth colour ramp crossed by non-intersecting paths. The Procrustes error against the chart is printed under each panel. The axes carry no numbers because an embedding is defined only up to a similarity and its units mean nothing. 90

6.2 t-SNE on one sample at four perplexities. Left to right the picture loses every visible feature and gains an order of magnitude in accuracy. Colour is the first chart coordinate; the black paths are level sets of the second, and the axes are unnumbered for the same reason as in the previous figure. 90

6.3 Procrustes error against the chart across the fold sweep, no noise. The band is the full spread across the twenty benchmark seeds and is the result; the solid line is its median, drawn to locate the band rather than to summarise it. The dashed grey line in each panel is Isomap's median, for scale. 92

6.4 Procrustes error against neighbourhood size, at three fold angles. Each panel has its own vertical scale, because the levels differ by more than an order of magnitude and the shape of each curve is the point. Isomap has an interior optimum that moves to smaller neighbourhoods as the sheet folds; every neighbour method is still improving at the largest size swept. 93

6.5 Rank preservation against the chart across every neighbourhood size, on one sample. The two methods that consume distance or geodesics are drawn dashed. No neighbour method reaches PCA at any neighbourhood size, and t-SNE is the only curve that falls as the neighbourhood grows. 95

8.1 Two valid flat crease patterns. Only the left one has a certified rigid folding; the right one is drawn here and never folded anywhere in this book. Mountain and valley are distinguished by colour and by line style together. 105

8.2 Worst within-facet distance error across the folding sweep, at six opening angles spanning the range the contract asserts. The dashed line is the contract; every curve sits five orders of magnitude below it and shows no trend in either the folding parameter or the opening angle. 107

8.3 The chart spectrum of one product cell, and the bound it produces. Panel A: the four eigenvalues of the four-dimensional chart, as shares of its variance. Panel B: the bound at each target dimension, as a line, with the rank-d projection of the chart drawn on top of it as crosses. The tail of A is the square of B. . . 109

8.4 Achieved error against target dimension, by what each method consumes, with the irreducible-loss bound as the heavy line and the region below it shaded. Both arms have a four-dimensional chart, so the bound reaches zero at target dimension four. The shaded region is unreachable by any configuration whatsoever. . 111

8.5 Mean excess over the floor at target dimension two, against each family's own difficulty parameter — the folding parameter on the left, the number of turns on the right. The vertical scales differ between panels and the two arms are matched on nothing, so this is a statement about the construction and not about which family is harder. 112

9.1 What the alignment quotients out, and what it does not. All three panels are scored against the same chart. Point colour is the first chart coordinate and the grey paths are its level sets, drawn so the structure survives greyscale printing. 120

9.2 The candidates, Procrustes-aligned to the same answer key and scored against it. Point colour is the first chart coordinate and the grey paths are its level sets, which the exact chart draws as straight nested lines and a wrong chart does not. The wrongness is a function of the fold angle, which is why the audit sweeps it rather than fixing it. 122

9.3 Inversion rate against the folding parameter: the share of seeds in which the evaluator ranks the exact chart below a two-component PCA of the same cloud. Ambient reference, n = 800. Neighbourhood size is encoded by colour and line style together. The panels do not pool over sample size, because the two sample sizes are two different phenomena. 123

9.4 The two candidates scored against ambient distance across the whole fold sweep, at the book's default neighbourhood size and n = 800. All five quantities lie on the unit interval and share one axis, so the comparison is not an artefact of scaling. Candidate is encoded by colour and line style together. 126

9.5 Metric agreement across the benchmark grid. Each point is one scored cell. The two rank metrics in the left panel are computed at different neighbourhood sizes against different reference geometries and still move together; the right panel is the gap this chapter is about. 128

9.6 Where continuity's inversion sets in. The threshold is the smallest fold angle from which the evaluator inverts on at least half of seeds and never recovers. Plotted against the neighbourhood fraction k/n, the two sample sizes fall on one curve; plotted against k alone they would not. Sample size is encoded by colour, shape and line style together. 129

10.1 What happened to every fit, by method and noise condition. Counts are printed on each segment that is not a clean run, so nothing here is carried by colour alone. 137

10.2 Resolving power against the folding parameter, one line per noise condition, on a logarithmic scale. The dotted line is the pre-registered floor below which the selection rule declines to rank anything. 139

10.3 Normalised Procrustes reconstruction error against the folding parameter on the clean sheet, grouped by what each method consumes. The line is the mean over twenty seeds; the band is the full seed range. PCA and classical MDS coincide exactly and plot on top of each other. 140

10.4 Rank by mean reconstruction error against the folding parameter, one panel per noise condition. Rank one is at the top. The leader changes with the fold angle under two conditions and is a different method entirely under the third. 142

10.5 How far apart the metrics place the eight methods, on the clean sheet: the range of the per-method means at each fold angle, on a logarithmic scale. 142

11.1 Position along the spiral measured two ways, on the same sample. The grey diagonal is where the two agree. The Archimedean roll's customary angle chart departs from arc length across the whole sweep; the Euler roll's departs far less, because its parameterisation is unit speed. 149

11.2 The two difficulty axes. Each family sweeps its own parameter; difficulty rises to the right. The shaded band is empty, so over this design there is no setting at which a folded Miura and a classical surface are comparable on both axes at once. 152

11.3 Mean reconstruction error of eight methods on a Swiss roll and on a Miura matched to it in gap-to-spacing ratio. Methods are ordered by their Swiss-roll score, and the connecting segments make the reordering visible: the two linear methods move from joint last to near the front. 154
11.4 How far apart each benchmark holds PCA and Isomap, across its whole sweep and at every seed the grids carry. Difficulty rises to the right. The Miura curve stays near the floor throughout. The Swiss roll separates the two methods widely over a middle window and then stops separating them at all, because past that window Isomap short-circuits and fails as completely as PCA. . . 157

Preface

R. Heller · Date: 2026-07-29 · Last changed: 2026-11-09

A rigid origami crease pattern generates a manifold whose true low-dimensional structure is known exactly, in closed form. Rigid folding bends the sheet along creases only; the facets stay flat. The folded surface is therefore isometric to the flat sheet, and geodesic distances measured along the surface are unchanged by folding. The flat crease pattern is not an approximation of the correct two-dimensional embedding. It *is* the correct embedding — the answer key.

That single property is what this book is built on.

Standard benchmarks for dimension reduction — the Swiss roll, the S-curve, the severed sphere — give one fixed geometry and a visual impression of success. A crease pattern gives an exact reconstruction error and a parameterised family of test problems. The fold angle θ runs continuously from a flat sheet to a fully compressed one, so every method's performance becomes a curve rather than a single number, and no configuration can be cherry-picked.

Chari and Pachter have argued that two-dimensional embeddings of high-dimensional biological data are systematically misleading. The usual rebuttal is anecdotal. On a manifold where the truth is known exactly, the distortion can be *measured* instead of argued about — and, more usefully, the evaluation metrics themselves can be checked against truth. That is the reason this book exists.

What this book is not

It is not a treatment of rigidity theory, computational origami design, or deep generative modelling. It is not a tutorial on `bookdown`, `make`, or project tooling. Where a section could not be tied back to the ground-truth claim above, it was cut rather than kept for completeness.

Companion package

The companion R package foldbenchR is publicly available. This book renders with the helper scripts versioned alongside its chapters, so the package is not required to build the book or inspect its recorded benchmarks.

Impressum

R. Heller · Date: 2026-07-29 · Last changed: 2026-07-29

Folding in R: Origami, Manifolds, and Dimension Reduction

R. Heller
ORCID: 0000-0001-8006-9742

Source repository: https://github.com/r-heller/folding-in-r
Online edition: https://r-heller.github.io/folding-in-r/

Prose is licensed under CC BY 4.0. Code is licensed under the MIT License; see `LICENSE-CODE.md` in the repository.

This is a living document. The version you are reading was rendered on 2026-11-10.

Acknowledgments

R. Heller · Date: 2026-07-29 · Last changed: 2026-07-29

This book stands on two bodies of work that developed independently and rarely cite each other: the mathematics of rigid origami, and the practice of non-linear dimension reduction. Bringing them together is the only original move here; both halves were built by others.

The R package ecosystem this book depends on — `bookdown`, `Rtsne`, `umap`, `igraph`, `vegan`, `torch` and their dependencies — is maintained largely by volunteers. The bibliography in `packages.bib` is generated automatically so that those authors are cited by name rather than thanked in the abstract.

How to use this book

R. Heller · Date: 2026-07-29 · Last changed: 2026-08-31

Standing methodological rules

Three rules apply throughout and are not restated in each chapter.

1. **Every stochastic result is reported across at least 20 seeds, and across more where the spread requires it.** t-SNE and UMAP are seed-dependent; a single run is an anecdote. Twenty is a floor, not a design: the count each method needs is measured, by inverting the 0.02 reportable difference against that method's within-cell spread, and a method needing more gets more. `scripts/measure-seed-budget.R` does the measuring and `data/processed/seed-budget.rds` is its record. Seeds are fixed in `_common.R` as `BENCH_SEEDS` and disclosed with every figure.

   Where an artefact is produced below the floor — a pilot, or an arm whose cost forbids it — the count is named in the text that reports it, and `scripts/check-artefact-producers.R` will not let it pass unnamed.

2. **Every θ-dependent result is reported as a curve, never as a single number.** Fold angle is a continuous difficulty parameter. Reporting one value invites cherry-picking, and hides exactly the regime changes the benchmark exists to expose.

3. **Every figure carries `fig.alt`.** No result is encoded by colour alone.

What you need

R 4.1 or newer and the packages pinned in `renv.lock`. Run `renv::restore()` once after cloning. There is nothing else to install: the pattern constructors, the sampler and the evaluation metrics are plain scripts in `R/`, sourced by `_common.R` when the book renders. The code that produces every result in this book is in the same repository as the book, at the same commit.

The Chapter 10 benchmark grid takes hours to compute. It is precomputed and committed under `data/processed/`; `scripts/run-benchmark-grid.R` is kept as the provenance record rather than run during the build.

How the chapters are shaped

Every body chapter uses the same nine sections, in the same order, with the same anchors. The anchors are namespaced by a short chapter mnemonic — `geom-`, `intro-`, `eval-` — because pandoc slugs headings by their text, and twelve chapters each carrying a heading called "Results" would otherwise collide and silently renumber each other's cross-references.

```
## What this chapter answers  {#<mn>-question}
## Setup                      {#<mn>-setup}
## Background                 {#<mn>-background}
## <free-wording core>         {#<mn>-core}
## Results                    {#<mn>-results}
## Diagnostics                {#<mn>-diagnostics}
## Where it fails             {#<mn>-limits}
## Reproduce this             {#<mn>-reproduce}
## Further reading            {#<mn>-reading}
```

Only the core section is freely worded — it is the one that carries the chapter's actual argument, and calling it "Core" in every chapter would be worse than useless. Everything else is fixed, and `scripts/lint-chapters.R` fails the build on a chapter that is missing a slot, has them out of order, or anchors one to the wrong namespace.

Read it as a contract with you rather than a template for us. *Where it fails* is in every chapter because every method in this book fails somewhere, and a chapter that could not fill that section would not be finished. *Reproduce this* names the artefact and the script that produced it, so any number in the chapter can be traced to the run that generated it.

Callouts

Five callout classes are defined in `style/style.css`, each with a LaTeX fallback so they survive into the PDF.

- `boxinfo` — context a reader can skip without losing the thread.
- `boximportant` — a result that is easy to misread, or a default that will bite.
- `boxquestion` — something to try, with the answer available in the sources.
- `boxreport` — an aside on provenance, attribution or the state of the literature.
- `boxempty` — an unadorned block, no icon and no colour. It carries the suggested citation in Citing this book.

Write them as knitr block chunks:

```
\begin{boximportant}
umap::umap() defaults to preserve.seed = TRUE.
\end{boximportant}
```

Never as a raw `<div class="boxinfo">` and never as a pandoc fenced div (`::: boxinfo`). Those two forms are not what the stylesheet targets: they render as unstyled markup in HTML and disappear entirely in PDF, where the boxes are

LaTeX environments declared in `style/preamble.tex`. That is how a sibling volume shipped eighty unstyled boxes.

How to read it

Part I builds the object. Part II tests four families of method against it. Part III is the benchmark proper and contains the book's main contribution — Chapter 9, where the evaluation metrics themselves are checked against exact truth. Part IV asks whether any of it changes a real decision.

Readers who only want the benchmark can start at Chapter 8 and refer back to Chapter 3 for the generative model.

Part I

Folding

Chapter 1

Introduction

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

Why should a crease pattern tell you anything about a gene expression matrix?

1.1 What this chapter answers

It should not, and this book never argues that it does. A gene expression matrix is not a folded sheet of paper, and no chapter here claims a resemblance between them.

What a crease pattern has is a property no real dataset has. Its correct two-dimensional structure is known exactly, in closed form, before a single point is sampled. Rigid folding bends a flat sheet along its creases while the facets stay flat and unstretched, so the folded surface is isometric to the sheet it came from. The flat pattern is not an approximation to the folded surface's intrinsic geometry. It **is** that geometry.

One thing follows, and the book is built on it: an evaluator can be handed the right answer and asked whether it recognises it. A metric that ranks the correct embedding below a plainly wrong one where the correct embedding is *known* has not earned trust where it is not. That is a necessary condition rather than a resemblance, and it is the only transfer to real data this book asserts.

The pattern in panel A is the same however panel B is turned. What changes is how much of it any one view exposes — and a dimension-reduction method sees something closer to B than to A. The gap between the two is what the rest of this book measures.

1.2 The argument this book joins

Chari and Pachter argued that the two-dimensional embeddings on which single-cell biology has come to depend distort their data badly enough that conclusions drawn from the pictures are unsafe (Chari and Pachter, 2023). The reply from Lause, Berens and Kobak was not that the distortion is small. It was that the

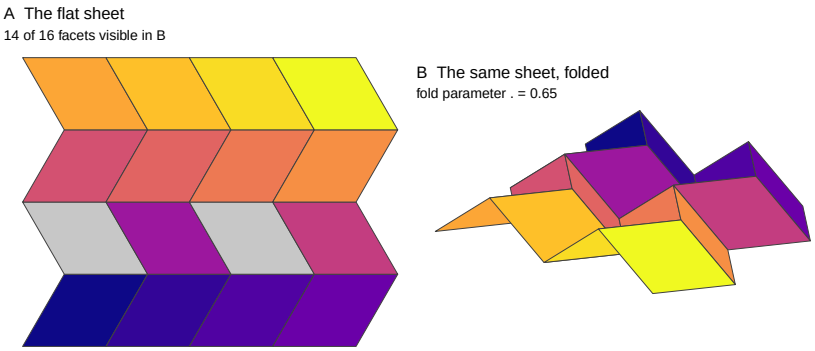


Figure 1.1: A flat crease pattern and the same sheet folded. Turning the folded sheet changes which facets you can see; it does not change the pattern.

case against the embeddings rested on metrics that "focused on preservation of distances, where 2D PCA was unsurprisingly the best" (Lause et al., 2024) — that the disagreement had moved from the embeddings to the instruments used to judge them.

That framing is not this book's, and saying so belongs here rather than in a footnote. It is the published position of one of the two parties to the dispute. What this book adds is not the position but a measurement of it: how far each evaluator sits from an evaluator that has the answer key.

Machado, Behrisch and Telea supply the other half of the reason to look (Machado et al., 2025). They construct projections that score well on trustworthiness and continuity while displaying structure the data does not contain. That is the false-*positive* direction, and it needs no ground truth: to show that a bad embedding scores high, it is enough to know the embedding is bad. The complement — that the *right* embedding scores low — cannot be shown without an answer key. It is rarely shown for that reason, and it is what Chapter 9 does.

```
audit <- read_run("data/processed/evaluator-audit.rds",
 ↪  "d7982cf7eb3afe7d")
prov  <- attr(audit, "provenance")

# knitr rounds an inline numeric to options(digits), which turns
 ↪  4e-16 into 0.
# These return character, so a small number survives the trip
 ↪  into prose.
# trimws() because formatC() pads "fg" output to a common width,
 ↪  and a padded
# number lands in the prose with its leading spaces intact.
fmt <- function(x, d = 3L) trimws(formatC(signif(x, d), format =
 ↪  "fg", digits = d))
sci <- function(x, d = 2L) formatC(x, format = "e", digits = d -
 ↪  1L)
```

```
amb <- audit$inversions[audit$inversions$reference == "ambient" &
                        audit$inversions$n == max(prov$n), ]
inverts    <- tapply(amb$inverted, amb$metric, mean)
trust_span <- range(c(amb$truth[amb$metric == "trust"],
                      amb$pca2[amb$metric == "trust"]))

perfect <- audit$scores[audit$scores$reference == "ambient" &
                        audit$scores$candidate == "truth" &
                        audit$scores$k == K_DEFAULT &
                        audit$scores$n == max(prov$n), ]
ceiling <- aggregate(cbind(knn, qnx) ~ theta, perfect, mean)
```

The design is one sentence. Hand each evaluator the exact chart as though it were a submission, alongside a two-component PCA of the same folded cloud, and record which of the two it prefers. Across the audit's ambient-referenced cells at the larger sample size, continuity ranks the wrong answer above the right one in 41.8 per cent of them. Trustworthiness almost never inverts, and only because it barely moves: over the whole sweep it rates every candidate in the audit, the exact chart and the wrong one alike, at 0.984 or above. At the last fold angle, kNN preservation rates the *exact* unfolding 0.58 and Q_{NX} rates it 0.714. A reader shown the first of those would call the embedding a partial failure. It is the answer.

The law that chapter set out to state does not exist, and this introduction will not promise it. The plan of record specified a threshold in the fold angle: rank metrics are invariant to a monotone contraction of distances until neighbour identity changes, so they should invert only past some critical θ. They do invert, and the mechanism is the one predicted. The threshold is not. It slides with the neighbourhood size across most of the available range (Figure 9.6), so there is no threshold in θ alone. What survives is a threshold surface in θ and the neighbourhood *fraction* k/n, and it survives for continuity only. That is a smaller finding than the one budgeted for it, and Chapter 9 reports it as the smaller thing it is.

None of this arbitrates the dispute above. Espadoto and colleagues survey what dimension-reduction evaluations actually report (Espadoto et al., 2021); the claim in this book is about those reports, and it requires no position on what a two-dimensional embedding is for.

1.3 The answer key

The construction is one pattern family and it is not new. Miura's report describes the tessellation this book folds as "the isometric transfer of an infinite plane subject to biaxial shortening" (Miura, 1985), which is the whole claim in the words of the person who first wrote it down. Chapter 2 derives the folding rather than citing it, because the derivation is what makes the answer key exact instead of approximate.

```
p <- miura_ori(nx = PATTERN_GRID$miura[["nx"]], ny =
 ↪  PATTERN_GRID$miura[["ny"]])
m <- sample_manifold(p, theta = FIG_INTRO_THETA, n = N_DEFAULT,
                     seed = BENCH_SEEDS[1])

iso  <- facet_isometry_error(p, FIG_INTRO_THETA)
pca2 <- stats::prcomp(m$X, rank. = EMBED_DIM)$x[,
 ↪  seq_len(EMBED_DIM)]
err  <- c(chart = reconstruction_error(m$truth, m$truth),
          pca   = reconstruction_error(pca2, m$truth))
qn   <- c(chart = qnx(m$truth, reference_dist(m, "chart"), K =
 ↪  K_QNX),
          pca   = qnx(pca2,    reference_dist(m, "chart"), K =
           ↪  K_QNX))
```

Two numbers fix what "answer key" means here. Folding preserves every pairwise distance within every facet to 4.4e-16 — floating-point noise on a closed form, not a tolerance anyone chose. And handing the chart back as though it were a submission scores 1.7e-16 against itself under the book's headline error, where a two-component PCA of the same folded cloud scores 0.0775.

θ is a parameter, not an angle. It runs on the unit interval: zero is the flat sheet, one is flat-folded, and the dihedral angle at a crease is derived from it rather than being another name for it. Because it is continuous and the geometry moves monotonically along it, every result in this book is a curve rather than a number — standing rule 2, and the reason a benchmark with one fixed geometry cannot produce the observations Part III reports.

Two headline numbers per cell, and the pair is deliberate. The first is a normalised Procrustes RMSE: the isometry-recovery score, admitting translation, rotation, reflection and one isotropic scale, and charging for everything else. The second is Q_{NX} computed with the exact chart as its reference geometry, which is invariant under any diffeomorphism of the plane and therefore neutral between preserving distances and preserving neighbourhoods. On the sample above the two disagree about PCA: an error of 0.0775 against a Q_{NX} of 0.899, where the exact chart scores 1 on the second and 1.7e-16 on the first. The gap between them, as a function of θ, is a measurement of how much observed error is genuine structural loss and how much is harmless reparameterisation. Reporting only the first would build the book's scoring function out of one side of the dispute it claims to measure.

Folding preserves one kind of distance and destroys the other. Distances along the surface are unchanged, exactly; distances through the surrounding space contract by a factor that depends only on the dihedral angle. Every method in Part II consumes the second kind and is scored against the first. That mismatch is not a flaw in the benchmark, it is the benchmark's subject — and it comes with a cost worth knowing before Part III: a crease has reach exactly zero, so the sampling guarantees usually invoked to justify these methods do not apply to this book's manifolds at all — not weakly, not approximately. Chapter 2 says what replaces them.

1.4 What is claimed, and what is not

Most of the methods under test never claimed what the headline metric measures. Isomap consumes geodesic distances, and its assumption is satisfied exactly on a crease pattern (Tenenbaum et al., 2000); it is the one method in Part II trying to do the thing a Procrustes score scores. t-SNE optimises a divergence between neighbour probability distributions (van der Maaten and Hinton, 2008), UMAP a cross-entropy on a fuzzy simplicial set (McInnes et al., 2018), and neither claims to reproduce distances at all — a t-SNE embedding's very extent is a function of its perplexity and its iteration count. Local linear embedding, the Laplacian eigenmap and an autoencoder are in the same position. Scoring them against isometric truth and reporting that they lose is not a finding about them; it is a category error dressed as a measurement. The second headline number exists so that a method preserving neighbourhoods without preserving distances can be seen doing it, and each Part II chapter restates the point for the method it covers.

```
e1  <- read_run("data/processed/e1-controlled.rds",
 ↪  "01d82cfe8f45bb44")
ctl <- local({                          # both difficulty axes
 ↪  inside the joint range
  s <- e1[is.finite(e1$rmse), ]; s$lg <- log(s$gs)
  s[s$lg >= max(tapply(s$lg, s$kind, min)) & s$lg <=
 ↪  min(tapply(s$lg, s$kind, max)) &
    s$np >= max(tapply(s$np, s$kind, min)) & s$np <=
 ↪  min(tapply(s$np, s$kind, max)), ]
})
w <- reshape(ctl[, c("kind", "label", "seed", "method", "rmse")],
             idvar = c("kind", "label", "seed"), timevar =
              ↪  "method",
             direction = "wide")
w$spread <- abs(w$rmse.pca - w$rmse.isomap)
sp    <- tapply(w$spread, w$kind, mean)
ratio <- sp[["classic"]] / sp[["crease"]]

set.seed(11)                            # cluster bootstrap,
 ↪  resampling whole settings
ci <- quantile(replicate(2000, {
  r <- vapply(split(w, w$kind), function(d) {
    l <- sample(unique(d$label), length(unique(d$label)), replace
 ↪  = TRUE)
    mean(unlist(lapply(l, function(x) d$spread[d$label == x])))
  }, numeric(1))
  r[["classic"]] / r[["crease"]]
}), c(0.025, 0.975))

fit   <- lm(spread ~ I(kind == "classic"), data = w)   #
 ↪  cluster-robust by setting
X     <- model.matrix(fit); u <- residuals(fit); bread <-
 ↪  solve(crossprod(X))
```

```
meat  <- Reduce(`+`, lapply(unique(w$label), function(l) {
  i <- w$label == l
  tcrossprod(crossprod(X[i, , drop = FALSE], u[i]))
}))
tval  <- coef(fit)[2] / sqrt((bread %*% meat %*% bread)[2, 2])
pval  <- 2 * pt(-abs(tval), df = length(unique(w$label)) - 2)
```

The Swiss roll discriminates better than this benchmark does. The claim this project set out to make was that a crease pattern is a *harder* test than the classical surfaces. Experiment E1 was run before Chapter 11 was drafted, precisely so that the prose could not outlive a refutation, and it refuted it. Matched on both axes of difficulty — branch separation and ambient non-planarity — over 9 crease settings and 5 Swiss rolls, the mean gap between PCA and Isomap is 0.574 on the classical side against 0.113 on the crease side: a ratio of 5.06, with a cluster bootstrap interval over settings of 1.7 to 9.8 and a cluster-robust p of 0.02. Inside the overlap the rankings are identical — Isomap first on both families, the linear methods behind it on both — so nothing flips. Crease patterns are not a harder benchmark. They are an easier one that tells methods apart about five times less well, and a reader choosing between the two for a straight comparison of methods should choose the roll. E1 pre-registered three outcomes and none of them occurred; the replacement claim is in this book because it fits the evidence, not because its antecedent was met.

The difficulty axis stops short of the failure it was expected to reach. Folding corrupts the neighbourhood graph a method consumes, and Chapter 3 measures the corruption growing along the whole sweep. What it does not find, anywhere on the committed grid, is a single graph edge joining facets two or more cells apart — the sheet-to-sheet short circuit that classically breaks Isomap (Balasubramanian and Schwartz, 2002) lives past the end of the axis, because the sweep stops before the sheet begins collapsing onto itself. This benchmark measures the graded corruption that precedes that failure. It does not reach the failure.

Exactness alone is not the differentiator. An arc-length-charted Swiss roll also has an exact isometric key, and it is published prior art: Schoeneman and colleagues noticed that the customary answer key is not the isometric one and replaced the Archimedean spiral with a unit-speed one (Schoeneman et al., 2017). Claiming exact ground truth as the novelty would be wrong. What a crease pattern has that theirs does not is reach exactly zero rather than positive, so that it sits outside the smoothness theory rather than at its edge; intrinsic dimension above two while the chart stays in closed form, which is what makes an irreducible-loss bound computable at all; and a difficulty parameter whose critical value can be derived rather than found by sweeping and looking.

One family, and two documented negative results. Two further pattern families were specified and both were withdrawn on evidence, which Chapter 8 reports as results. A benchmark built on one family cannot support claims about pattern variety, and none are made.

1.5 How the book is organised

Part I builds the object. Chapter 2 derives the folding and proves what it preserves and what it destroys, including a proposition that an ambient-referenced stress must rank the exact unfolding below a two-component PCA — a theorem rather than a simulation. Chapter 3 turns the surface into data: sampling, noise models, and the four measures of difficulty that the θ axis moves.

Part II tests the methods one family at a time, each against the same answer key. Chapter 4 takes the linear projections, Chapter 5 the geodesic methods and the short-circuit failure they are prone to, Chapter 6 the neighbour embeddings, and Chapter 7 the learned folds — where the honest result is that the autoencoder row is declared unavailable rather than quietly dropped.

Part III is the benchmark. Chapter 8 builds the manifolds and the irreducible-loss bound that says how much of an error was unavoidable. Chapter 9 runs the known-answer test on the evaluators themselves. Chapter 10 reads the full grid, and asks whether the benchmark separates methods before asking which method wins. Chapter 11 puts the construction against the Swiss roll, the S-curve and the severed sphere, and reports that it loses on discrimination.

Part IV spends the result. Chapter 12 uses the grid to choose a method and its settings for a real dataset, under a rule fixed in code before the grid existed, and reports what happens.

Two appendices carry the reference material: A fixes every symbol, and B documents the committed artefacts. Every artefact is read with its digest, so a regenerated run is a build failure rather than a silently changed sentence, and the three standing rules the chapters do not restate are set out in *How to use this book*.

Not every reader wants the whole path. Someone who came for the geometry can read Chapters 2 and 3 and stop; someone who came for the numbers can start at Chapter 10 and work backwards through the two chapters it depends on; and someone deciding whether to use this benchmark at all should read Chapter 11 first, because that is where the case against it is made.

1.6 Further reading

Where to start depends on what you came for. For the origami, Hull's *Origametry* (Hull, 2020) is the single best entry point and covers both the classical flat-foldability theorems this book uses as checks. For the methods and for the rank metrics Part III audits, Lee and Verleysen (Lee and Verleysen, 2007) is the textbook treatment of everything Part II fits.

For the dispute, read Chari and Pachter (Chari and Pachter, 2023) and the reply from Lause, Berens and Kobak (Lause et al., 2024) in that order; they are short, and between them they define the question this book answers with measurement rather than with argument. Espadoto and colleagues (Espadoto et al., 2021) survey what evaluations report in practice, and Machado, Behrisch and Telea (Machado et al., 2025) is the direct complement to Chapter 9. For the

nearest prior benchmark with an exact answer key, Schoeneman and colleagues (Schoeneman et al., 2017) — read alongside Chapter 11, which measures this construction against theirs and reports that theirs separates methods better.

Miura's own report (Miura, 1985) is worth finding for the sentence quoted above, and each chapter closes with a *Further reading* section of its own that goes deeper on its own subject than this one can.

Chapter 2

The geometry of folding

R. Heller · Date: 2026-07-29 · Last changed: 2026-08-31

What does folding preserve, and what does it destroy?

2.1 What this chapter answers

A sheet of paper folded along creases is a surface in space. Flattened out again it is a region of the plane. The two are the same paper, and that is the whole of this book's method: the flat region is not a good approximation to the folded surface's true two-dimensional structure, it *is* that structure, written down in closed form before any data are generated.

This chapter establishes what that sameness does and does not mean. Folding preserves every distance measured **along** the surface and destroys a great many measured **through** the space around it, and almost every difficulty a dimension-reduction method has with a folded sheet is a consequence of that one sentence. A method sees the second kind of distance and is scored against the first.

Nothing here is asserted. Every claim in this chapter is a check in `tests/testthat/`, and every number is computed while the page is built.

2.2 Setup

```
p <- miura_ori(nx = 6L, ny = 6L, alpha = pi / 3)
f <- fold(p, theta = 0.6)
```

`miura_ori()` returns a flat crease pattern; `fold()` places it in space at a folding parameter θ. Both are in `R/`, sourced when this book renders, and the notation appendix fixes the symbols.

2.3 Background

2.3.1 What a crease pattern is, and what it is not

A crease pattern is a planar straight-line graph on a region of paper: vertices, straight creases between them, and the facets those creases enclose. It carries one more piece of information per crease, an **assignment** — mountain or valley — which says which way the paper turns there.

The literature on origami mathematics is largely about which patterns fold and how to design one that folds to a given shape (Lang, 2011; Demaine and O'Rourke, 2007). This book needs almost none of that. It needs one family that provably folds, and it needs the folding written as a map it can evaluate at any θ.

> **Not the Huzita–Hatori axioms.** A reader who has met origami mathematics before will expect the construction axioms — the seven operations that say which points and lines a folder can produce with a sequence of folds. They are not what makes a crease pattern a manifold, and they play no part in this book. They describe what can be *constructed*; the question here is what a given pattern, however obtained, *is* once folded. Hull's *Origametry* [@hull2020origametry] covers both and is the place to start if the constructive side is what you came for.

2.3.2 Rigid folding

The foldings this book uses are **rigid**: the facets stay flat and unstretched, and all the motion happens at the creases. That is a strong restriction and it is the source of everything that follows. A folding that let facets bend would still produce a surface, but its flat pattern would no longer be its unfolding, and the answer key would be gone.

Two classical conditions constrain which flat patterns can fold flat at all. At every interior vertex, **Maekawa's theorem** says the numbers of mountain and valley creases differ by exactly two, and **Kawasaki's theorem** says the alternating sum of the sector angles around the vertex is zero (Hull, 2020). Both are necessary and neither is sufficient. Chapter 8 returns to how little they guarantee; this chapter uses them as checks rather than as foundations, which is the only honest way to use a necessary condition.

The attribution here is tangled and it is better to say so than to pick a side. Kawasaki's condition appears independently in work by Justin and by Murata, and no citation resolves cleanly to a single first statement. *Origametry* is the honest reference until one does.

2.3.3 Two kinds of isometry

The word *isometry* does two jobs and the difference between them is this book's crux.

An **intrinsic** isometry preserves distances measured along the surface — the length of the shortest path an ant could walk without leaving the paper. Rigid

folding is an intrinsic isometry, exactly and by construction: the facets do not stretch, so a path drawn on the flat sheet has the same length after folding.

An **ambient** isometry preserves distances measured through the surrounding space — straight lines through the air. Folding is emphatically *not* an ambient isometry. Bring two points on either side of a crease together by folding, and the straight-line distance between them collapses while the path along the paper does not change at all.

Every method in Part II consumes ambient distance or something built from it. The answer key is intrinsic. That mismatch is not a flaw in the benchmark; it is the benchmark's subject.

2.3.4 Curvature, and where it is not

The natural wrong intuition about a folded sheet is that its curvature has been pushed into the vertices. It is worth writing against rather than merely avoiding, because it is wrong in a way that would undermine the whole construction if it were right.

Discrete Gaussian curvature at a vertex of a polyhedral surface is its angle defect: 2π minus the sum of the sector angles meeting there. For a folded flat sheet those angles are the flat pattern's angles, unchanged by folding, and they sum to exactly 2π because the vertex was an interior point of a flat piece of paper. **The Gaussian curvature is identically zero, at every vertex and on every facet.** Were it otherwise the sheet would not be developable and the flat pattern would not be its unfolding.

What does concentrate on the creases is *extrinsic*: the second fundamental form, which is a measure supported on the one-skeleton rather than a function. The surface is intrinsically flat and extrinsically as bent as one likes.

For manifold learning the consequence that matters is a different quantity entirely. The **reach** of a submanifold (Federer, 1959) is the largest distance within which every point of the ambient space has a unique nearest point on the surface. Sampling theory for manifold learning is generally stated in terms of it (Niyogi et al., 2008), and a crease has reach exactly zero: a point just outside the fold is equidistant from both facets. So the standard sampling guarantees do not apply here at all — not weakly, not approximately. This book's manifolds sit outside the hypotheses of the theory usually invoked to justify the methods being tested on them, which is worth knowing before Part III reports that some of those methods struggle.

2.4 The isometry, in five steps

2.4.1 One: the folding is written down, not solved for

The Miura-ori is a tessellation of congruent parallelograms (Miura, 1985). Its flat vertex (i, j) sits at

$$x = i\,a + (j \bmod 2)\,b\cos\alpha, \qquad y = j\,b\sin\alpha,$$

where a and b are the cell side lengths and α its acute angle. The $(j \bmod 2)$ term is the entire pattern: it makes the transverse crease lines zigzag with period two instead of running straight, which is what turns a plain parallelogram lattice into something that folds.

The folded placement is obtained by *positing* a form and solving for the constants rather than by rotating facets about creases and hoping the result closes. Look for

$$x = iL_x + (j \bmod 2)\, d_x, \qquad y = jL_y, \qquad z = (i \bmod 2)\, H.$$

Two facet side vectors follow immediately, $e_1 = (L_x, 0, \pm H)$ for the horizontal edge of flat length a, and $e_2 = (\pm d_x, L_y, 0)$ for the zigzag edge of flat length b. Because both are shared by opposite sides, every facet is a parallelogram and therefore planar with no further conditions. Isometry then needs only the two edge lengths and the angle between them:

$$L_x^2 + H^2 = a^2, \qquad d_x^2 + L_y^2 = b^2, \qquad L_x d_x = ab\cos\alpha.$$

Three equations in four unknowns: a one-parameter family, which is the rigid folding. Writing $L_x = a\cos\varphi$ and $H = a\sin\varphi$ solves the first identically and leaves $d_x = b\cos\alpha/\cos\varphi$ and $L_y = b\sqrt{1 - \cos^2\alpha/\cos^2\varphi}$.

L_y is real only while $\varphi \le \alpha$, and vanishes exactly at $\varphi = \alpha$, which is the flat-folded state. So φ runs on $[0, \alpha]$ and the natural normalised parameter is $\theta = \varphi/\alpha$.

θ is a parameter, not an angle. It runs on $[0, 1]$: zero is the flat sheet and one is flat-folded. The dihedral angle ρ at a crease is *derived* from it and is never a synonym for it — a distinction this project got wrong once and now states in three places.

2.4.2 Two: the isometry is exact, not converged

Because the placement was derived by imposing edge lengths and facet angles, the isometry holds by construction rather than to a solver's tolerance. That is a claim about algebra, and algebra is exactly the thing that might be wrong, so it is measured.

```
worst_iso <- max(vapply(seq(0, 1, by = 0.05),
                        function(t) facet_isometry_error(p, t),
                         ↪  numeric(1)))
```

`facet_isometry_error()` compares **all pairwise distances** between a facet's vertices, folded against flat, not just its edges: an implementation that preserved every edge of a parallelogram while shearing it would pass an edge test and would not be a rigid folding. Over the whole θ sweep the worst departure on this pattern is 0, which is floating-point noise on a closed form rather than a tolerance anyone chose.

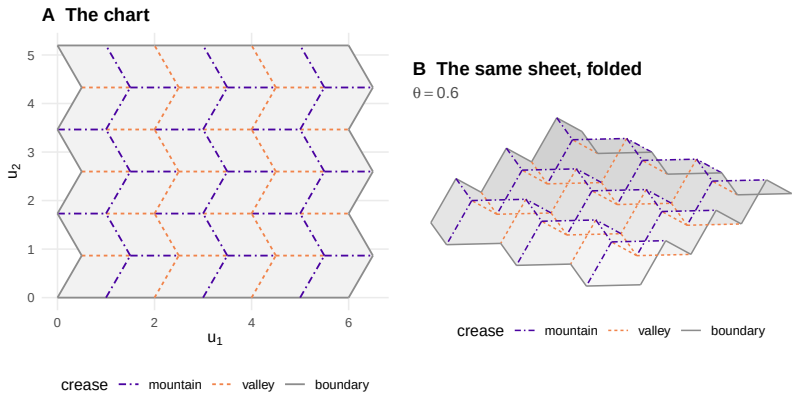


Figure 2.1: The flat crease pattern and the same sheet folded. Mountain and valley creases are distinguished by colour and by line style together, so the distinction survives greyscale printing.

Panel A of Figure 2.1 is the answer key. Panel B is what a dimension-reduction method is shown. Everything in Part III is a measurement of the gap between them.

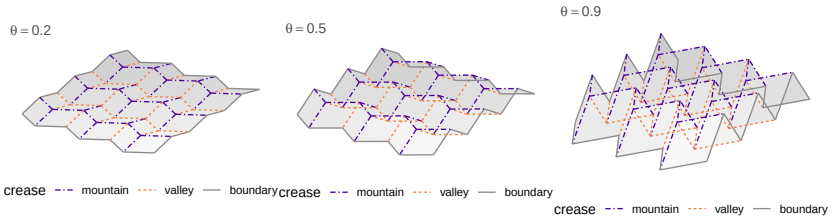


Figure 2.2: One pattern at three points of its one-parameter family. The sheet contracts in both directions as it folds; the flat chart behind it does not change at all.

Figure 2.2 is that family at three values of θ, drawn at one scale. Two things are visible and both matter later. The sheet contracts in **both** directions at once, which is why fold angle is a difficulty parameter rather than a nuisance: it moves the ambient geometry continuously while leaving the chart exactly where it was. And the contraction is not uniform — the far end of the range does most of it, which is why the θ grid in Part III is even in θ and uneven in everything a method cares about.

The derivation above is for one family. Nothing in this book requires a second, and two candidates for one were withdrawn on evidence rather than dropped quietly; Chapter 8 reports both as results. What a reader should take from that is the standard a folding has to meet here: a closed form, an exact isometry, and a fold that reaches every crease family. Two of the three are easy.

2.4.3 Three: what folding does to distance along the surface

Nothing. That is the point, and it needs no further argument once the facets are congruent: a path on the sheet crosses the same facets in the same places at any θ, so its length is the same. Geodesic distance between two sample points is therefore exactly the distance between their positions in the flat chart, subject to one caveat that is stated in `geom-limits` rather than hidden here.

2.4.4 Four: what folding does to distance through space

Here folding does a great deal, and the amount is exact.

Take two points on opposite sides of a single crease, each at perpendicular distance d from it and at the same position along it. Their chart distance is $2d$ at any fold angle. Their ambient distance, once the crease has closed to a dihedral angle ρ, is $2d\sin(\rho/2)$. So

$$\frac{d_A}{d_U} = \sin(\rho/2),$$

which depends on ρ **alone**. Not on d, not on where along the crease the pair sits. That independence is the substance of the claim; a ratio that drifted with distance would mean the relation was an approximation.

```
ratio_err <- max(vapply(seq(pi, 0.05, length.out = 40),
 ↪  function(rho) {
  max(vapply(c(0.1, 1, 7.3), function(d) {
    P <- c( d * sin(rho / 2), 0, d * cos(rho / 2))
    Q <- c(-d * sin(rho / 2), 0, d * cos(rho / 2))
    abs(sqrt(sum((P - Q)^2)) / (2 * d) - sin(rho / 2))
  }, numeric(1)))
}, numeric(1)))
```

Over forty dihedral angles and three separations the worst departure from $\sin(\rho/2)$ is 0 — one unit in the last place, which is what "exact" means in double precision.

A flat sheet has $\rho = \pi$ and contracts nothing. Every fold contracts, and since $\sin(\rho/2) < 1$ for every $\rho < \pi$, **ambient distance is strictly less than chart distance for every $\theta > 0$.**

2.4.5 Five: the floor under "strictly"

That last sentence is false as written, and the chapter says so rather than leaving a reader to trip over it in Part III.

Because $1-\sin(\rho/2) \approx (\pi-\rho)^2/8$ near the flat state, the contraction is quadratic in the fold angle while machine epsilon is not. There is therefore a fold angle below which the contraction exists mathematically and is not representable.

```
gap        <- function(g) 1 - sin((pi - g) / 2)          # g = pi
 ↪  - rho
floor_pred <- sqrt(8 * .Machine$double.eps / 2)
theta_min  <- min(THETA_GRID[THETA_GRID > 0])
```

The predicted floor is 0 radians, and it behaves: `gap()` returns 0 at a fold angle of 10^{-3} and exactly 0 at 10^{-8}. The smallest non-zero θ this book's grid uses is 0.05, which is above the floor by a wide margin — and that margin is the only reason Part III may treat the contraction as strict.

2.5 Results

Three consequences follow, and they are the reason this chapter comes before the methods rather than after them.

The answer key is not what the methods reproduce. A method that consumes ambient distance is being asked to reproduce a configuration that folding has already contracted. Scoring it against the intrinsic chart is a legitimate question — it is the question of whether the method recovers the *surface* — but it is not the question the method's loss function asks.

The exact answer scores badly on the wrong metric, provably. This deserves stating as a proposition rather than demonstrating with a simulation.

> **Proposition.** Let stress be measured against ambient distance, and let the candidate embeddings be the exact chart $\mathcal{U}$ and the two-component PCA of the folded cloud. Then for every $\theta > 0$, $\text{stress}(\mathcal{U}) > \text{stress}(\text{PCA})$.

The proof is one line given step four. Ambient-referenced stress measures reproduction of d_A; the contraction gives $d_A < d_U$ strictly; and two-component PCA is by construction the least-squares-optimal *linear* approximant to that same ambient configuration (Pearson, 1901; Eckart and Young, 1936). The exact unfolding is not competing to reproduce ambient distance and loses to something that is.

An evaluator that ranks the true answer below a plainly wrong one is not malfunctioning. It is answering the question it was asked. Chapter 9 measures how far that goes on real evaluators, and finds it goes further than this proposition requires.

Difficulty has a knob. Because θ is continuous and the contraction is monotone in it, every result in this book is a curve rather than a number. That is standing rule 2, and it is not a stylistic preference. Chapter 8 reports a case where two method families swap places partway along the θ axis, so that the average over θ names a different winner from the curve at seven of its nine points. A benchmark with one fixed geometry cannot produce that observation, because it has nowhere to put it.

This is the specific sense in which a crease pattern is a better instrument than a Swiss roll, and it is worth separating from the claim it is often confused with. It is not that folded sheets resemble real data more closely. It is that the truth is

known **and** the difficulty is continuously adjustable, so a method's performance is a function rather than a point, and the shape of that function is a property of the method. The critique that two-dimensional embeddings distort more than their users admit (Chari and Pachter, 2023; Kobak and Berens, 2019) is an argument this book answers with measurement, and measurement of that kind needs an axis to measure along.

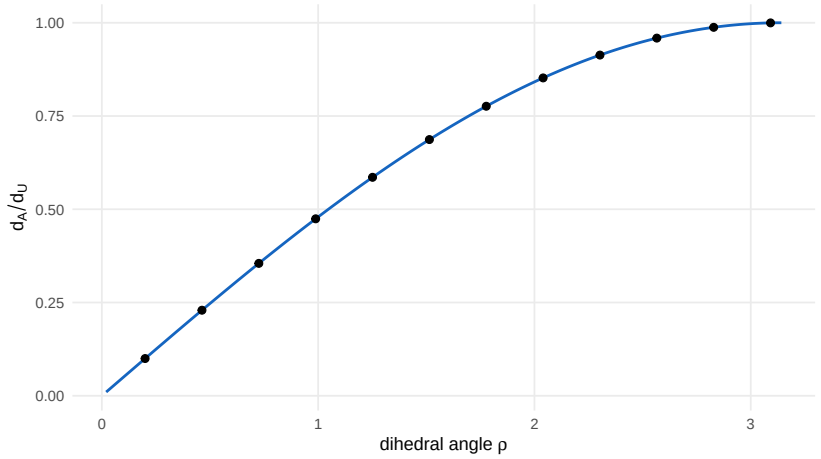


Figure 2.3: The ambient-to-geodesic distance ratio across a single crease, against the dihedral angle. The curve is $\sin(\rho/2)$; the points are measured from folded coordinates. A flat sheet sits at the right-hand end and contracts nothing.

2.6 Diagnostics

The most useful material in this chapter is about two things this project got wrong. Both were caught, both are cheap to state, and both generalise past origami.

2.6.1 An isometry test cannot tell a fold from a pleat

A second pattern family, the Yoshimura, shipped for three days with a folding that was not one. It folded its horizontal creases and left every diagonal at $\rho = \pi$ — an accordion pleat wearing a diamond tessellation's crease pattern.

It passed the facet-isometry test perfectly, because **a pleat is a rigid folding**. It passed Kawasaki, because the flat pattern was never the problem. Neither test could have caught it: both are properties a correct implementation has, and a pleat has them too.

What caught it was deriving the mountain/valley assignment from the *folded* dihedral angles and testing Maekawa on the result. A pleat leaves whole crease families flat, and a vertex with unfolded creases cannot balance:

```
a  <- crease_assignment(p)
iv <- interior_vertices(p)
```

```
maekawa <- vapply(iv, function(v) {
  sel <- (p$creases$i == v | p$creases$j == v) & a != "B"
  abs(sum(a[sel] == "M") - sum(a[sel] == "V"))
}, numeric(1))
```

On this Miura all 25 interior vertices satisfy $|M - V| = 2$. On the pleat, twelve of sixteen failed.

2.6.2 A test that a correct implementation passes is not a test a wrong one fails

That is the moral, and the same chapter has to record that this book learned it twice.

The derived assignment above was itself wrong for four days, in a way Maekawa could not see. Maekawa counts labels at a vertex; it is therefore invariant under swapping every mountain for every valley, so it passes on a labelling, on that labelling's exact inverse, and on a third that agrees with neither on half the creases. All three were live in this repository at once, and the one the figures drew was correct on half the interior creases.

```
inverted <- ifelse(a == "M", "V", ifelse(a == "V", "M", "B"))
maekawa_inv <- vapply(iv, function(v) {
  sel <- (p$creases$i == v | p$creases$j == v) & inverted != "B"
  abs(sum(inverted[sel] == "M") - sum(inverted[sel] == "V"))
}, numeric(1))
```

The inverted labelling — every mountain a valley — satisfies Maekawa at 25 of 25 interior vertices, which is all of them. The check that finally settled it computes the assignment a second time by an independent route, from facet normals oriented against the ambient frame rather than against the mesh's own winding, and compares.

2.6.3 A first-order flex at the flat state proves nothing

The third instance, recorded here because it is the same error in a different costume. At $\theta = 0$ every z column of a pattern's rigidity Jacobian vanishes, so the count of non-trivial infinitesimal flexes is $V - 3$ for **any** planar pattern whatsoever — including patterns with no rigid folding at all. A first-order flex at the flat state is a property of being flat, not of being foldable. Second-order analysis and a finite continuation are what decide, and Chapter 8 records what they decided.

2.7 Where it fails

2.7.1 The chart is the geodesic only inside a convex region

Step three claimed that geodesic distance equals chart distance. That is true when the straight segment between two chart points stays on the paper, and a Miura's unfolded outline is **not convex** — its edge is a zigzag with teeth. For

a pair straddling a tooth the straight chart segment leaves the sheet, the true geodesic must detour, and the chart distance is a *lower bound* rather than an equality.

```
m_in  <- sample_manifold(p, theta = 0.6, n = 400L, seed = 1001L)
m_out <- sample_manifold(p, theta = 0.6, n = 400L, seed = 1001L,
 ↪  boundary = TRUE)
exit_in  <- chart_exit_fraction(m_in,  p)
exit_out <- chart_exit_fraction(m_out, p)
```

`chart_exit_fraction()` measures how often it happens by sampling pairs and walking the segment between them. Drawing strictly inside a convex sub-rectangle gives an exit fraction of 0; including the boundary strip gives 0.013. Every artefact in this book that the answer key is read against is drawn the first way, and the producer that generates the irreducible-loss suite asserts the zero rather than trusting this paragraph.

2.7.2 Self-intersection

Nothing in the algebra of step one prevents a folded placement from passing through itself, and a self-intersecting surface would invalidate every ambient distance measured on it.

```
gap_at <- vapply(c(0.3, 0.6, 0.9), function(t) facet_gap(p,
 ↪  t)$gap, numeric(1))
```

`facet_gap()` computes the exact minimum distance between non-adjacent facets — polygon to polygon, analytically, not by sampling points on them. At θ of 0.3, 0.6 and 0.9 it is 0.851, 0.786 and 0.526: falling as the sheet closes, as it must, and positive throughout.

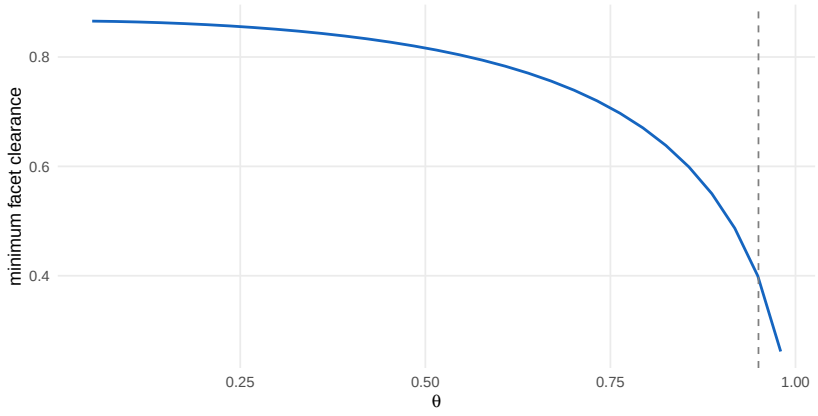


Figure 2.4: Minimum clearance between non-adjacent facets, against the folding parameter. The surface approaches itself as it folds and never touches; the dashed line marks the largest fold angle the benchmark grid uses.

Figure 2.4 is the whole range. The clearance falls smoothly and stays positive, and the largest θ the benchmark grid uses is marked: at 0.95 the sheet still

has 0.396 of clearance. That is the check behind an assumption every ambient distance in this book depends on, and it is a check rather than an argument because the algebra of step one does not forbid self-intersection — it simply does not produce one for this family in this range.

2.7.3 Sampling, and what a finite sample of a zero-reach surface is

The reach remark in `geom-background` has a practical edge that belongs here. Every guarantee about recovering a manifold from a finite sample is stated in terms of the density of the sample relative to the reach (Niyogi et al., 2008), and a crease has reach zero, so no such guarantee is available at any sample size. What replaces it in this book is not a weaker theorem but a measurement: the sampled surface is checked for how close distant parts of it come, in units of the local spacing, and that ratio is reported alongside every result that depends on it. Isomap's failure mode on tightly folded surfaces is precisely a short circuit between sheets that the sampling cannot distinguish (Tenenbaum et al., 2000; Balasubramanian and Schwartz, 2002), and the quantity that predicts it is a property of the sample rather than of the surface.

2.7.4 One family

This book ships one pattern family that folds. Two others were specified and both were withdrawn on evidence, which Chapter 8 reports as results rather than as gaps. A benchmark built on a single family cannot support claims about pattern variety, and this book does not make any.

2.7.5 What the chapter does not settle

Whether a folded sheet is a *good* model of the manifolds that arise in real data is not a geometric question and is not answered here. Chapter 11 measures what this benchmark discriminates that the classical ones do not, and Chapter 12 takes the result to a real dataset. The honest position in between is that a crease pattern is a manifold whose truth is known exactly, which is a property no real dataset has and no previously used benchmark provides.

2.8 Reproduce this

Every claim above is an assertion in the test suite, and the suite runs on every push:

- `tests/testthat/test-folding.R` — facet isometry across the θ sweep and the α range, no self-intersection, the dihedral contract.
- `tests/testthat/test-patterns.R` — congruence, Kawasaki, Maekawa, vertex degree.
- `tests/testthat/test-crease-assignment.R` — the mountain/valley derivation against an independent criterion, and the demonstration that Maekawa cannot discriminate.
- `tests/testthat/test-contraction.R` — $\sin(\rho/2)$ and the floating-point floor.

Run them with `Rscript tests/testthat.R` from the repository root.

2.9 Further reading

Hull's *Origametry* (Hull, 2020) is the single best entry point to the mathematics of flat-foldability and is the reference for both classical theorems above. Demaine and O'Rourke (Demaine and O'Rourke, 2007) is the standard treatment of the computational side, and Lang (Lang, 2011) of design. Miura's original report (Miura, 1985) introduces the pattern this book uses; Schenk and Guest (Schenk and Guest, 2013) give the mechanics of folded shell structures and are the natural next step for a reader interested in the surfaces rather than the sheets.

On the manifold-learning side, Federer's reach (Federer, 1959) and its role in sampling guarantees (Niyogi et al., 2008) are what the zero-reach remark above refers to, and Lee and Verleysen (Lee and Verleysen, 2007) is the textbook treatment of the methods Part II takes up.

Chapter 3

Folding as a generative model

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

How do you turn one pattern into a family of test problems?

3.1 What this chapter answers

Chapter 2 established one object: a crease pattern whose flat chart is its exact two-dimensional structure. One object is not a benchmark. A benchmark is a *family* of problems ordered by difficulty, and this chapter is where the single sheet becomes that family.

The generative model has four scalars and three switches. Three of the scalars — the cell sides a and b and the acute angle α — fix the chart. The fourth, θ, moves the sheet through space and does not touch the chart at all. The switches are the sample size, the noise model and the seed. Every row of every artefact this book ships is one setting of those seven things, and every curve in Parts II and III is a sweep over the fourth scalar with the other six held.

This chapter is the reference for that model. It derives the folded placement in closed form and verifies the derivation here rather than citing one; it says exactly how points are drawn and what a careless draw would do to the density; it names the three noise models and no others; and it measures what θ does to the k-nearest-neighbour graph a method actually consumes. Chapter 8 builds constructions on top of this model, and every chapter from 4 onward reads an artefact produced by it.

3.2 Setup

```
p  <- miura_ori(nx = 6L, ny = 6L, a = 1, b = 1, alpha = pi / 3)
th <- max(THETA_GRID)
s  <- sample_manifold(p, theta = th, n = N_DEFAULT, seed = 1001L)
```

`p` is the pattern the benchmark grid uses — `PATTERN_GRID` names its size and `R/constants.R` explains why there is one entry — `th` the largest fold the grid reaches, and `s` a sample of 800 points drawn from the sheet at that fold. Nothing in this chapter reads a committed artefact; every number below is computed while the page is built. The notation appendix fixes the symbols.

3.3 Background

Dimension reduction is usually benchmarked against a *collection*: a fixed list of datasets, some real and some synthetic, each scored once and averaged over (Espadoto et al., 2021). The synthetic members of such a collection — the Swiss roll, the S-curve, the severed sphere — are already generative in the weak sense that they come with a parameterisation, so a ground-truth chart exists (Tenenbaum et al., 2000; Lee and Verleysen, 2007). What they mostly lack is a knob: a continuous parameter that makes the *same* problem harder while leaving the answer exactly where it was.

A generative model that can carry a benchmark has to supply four things.

An exact chart. Not a low-dimensional summary that scores well, but the coordinates the data were generated from. A rigid folding gives this for free, which is the whole reason the book uses one.

A difficulty axis that moves the ambient geometry and nothing else. Folding moves every point in $\mathbb{R}^3$ and moves no point of the chart. That is the property θ has and a dataset collection does not: two cells of the grid can differ in difficulty while their answer keys are literally the same matrix.

A noise model that is not confounded with the axis. This one is easy to get wrong. A noise scale keyed to the folded object would shrink as the sheet closes, so a fixed setting would become a rising noise level along the θ sweep and the two axes could not be separated afterwards. The scale here is a fraction of the *chart* diameter, which does not move.

A seed, and a sampler that respects it. Every stochastic result in this book is reported across 20 seeds (standing rule 1), which requires a sampler whose draw is a function of its seed and whose call does not disturb the caller's random stream.

One thing the model cannot supply is a sampling guarantee. The theory that says how many points are enough to recover a submanifold is stated in terms of its reach (Niyogi et al., 2008), and Chapter 2 established that a crease has reach exactly zero. Nothing here is covered by that theory, so sampling adequacy is *measured* on every sample instead — which is what `gen-results` does.

3.4 The Miura-ori in four scalars

Four scalars go in and four come out, and the two sets are not the same four.

Going in are a, b, α and θ. The first three fix the flat pattern: vertex (i, j) of the tessellation sits at

$$x = i\,a + (j \bmod 2)\,b\cos\alpha, \qquad y = j\,b\sin\alpha,$$

which is Miura's parallelogram tessellation (Miura, 1985) and is all of `miura_ori()`. The counts n_x and n_y set how many cells there are, not what shape they have. θ appears nowhere in that expression, and that absence is the model's load-bearing property.

Coming out are the four metric constants of the folded placement. Chapter 2 derived them; the derivation is repeated here in one line because this chapter is where a reader is sent to find it. Posit a folded placement of the same combinatorial form,

$$x = iL_x + (j \bmod 2)\,d_x, \qquad y = jL_y, \qquad z = (i \bmod 2)\,H,$$

and impose isometry on one facet. Its two side vectors are $e_1 = (L_x, 0, \pm H)$ of flat length a and $e_2 = (\pm d_x, L_y, 0)$ of flat length b, so preserving both lengths and the angle between them gives

$$L_x^2 + H^2 = a^2, \qquad d_x^2 + L_y^2 = b^2, \qquad L_x d_x = ab\cos\alpha.$$

Three equations, four unknowns, one free parameter. Writing $L_x = a\cos\varphi$ and $H = a\sin\varphi$ solves the first identically and leaves $d_x = b\cos\alpha/\cos\varphi$ and $L_y = b\sqrt{1 - \cos^2\alpha/\cos^2\varphi}$. That L_y is real only while $\varphi \le \alpha$ and vanishes at $\varphi = \alpha$, which is the flat-folded state — so φ runs on $[0, \alpha]$ and $\theta = \varphi/\alpha$ is the normalised parameter `fold()` takes.

3.4.1 The derivation, checked

Those three conditions are what the placement was built to satisfy, so testing them on the formula would be circular. What is not circular is reading the four constants back *out* of the coordinates `fold()` returns and testing the conditions on those.

```
scalars_of <- function(pattern, theta) {
  r   <- pattern$params
  vid <- function(i, j) j * (r$nx + 1L) + i + 1L        # (i, j)
 ↪  -> vertex row
  V   <- unname(fold(pattern, theta)$vertices3)
  c(Lx = V[vid(1L, 0L), 1L] - V[vid(0L, 0L), 1L],
    H  = V[vid(1L, 0L), 3L] - V[vid(0L, 0L), 3L],
    dx = V[vid(0L, 1L), 1L] - V[vid(0L, 0L), 1L],
    Ly = V[vid(0L, 1L), 2L] - V[vid(0L, 0L), 2L])
}
conditions <- function(pattern, theta) {
  r <- pattern$params
  z <- scalars_of(pattern, theta)
```

```
  c(z[["Lx"]]^2 + z[["H"]]^2  - r$a^2,
    z[["dx"]]^2 + z[["Ly"]]^2 - r$b^2,
    z[["Lx"]] * z[["dx"]]     - r$a * r$b * cos(r$alpha))
}
alphas <- seq(MIURA_ALPHA_RANGE[1], MIURA_ALPHA_RANGE[2],
 ↪  length.out = 14L)
worst  <- max(vapply(alphas, function(A) {
  pa <- miura_ori(4L, 4L, a = 1, b = 1.3, alpha = A)
  max(abs(vapply(THETA_GRID, function(t) conditions(pa, t),
   ↪  numeric(3))))
}, numeric(1)))
ends <- rbind(flat = scalars_of(p, 0), folded = scalars_of(p, 1))
```

Over fourteen opening angles spanning `MIURA_ALPHA_RANGE` and the whole θ grid, the worst violation of any of the three conditions is 6.7e-16 — floating-point noise on a closed form, not a solver tolerance. The endpoints behave too: at $\theta = 0$ the four constants are 1, 0, 0.5, 0.866, which is the flat pattern recovered exactly rather than in a limit, and at $\theta = 1$ they are 0.5, 0.866, 1, 0 — L_y has gone to zero and the sheet has no extent in y at all.

The four constants also say what folding does to the sheet as an object. L_x and L_y both fall while H and d_x rise, so the sheet's overall extent shrinks in *both* in-plane directions at once while it gains depth in the third — the auxetic behaviour the Miura is usually presented for, arriving here as a consequence of the three conditions rather than as a design goal. Nothing else in the model depends on n_x and n_y beyond the count of cells: a larger tessellation is a larger sheet made of the same cell, which is why crease count is a difficulty knob in its own right and is swept separately rather than folded into θ.

> **This derivation is not quoted from anywhere.** Schenk and Guest [@schenk2013geometry] give unit-cell quantities for the Miura under a different angle convention, and their parameterisation and this one are *related work rather than the same statement*. No map between the two is asserted here, because none has been verified in this repository, and a reconciliation asserted without one would be exactly the citation-by-osmosis this book is trying to avoid. The algebra above stands on the check in `gen-scalars` and on `tests/testthat/test-folding.R`, and on nothing else.

Figure 3.1 is the family. Reading down a column changes the chart; reading across a row does not change it by one coordinate. A benchmark cell is a position in that array.

Figure 3.2 is the one-parameter family written out. Two features of it matter downstream. The constants are smooth and monotone in θ, so difficulty is ordered; and they are markedly *not* linear in it, so the last fifth of the sweep does most of the geometric work. Both facts turn up again in `gen-results`.

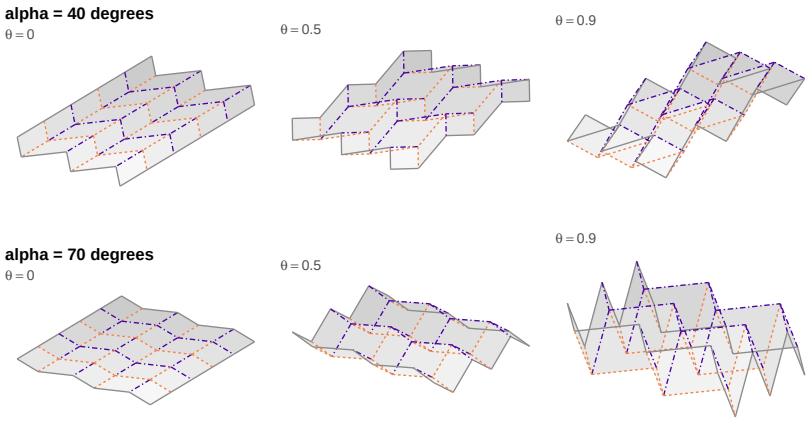


Figure 3.1: The generative family. Rows are two opening angles α; columns are three values of the folding parameter. The chart differs between rows and is identical along each row: θ moves the sheet and never the answer key.

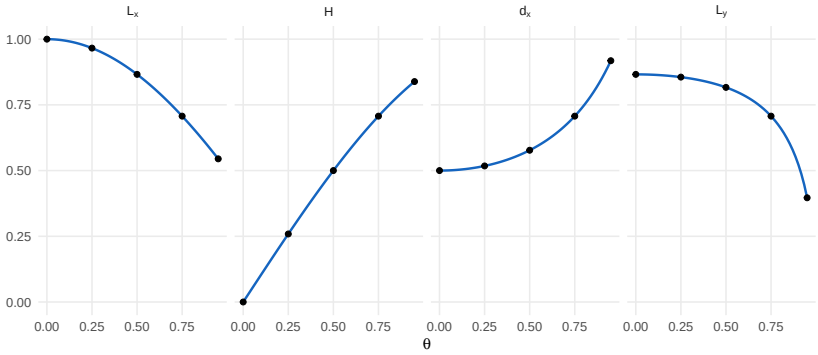


Figure 3.2: The four metric constants of the folded placement against the folding parameter, for the pattern of ‘gen-setup‘. Lines are the closed form; points are the same constants read back out of the coordinates ‘fold()‘ returns.

3.5 Sampling and noise

The chart is the sample space. `sample_manifold()` draws points in the flat pattern, records them as `truth`, and carries each one into $\mathbb{R}^3$ through the rigid motion of the facet it landed on to give `X`. The two matrices are the same points seen twice, and nothing is interpolated between folded vertices — the per-facet map is built from a frame and then *checked* against the corners, so a folding that stopped being rigid would fail rather than quietly return positions no point of the surface occupies.

Drawing in the chart means drawing uniformly over area, and that is where a naive implementation goes wrong. `.fan_triangles()` splits every facet into a fan from its first vertex and returns the area of each triangle; the fan is checked rather than assumed, since it tiles a convex polygon and not a re-entrant one. `.draw_chart_points()` then picks a triangle with probability proportional to its area and maps a pair of uniforms into it — with a square root on the first.

```
tri <- .fan_triangles(p$vertices, p$facets)          # internal:
 ↪  see R/sampling.R
draw_chart <- function(m, root) {                    # root =
 ↪  FALSE is the naive draw
  k  <- sample.int(nrow(tri$idx), m, replace = TRUE, prob =
 ↪  tri$area)
  va <- p$vertices[tri$idx[k, 1L], ]
  vb <- p$vertices[tri$idx[k, 2L], ]
  vc <- p$vertices[tri$idx[k, 3L], ]
  r1 <- stats::runif(m); if (root) r1 <- sqrt(r1)
  r2 <- stats::runif(m)
  list(u = (1 - r1) * va + (r1 * (1 - r2)) * vb + (r1 * r2) * vc,
   ↪  r1 = r1)
}
cell_cv <- function(u, bins = 36L) {
  n <- as.vector(table(cut(u[, 1L], bins), cut(u[, 2L], bins)))
  stats::sd(n[n > 0L]) / mean(n[n > 0L])
}
set.seed(7L)
good <- draw_chart(20000L, root = TRUE)
bad  <- draw_chart(20000L, root = FALSE)
apex <- c(area = mean(good$r1 <= 0.5), naive = mean(bad$r1 <=
 ↪  0.5))
cv   <- c(area = cell_cv(good$u),      naive = cell_cv(bad$u))
area_spread <- diff(range(tri$area)) / mean(tri$area)
```

Without the square root, the radial coordinate is uniform where the area element is not: the fraction of a triangle within half-scale of its apex is one quarter by area, and the naive draw puts 0.502 of its points there against 0.251 for the corrected one. The apex of every fan is a *vertex of the crease pattern*, so the naive draw piles density onto exactly the features the benchmark is about. Counted on a square grid over the chart, the cell-count coefficient of variation is 0.855 against 0.283 — a density gradient roughly 3 times as strong, and every neighbourhood metric in Part II reads a density gradient as structure.

The area weighting itself is a different correction and on this family it does nothing: every Miura facet is congruent, so the spread of triangle areas is 1.2e-15, and area-proportional and facet-uniform draws coincide. That is a fact about the pattern, not about the sampler, and it is listed in `gen-limits` as something this book does not test.

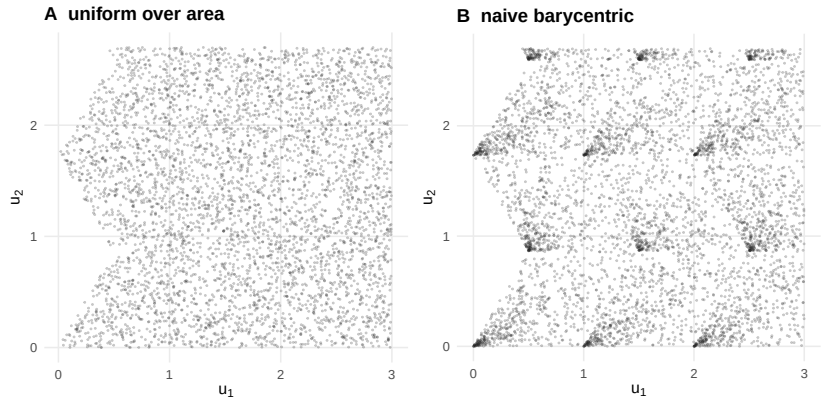


Figure 3.3: The same triangles, drawn from twice. Panel A is the sampler's draw, uniform over area. Panel B is the naive barycentric draw without the square root, which concentrates points at the fan apex of every triangle — that is, at the crease-pattern vertices.

`boundary = FALSE`, which is the default and what every committed artefact uses, keeps only points at least half a median crease length from the sheet edge, by rejection from the same uniform draw. Rejection rather than re-triangulation is deliberate: survivors of a uniform draw are uniform over whatever region survives, however awkward its shape.

Then noise, and there are exactly three models. `"none"` leaves `X` alone. `"ambient"` adds isotropic Gaussian displacement in $\mathbb{R}^3$, applied *after* folding so that it moves points off the surface. `"outlier"` throws a fixed fraction of the points a fixed distance off it. There is no on-manifold noise model: displacement inside the chart is a reparameterisation of the same surface rather than measurement error, and `sample_manifold()` rejects any type it does not know.

```
NOISE <- list(none    = list(type = "none",    sd = 0),
              ambient = list(type = "ambient", sd = 0.02),
              outlier = list(type = "outlier", sd = 0.05))
theta_lo <- THETA_GRID[3L]
theta_hi <- THETA_GRID[19L]
noise_at <- function(theta) {
  base <- sample_manifold(p, theta, N_DEFAULT, seed = 1001L)
  do.call(rbind, lapply(names(NOISE), function(nm) {
    z <- sample_manifold(p, theta, N_DEFAULT, noise =
 ↪  NOISE[[nm]], seed = 1001L)
    d <- sqrt(rowSums((z$X - base$X)^2))
```

```
    data.frame(theta = theta, noise = nm, moved = sum(d > 0),
               rms = sqrt(mean(d^2)), worst = max(d),
               truth_fixed = identical(z$truth, base$truth),
               extent = max(stats::dist(base$X)))
  }))
}
nz <- rbind(noise_at(theta_lo), noise_at(theta_hi))
set.seed(11L); before <- stats::runif(1)
set.seed(11L); invisible(sample_manifold(p, th, 200L, seed = 5L))
after <- stats::runif(1)
stream_kept <- identical(before, after)
```

The `ambient` row moves all 800 points; the `outlier` row moves 40 of them, by exactly 3.97 each, which is half the chart diameter. `truth` is byte-identical to the clean draw in every row — the answer key does not move when the data get worse.

3.6 Results: what theta does

θ changes no chart coordinate. What it changes is the graph a method builds before it does anything else.

```
k <- K_DEFAULT
graph_at <- function(theta, n = N_DEFAULT, seed = 1001L, pattern
 ↪  = p) {
  z  <- sample_manifold(pattern, theta, n, seed = seed)
  dA <- as.matrix(stats::dist(z$X));     diag(dA) <- Inf
  dU <- as.matrix(stats::dist(z$truth)); diag(dU) <- Inf
  nn <- t(apply(dA, 1L, function(r) order(r)[seq_len(k)]))
  rk <- t(apply(dU, 1L, rank, ties.method = "first"))
  i  <- rep(seq_len(n), k)
  j  <- as.vector(nn)
  list(sample = z, i = i, j = j, chart_rank = rk[cbind(i, j)])
}
sweep_theta <- do.call(rbind, lapply(THETA_GRID, function(t) {
  g <- graph_at(t)
  data.frame(theta = t,
             false_nb   = mean(g$chart_rank > k),
             worst_rank = max(g$chart_rank),
             sci        = short_circuit_index(g$sample),
             gs         = branch_gap(g$sample)$ratio,
             clearance  = facet_gap(p, t)$gap)
}))
top  <- graph_at(th)
U    <- top$sample$truth
fls  <- top$chart_rank > k
fcol <- (top$sample$facet - 1L) %% p$params$nx     # facet column
 ↪  and row
frow <- (top$sample$facet - 1L) %/% p$params$nx
wider <- abs(U[top$i, 1L] - U[top$j, 1L]) > abs(U[top$i, 2L] -
 ↪  U[top$j, 2L])
```

```
mistakes <- c(
  n_false    = sum(fls),
  n_same     = sum((top$sample$facet[top$i] ==
   ↪  top$sample$facet[top$j])[fls]),
  n_one_cell = sum((abs(fcol[top$i] - fcol[top$j]) <= 1L &
                    abs(frow[top$i] - frow[top$j]) <= 1L)[fls]),
  across_f   = round(100 * mean(wider[fls])),
  across_h   = round(100 * mean(wider[!fls])))
```

Four quantities, all computed on the same samples, all monotone, and none of them linear.

The graph acquires false neighbours. An ambient k-nearest-neighbour edge is *false* when its far endpoint is not among the point's k nearest neighbours in the chart. At $\theta = 0$ the ambient and chart graphs are identical and the false fraction is 0. At $\theta = 0.95$ it is 0.302: about three edges in ten of the graph a method consumes join points that are not neighbours on the paper. The worst single edge joins a point to its 168th chart neighbour.

Edges contract. `short_circuit_index()` is the median ambient nearest-neighbour distance over the median true intrinsic distance between the same pairs, so it is one on a faithful graph. It falls from 1 to 0.787.

Distant sheets approach, but not far enough. Balasubramanian and Schwartz's ratio g/s — the closest approach between surface-distant sampled points, in units of the median nearest-neighbour spacing — is the quantity that predicts Isomap's topological instability (Balasubramanian and Schwartz, 2002). It falls from 20.4 to 5.84, which is a large fall and still leaves the sheets nearly six sample spacings apart.

The false-neighbour count above is measured at `K_DEFAULT` against the *chart*, and it is worth saying so because the book uses two other pairings and they are easy to conflate. Q_{NX} is scored at `K_QNX` against the chart, because that is the answer key and the co-ranking measure is noisy at small K; trustworthiness, continuity and kNN preservation are scored at `K_DEFAULT` against *ambient* distance, because they ask whether an embedding preserved the neighbourhoods a method could actually see. Three combinations of neighbourhood size and reference geometry, each answering a different question, all named in `R/constants.R` so that they cannot drift.

Figure 3.4 is why standing rule 2 exists. A single number reported at one θ can be read off any point of those curves, and the curves are shallow over the first half of the axis and steep over the last quarter. Averaging over θ would compress the only axis the benchmark has; Chapter 10 reports a case where two method families change places partway along it, which no summary at a fixed geometry can show.

Where the false edges sit is as informative as how many there are.

Figure 3.5 draws both halves of that graph on the chart, and the two halves occupy different parts of it. Honest edges fill the facets and thin out along the creases; false edges do the reverse and lie in narrow bands on the crease lines. Of the 2416 false edges, 0 have both endpoints in the same facet — inside a

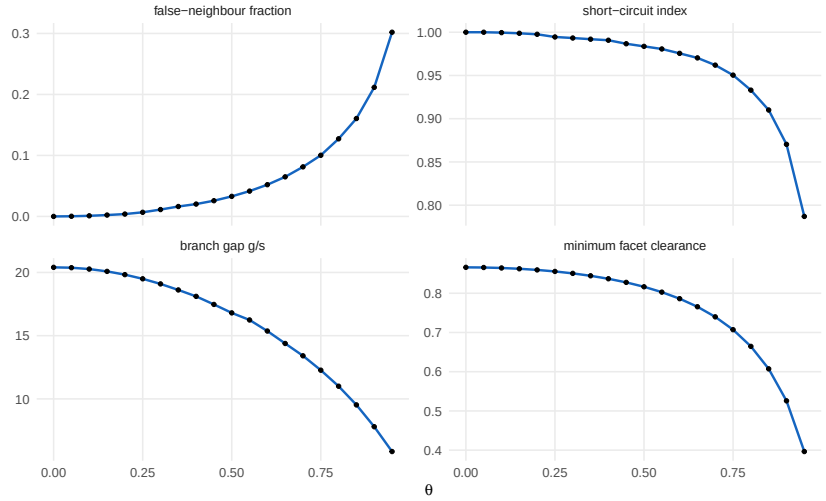


Figure 3.4: Four difficulty measures over the benchmark grid's θ sweep, computed on one seed at $n = 800$. All four are monotone and none is linear: the last quarter of the axis does most of the work.

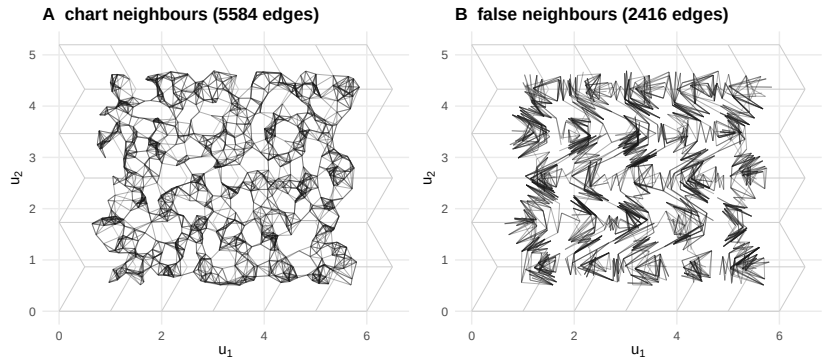


Figure 3.5: The ambient k-nearest-neighbour graph at the grid's largest fold, drawn on the chart it should have respected. Panel A holds the edges that are also chart neighbours; panel B holds the false ones. The false edges are not scattered over the sheet — they trace the crease lines, and every one of them straddles a fold.

facet, ambient distance *is* chart distance — so every false neighbour straddles a fold. The mistakes are also strictly local: all 2416 of them join facets no more than one cell apart in either direction. There is a mild directional tilt on top of that, 62% of false edges being longer across the corrugation than along it against 49% of honest ones, but the localisation is the dominant fact and the direction is not.

That localisation is why difficulty here is not simply "more noise". Additive noise degrades every neighbourhood alike; folding leaves the interior of every facet exact and corrupts a codimension-one set of neighbourhoods completely. That is a structured failure, and whether a method is robust to it is a different question from whether it tolerates noise — which is why the benchmark grid crosses the two axes instead of collapsing them.

3.7 Diagnostics

θ is not a dihedral angle. It is the fraction of the way to flat-folded, on $[0, 1]$, and `fold()` refuses anything outside that. The dihedral angles are derived and returned in `rho`; at $\theta = 0.95$ the creases of `p` span 0.504 to 3.14 radians, the upper end being the boundary edges, which do not fold at all. A flat sheet has $\rho = \pi$, not zero. Conflating the two is an error this project made once and corrected in three places.

`noise$sd` is a fraction of the chart diameter. Not of the folded object. The `ambient` row displaces points by an RMS of 0.272 at $\theta = 0.1$ and 0.272 at $\theta = 0.9$ — the same number — while the folded cloud's own extent falls from 6.19 to 3.93. Had the scale been keyed to the folded object, the noise axis and the fold axis would move together and no result could be attributed to either.

In the `outlier` row, `sd` is a contamination fraction. The grid passes `sd = 0.05` there and means five per cent of the points, not a standard deviation; the displacement magnitude is fixed, not random. This is a genuine trap in the interface and it is documented in `R/sampling.R` rather than fixed, because the artefacts were computed under it.

A seed reproduces a draw without costing the caller their stream. `sample_manifold()` saves `.Random.seed`, replaces it, and restores it on exit, so drawing a sample inside a loop does not silently re-align the loop's own randomness. Two draws from one stream, with a sample taken between them, agree: TRUE. Called without a seed it touches `set.seed()` not at all and records `NA`, so a returned object never claims a seed that would not reproduce it.

3.8 Where it fails

The classical short circuit lives past the end of the grid. `gen-results` found no edge joining facets more than one cell apart anywhere on `THETA_GRID`. That is not because folding cannot produce one.

```
cells_apart <- function(g, pattern) {
  cl <- (g$sample$facet - 1L) %%  pattern$params$nx
```

```
  rw <- (g$sample$facet - 1L) %/% pattern$params$nx
  pmax(abs(cl[g$i] - cl[g$j]), abs(rw[g$i] - rw[g$j]))
}
beyond <- do.call(rbind, lapply(c(max(THETA_GRID), 0.99, 0.999),
 ↪  function(t) {
  g <- graph_at(t)
  data.frame(theta = t,
             gs    = branch_gap(g$sample)$ratio,
             sci   = short_circuit_index(g$sample),
             jumps = mean(cells_apart(g, p) >= 2L))
}))
```

Past the grid the ratio g/s collapses — 5.84 at $\theta = 0.95$, 1.37 at $\theta = 0.999$ — and edges joining facets two or more cells apart appear: 0 of the graph at the grid's edge, 0.174 at the last value. The grid stops at 0.95 because past it the sheet is collapsing onto itself and the cell measures a degenerate object rather than a harder instance of the same one. The consequence is worth stating plainly: **this benchmark's difficulty axis does not reach the regime that classically breaks Isomap.** What it measures instead is the graded corruption of the neighbourhood graph that precedes it.

Area weighting is correct and untested by this family. Every Miura facet is congruent, so area-proportional and facet-uniform sampling agree to 1.2e-15 and no result in this book distinguishes them. The weighting is there because a family with unequal facets would need it, and this book ships no such family — see Chapter 2 on the two that were withdrawn.

Trimming the boundary changes the region, not just the count. With `boundary = FALSE` the facets nearest the sheet edge keep a small share of their points: across the 36 facets of `p` the occupancy of `s` runs from 3 to 40. The survivors are uniform over the region that survives, which is the guarantee that was wanted, but that region is not the whole sheet.

Three scalars are a cell, not a family of patterns. Varying a, b and α changes the cell shape and never the combinatorics: the same 49 vertices and 36 facets, connected the same way. This model produces a one-parameter deformation of one tessellation, so nothing here supports a claim about pattern variety.

The chart is the geodesic only inside a convex region. The caveat is Chapter 2's and it applies to every sample drawn here; `boundary = FALSE` is what keeps it negligible, and `chart_exit_fraction()` is what measures it rather than assuming it.

3.9 Reproduce this

Every claim above is an assertion in the suite, which runs on every push:

- `tests/testthat/test-sampling.R` — area-uniformity against an independent shoelace computation, points lying in the facet they claim and on the folded surface, seed reproducibility, and the RNG-stream restore.

- `tests/testthat/test-folding.R` — facet isometry over the θ sweep and the α range, the derived dihedral contract, and the absence of self-intersection.
- `tests/testthat/test-patterns.R` — congruence, Kawasaki and Maekawa.

Run them with `Rscript tests/testthat.R` from the repository root. Every figure and every number in this chapter is recomputed at render time from `R/` alone, so a change to the sampler or the folding map shows up here as a changed page rather than as a stale one.

The model's settings are not scattered: `THETA_GRID`, `N_DEFAULT`, `PATTERN_GRID`, `MIURA_ALPHA_RANGE`, `K_DEFAULT` and `K_QNX` are all in `R/constants.R`, and `scripts/run-benchmark-grid.R` is the producer that turns them into `data/processed/benchmark-grid.rds`. Chapter 4 opens by reading that artefact with its digest, and Appendix B is its codebook.

3.10 Further reading

Miura's original report (Miura, 1985) introduces the tessellation; Schenk and Guest (Schenk and Guest, 2013) treat the folded sheet as a mechanical metamaterial and are the natural next step for a reader who wants the unit-cell geometry from the engineering side. Hull's *Origametry* (Hull, 2020) is the reference for flat-foldability and for why a pattern folds at all.

On the benchmarking side, Espadoto and colleagues (Espadoto et al., 2021) survey how dimension reduction is usually evaluated and make the case for scale that this book answers with a difficulty axis instead; Machado and colleagues (Machado et al., 2025) argue that projection quality metrics are necessary and not sufficient, which is the question Chapter 9 takes up with an exact answer key in hand. Lee and Verleysen (Lee and Verleysen, 2007) remains the textbook treatment of the methods Part II tests against this model.

Part II

Methods under test

Chapter 4

Linear projections and where they break

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

When is a projection just a shadow?

4.1 What this chapter answers

A linear method asks one question: which flat slice of the ambient space keeps most of the cloud? On a folded sheet the answer is a shadow. The sheet is intrinsically two-dimensional, so an exact two-dimensional answer exists — Chapter 2 writes it down in closed form — but folding puts it out of a projection's reach, because unfolding is not a linear map. This chapter measures how far out of reach, and establishes three things the rest of the book runs on.

The reading convention. Every chapter from here on opens by reading a committed artefact together with the digest it was written against — that is `read_run()`, and this is the chapter that explains what it refuses.

The baseline. PCA's reconstruction-error curve over θ is the reference every later method is judged against. It is not a straw man: PCA is exact on the flat sheet, and on this grid it is the best method of all until the fold is deep enough for Isomap to overtake it — which happens later than the ordering in Chapter 10 might suggest, and later still on Q_{NX} than on reconstruction error. Chapter 5 locates both crossings.

The coincidence. PCA and classical MDS are not two methods. The registry keeps both because showing that they coincide is worth a figure, and Chapter 10 must not report them as two results.

None of this argues against linear projection. Linear methods are the honest baseline here precisely because they cannot represent a fold and say so plainly, rather than returning a picture with structure the data do not have.

4.2 Setup

```
grid <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")
lin  <- subset(grid, method %in% c("pca", "mds"))
p    <- miura_ori(nx = 6L, ny = 6L, alpha = pi / 3)
```

`read_run()` reads a committed `.rds` and refuses three things. An artefact with **no provenance** cannot be traced to the code that produced it, so no number may be reported from it. One written by a **`--quick` run** is a smoke test at `N_QUICK` points, not the book's evidence. And one whose **digest** is not the digest named here is not the run these sentences were read off — the chunks would re-run, the figures would redraw, and the prose around them would quietly keep the old values.

The digest is taken with the provenance block stripped. Provenance carries a commit, a timestamp and an elapsed time, so hashing it would fire the check on every regeneration, and a check that always fires is one that gets deleted. What is hashed is the answer.

```
# The third refusal. The message hands over the digest it found,
 ↪  so recording a
# new one is easier than not recording it.
read_run("data/processed/benchmark-grid.rds", "0000000000000000")
#> Error: data/processed/benchmark-grid.rds is not the run this
 ↪  chapter was
#>   written against.
#>   expected digest: 0000000000000000
#>   found:           2483a468d82d4d64
```

4.3 Background

4.3.1 The projection

PCA is the least-squares fit of a d-dimensional affine subspace to a cloud of points (Pearson, 1901), and equivalently the rotation onto the directions of greatest variance (Hotelling, 1933). Both descriptions are one singular value decomposition, and Eckart–Young (Eckart and Young, 1936) says the truncated decomposition is *optimal* among approximations of that rank rather than merely good. That matters twice below: it is why PCA is the strongest thing a linear method can be, and it is what `metric_floor()` is built from.

`embed_pca()` computes it with `prcomp()`, an SVD of the centred data matrix. The plan this book was written from preferred that route to `princomp()`'s eigendecomposition of the covariance on accuracy grounds. The argument does not reproduce here and is not made: `lin-core` compares the two and they agree to the last representable place. `prcomp()` is a convention, not a result.

4.3.2 The distance version

Classical multidimensional scaling (Torgerson, 1952) starts from a matrix of distances, double-centres their squares, and takes the leading eigenvectors. It is not Kruskal's multidimensional scaling (Kruskal, 1964), which minimises a stress functional numerically; the classical version is a closed-form eigenproblem, and `embed_mds()` is `cmdscale()` on the ambient distances.

For Euclidean distances the double-centred matrix of squares *is* the Gram matrix of the centred coordinates, whose eigenvectors are the principal directions. So classical MDS on Euclidean distances is PCA — a textbook identity (Lee and Verleysen, 2007). What the next section adds is a measurement of it: "provably identical" and "identical in this implementation, at this tolerance, on this data" are different claims, and only the second can be checked.

4.3.3 What is being scored

`reconstruction_error()` is the normalised full Procrustes RMSE. It quotients out translation, rotation, reflection and one isotropic scale (Gower, 1975) and nothing else, so zero means the embedding is the chart up to a similarity and one means it explains none of the chart's variance at all.

`metric_floor()` says when zero is even available: it is the tail of the chart's own eigenvalue spectrum, the Eckart–Young residual of forcing a chart of intrinsic dimension p into $d < p$. Here chart and target are both two-dimensional, so the `floor` column of this grid is zero in every one of its rows. Nothing below is loss the problem imposes; all of it belongs to the method.

4.4 A shadow, and two names for it

4.4.1 Exact on the flat sheet

```
thetas <- THETA_COARSE[c(1L, 3L, 5L)]
samp   <- lapply(thetas, function(t)
  sample_manifold(p, theta = t, n = N_DEFAULT, seed =
   ↪  BENCH_SEEDS[1]))
Y <- lapply(samp, function(s) embed("pca", s))
```

`embed()` takes the sample object, not a matrix. A method may consume the ambient cloud, a neighbourhood graph built from it, or geodesics, and the registry records which; handing it a bare matrix would throw away the distinction the whole benchmark is about.

```
err0       <- reconstruction_error(Y[[1]], samp[[1]]$truth)
floor0     <- metric_floor(samp[[1]])
flat_cells <- subset(grid, method == "pca" & theta == 0 & noise
 ↪  == "none")

sv     <- vapply(samp, function(s) prcomp(s$X)$sdev, numeric(3))
thick  <- sv[3, ] / sv[1, ]          # the corrugation, relative
 ↪  to the sheet
```

```
aspect <- sv[2, ] / sv[1, ]          # of the two directions PCA
 ↪  keeps
u      <- svd(scale(samp[[1]]$truth, scale = FALSE))$d
chart_aspect <- u[2] / u[1]          # of the chart, which
 ↪  folding never touches
```

At $\theta = 0$ the sheet lies in a plane, PCA finds that plane, and the recovery is exact. On this sample the error is 2.2e-16 against a floor of 0, and over the whole clean flat row of the committed grid — 20 seeds — the worst is 1.4e-15, with Q_{NX} exactly 1. That is not a good score. It is floating-point rounding on a problem whose answer is a rigid motion.

It is worth pausing on how unusual that is. At $\theta = 0$ PCA and classical MDS are the only methods on this grid that recover the chart exactly; every method in Chapters 5 and 6 posts a visible error on a sheet that has not been folded at all.

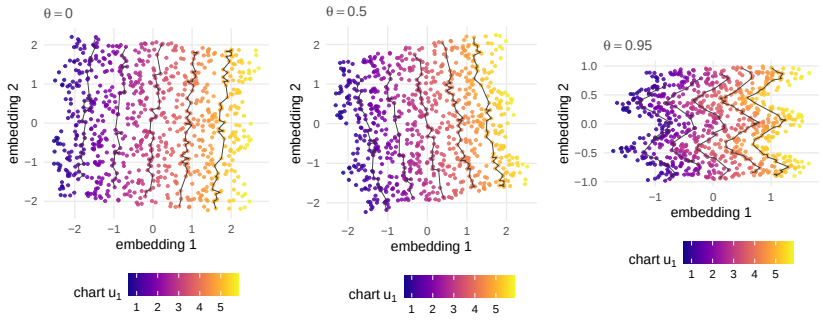


Figure 4.1: The two-component PCA of the same sheet at three fold angles, coloured by the exact chart coordinate along the corrugation. The grey paths are level sets of that coordinate: an embedding that recovered the chart draws them nested and non-crossing, and one that folded the chart back onto itself draws two of them on top of each other.

Figure 4.1 is the chapter's title in three panels. The left panel is the chart, recovered. The right panel is a shadow: the sheet has contracted along its corrugation and the projection has contracted with it, because a projection is the one thing that cannot put back a direction the data no longer extend into. Read the grey level sets rather than the colour. They stay ordered and never cross — PCA does not tear the chart or fold it back on itself — but they bow, and a bowed chart is a distortion no similarity transform can undo.

4.4.2 What the projection discards

Folding lifts the sheet out of its plane, and the projection's difficulty is that the lift is real. The third principal direction of the ambient cloud *is* the corrugation, and its spread relative to the leading direction rises from 0 on the flat sheet to 0.115 and 0.295 at the two folded angles of Figure 4.1. Two components cannot keep it, and by Eckart–Young no other pair of directions would keep more of the cloud, so the residual is not a choice anybody made.

Discarding it would be harmless if what remained were the chart. It is not, and that is the mechanism the reconstruction error picks up. The sheet does not contract equally in its two chart directions: at these three fold angles the pair PCA keeps has aspect ratio 0.861, 0.931 and 0.691, moving first up and then down, while the chart's own aspect ratio is 0.861 at every one of them because folding never touches the chart. What the projection returns is therefore an *anisotropically* rescaled chart — and full Procrustes admits translation, rotation, reflection and exactly one **isotropic** scale. Anisotropy is precisely what the alignment cannot absorb, which is why the score climbs even though the level sets in Figure 4.1 never cross.

4.4.3 One computation, four names

```
w <- reshape(lin[, c("theta", "noise", "seed", "method",
 ↪  "rmse")],
             idvar = c("theta", "noise", "seed"), timevar =
              ↪  "method",
             direction = "wide")
d_eps  <- abs(w$rmse.pca - w$rmse.mds) / .Machine$double.eps
rank_d <- max(abs(lin$qnx[lin$method == "pca"] -
 ↪  lin$qnx[lin$method == "mds"]))

mid    <- samp[[2]]
d_cmds <- reconstruction_error(embed("mds", mid), Y[[2]])
d_prin <- reconstruction_error(princomp(mid$X)$scores[, 1:2],
 ↪  Y[[2]])
d_eig  <- reconstruction_error(
  scale(mid$X, scale = FALSE) %*% eigen(cov(mid$X))$vectors[,
   ↪  1:2], Y[[2]])
```

Across all 1200 cells of the grid, PCA and classical MDS differ in reconstruction error by at most 12.4 units of machine epsilon, and 185 of those cells are bit-identical. On Q_{NX} the difference is 0 exactly — the two methods return configurations whose rank orders agree perfectly.

The same holds for the routes into PCA itself. At θ of 0.5, `cmdscale()`, `princomp()` and an eigendecomposition of the covariance matrix all land on `prcomp()`'s answer at Procrustes distances of 5.7e-16, 2.1e-16 and 2e-16. Four names, one computation. The accuracy argument for `prcomp()` has nowhere to show up on data of this conditioning.

Figure 4.2 is the reason both entries survive in the registry and the reason neither may be counted twice. Chapter 10 reports one linear result, not two, and `R/methods.R` says so at the definition of `embed_mds()` rather than leaving it to be remembered.

4.5 Results

```
pca   <- subset(grid, method == "pca")
```

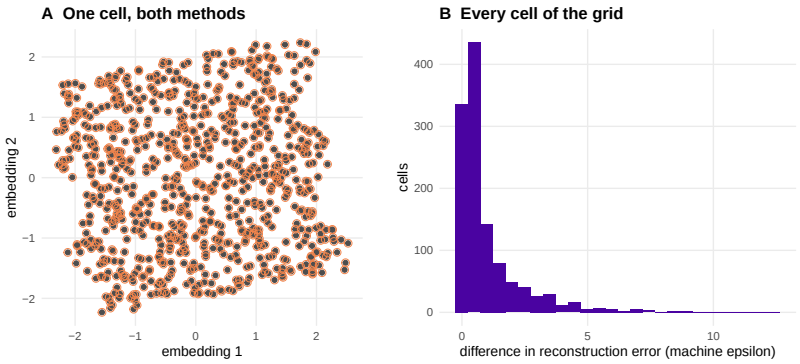


Figure 4.2: PCA and classical MDS on the same ambient cloud. Panel A overplots one cell: filled points are the PCA embedding, open circles the Procrustes-aligned MDS embedding, and every circle is centred on a point. Panel B is the difference in reconstruction error over every cell of the committed grid, in units of machine epsilon.

```
curve <- do.call(rbind, lapply(split(pca, list(pca$noise,
 ↪  pca$theta)),
  function(s) data.frame(noise = s$noise[1], theta = s$theta[1],
                         mean = mean(s$rmse), lo = min(s$rmse),
                         hi = max(s$rmse))))
curve$noise <- factor(curve$noise, levels = c("none", "ambient",
 ↪  "outlier"))
curve <- curve[order(curve$noise, curve$theta), ]

clean  <- subset(curve, noise == "none" & theta > 0)
expo   <- coef(lm(log(mean) ~ log(theta), subset(clean, theta <=
 ↪  0.5)))[[2]]
step   <- clean$mean[-1] / clean$mean[-nrow(clean)]
spread <- max((curve$hi - curve$lo)[curve$noise == "none"])
at     <- function(nz, t) curve$mean[curve$noise == nz &
 ↪  curve$theta == t]
```

Figure 4.3 has no feature in it, and that is the result rather than a disappointment. There is no threshold, no knee and no regime change: PCA degrades smoothly and monotonically from exact to poor, and a benchmark that reported a single fold angle would have found nothing here to report.

The degradation obeys a law. Over the first half of the sweep the clean curve is a power of θ with exponent 1.99 — quadratic, to three figures. Chapter 2 derives why: ambient distance across a crease contracts by $\sin(\rho/2)$, and $1 - \sin(\rho/2)$ is quadratic in the departure from flatness. The error PCA accumulates is that contraction, integrated over the sheet.

Past the middle the curve steepens. The step from the second-largest to the largest fold angle multiplies the error by 1.49 against a median step of 1.24, which is the same endgame acceleration Chapter 2 shows in the sheet's contrac-

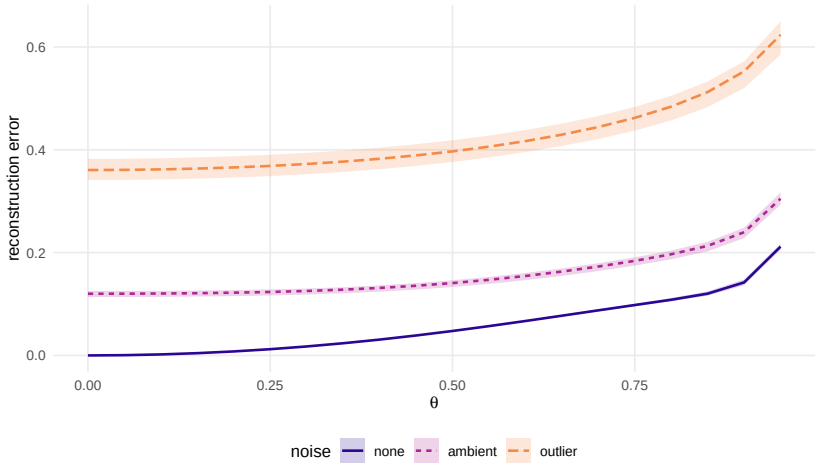


Figure 4.3: The reference curve. Two-component PCA's reconstruction error against the chart, over the fold sweep, for the three noise models. The line is the mean over the twenty benchmark seeds and the band is their full range; the band is narrower than the line except at the last fold angle. The floor is zero everywhere on this grid, so the whole of each curve is recoverable structure that the projection did not recover.

tion. Clean, the error reaches 0.048 at the middle of the sweep and 0.212 at its end.

Seeds contribute almost nothing. The widest range across the twenty benchmark seeds at any clean fold angle is 0.011, and it occurs at the last one; everywhere else the band is thinner than the line drawn over it. The curve is a property of the geometry and not of the draw, which is what makes it usable as a reference.

Noise moves the curve up without changing its shape. At $\theta = 0$ the error is already 0.12 under ambient Gaussian noise and 0.361 under outliers, on a sheet that is flat and whose floor is zero. Those intercepts are the noise, not the fold; the fold's contribution is what rides on top of them.

```
prod <- read_run("data/processed/product-grid.rds",
 ↪  "bb8ade76d88b95a3")
pr2  <- subset(prod, family == "crease" & d == EMBED_DIM & status
 ↪  == "ok")
exc  <- aggregate(cbind(rmse, floor, excess) ~ method, pr2,
 ↪  median)
exc  <- exc[order(exc$excess), ]
```

One number from Chapter 8's product construction belongs here, because a chapter with this title should admit where linear projection emphatically does not break. Pairing two folded sheets gives a chart of intrinsic dimension four whose floor in two dimensions is strictly positive — median 0.514 here — so a method can be scored by how far *above* the floor it lands. PCA's median excess is 0.00508. It is the smallest excess of any method in this book, below

Isomap's 0.0155, on a problem where the optimum is known exactly. When the obstruction is dimension rather than curvature, the optimal linear map is very nearly the optimal map.

4.6 Diagnostics

Two diagnostics a practitioner would reach for do not see the fold, and both would report success on the right-hand end of Figure 4.3.

```
scree <- do.call(rbind, lapply(THETA_GRID, function(t) {
  v <- vapply(BENCH_SEEDS[1:5], function(sd) {
    s   <- sample_manifold(p, theta = t, n = N_DEFAULT, seed =
 ↪  sd)
    lam <- prcomp(s$X)$sdev^2
    c(sqrt(sum(lam[-seq_len(EMBED_DIM)]) / sum(lam)),
      reconstruction_error(embed("pca", s), s$truth))
  }, numeric(2))
  data.frame(theta = t, ambient = mean(v[1, ]), chart = mean(v[2,
   ↪  ]))
}))
```

The scree plot does not collapse. The plan of record predicted that variance explained would collapse as the fold deepens. It does not. At the largest fold angle the first two components still hold 94.3 per cent of the ambient variance — a fall of under six percentage points across the entire sweep, while the chart error grows by more than two orders of magnitude. A reader choosing the number of components from a scree plot would be right about the dimension and uninformed about the geometry.

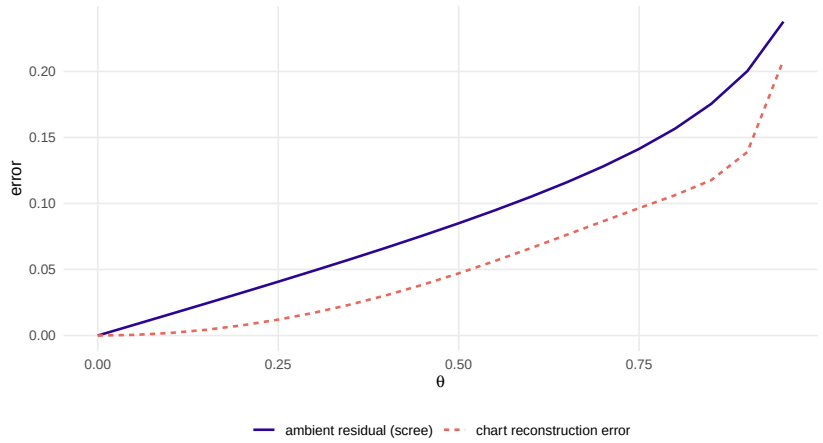


Figure 4.4: The residual a scree plot shows against the error the answer key shows, over the same sweep. Both are dimensionless and both are monotone, but they are not proxies for each other: the scree residual is measured against the ambient cloud and the reconstruction error against the chart, and their ratio is not constant.

Figure 4.4 shows why the two cannot be substituted. The scree residual overstates the chart error by a factor of 16.7 at the smallest non-zero fold angle and 1.14 at the largest: monotone in each other on this one axis, and nowhere near proportional.

Trustworthiness does not see it either. The grid scores trustworthiness and continuity at `K_DEFAULT` against *ambient* distance and Q_{NX} at `K_QNX` against the *chart*. Put the two conventions side by side:

```
tc <- aggregate(cbind(trust, cont, qnx, knn) ~ theta,
                subset(grid, method == "pca" & noise == "none"),
                 ↪  mean)
worst <- tc[nrow(tc), ]
```

At the largest fold angle PCA scores 0.992 on trustworthiness and 0.997 on continuity, while Q_{NX} against the answer key is 0.676 and k-nearest-neighbour preservation is 0.505. An embedding that has lost half its neighbours by the chart's reckoning is rated near-perfect by the ambient one. Neither number is wrong; they answer different questions, which is Chapter 9's subject.

4.7 Where it fails

A projection cannot unfold, and this is not a tuning problem. The optimal subspace is already what PCA returns (Eckart and Young, 1936), so no choice of plane avoids the trade-off in `lin-core`: keep the corrugation and spend a component on it, or discard it and accept an anisotropically squashed chart. Every later method in Part II earns its keep by consuming something other than ambient distance.

No robustness. The outlier intercept in Figure 4.3 is a covariance matrix being dragged by displaced points. PCA has no defence against it and none is offered here; the book reports the cost rather than repairing it, so that later methods meet a plain implementation.

The floor excuses nothing on this grid. `metric_floor()` is zero in every cell, so the whole curve is loss the projection is responsible for. That flips on the product construction, where the floor is large and PCA's excess above it is the smallest in the book. The two results are not in tension: a linear method is close to optimal exactly when the obstruction is dimension, and far from it when the obstruction is curvature.

One family, one size, one α. Everything above is measured on a single Miura tessellation at one opening angle, so the ordering of methods at a given fold angle is a property of this grid; Chapter 10 is where it meets anything else. A benchmark on one family cannot support a claim about pattern variety, and this chapter makes none.

4.8 Reproduce this

The grid is committed, not generated at render time. `scripts/run-benchmark-grid.R` produces it and declares the artefact path in its header; `scripts/check-artefact-producers.R`

checks that every artefact the book specifies has a producer. Re-running that script and reproducing its numbers leaves the digest in `lin-setup` alone; getting different numbers changes it, and the render then stops with both digests in the message.

The rest is checked by the suite, which runs on every push:

- `test-methods.R` — *"the ambient-metric methods recover a flat sheet exactly"* and *"PCA and classical MDS agree, because they are the same thing"*.
- `test-metrics.R` — *"a 2-D chart has no irreducible loss at d = 2"*, *"the floor is attained, exactly, by the top-d principal projection"*, and *"the floor does not move with theta or with noise"*.
- `test-artefacts.R` — *"a chapter cannot render against a run it was not written against"*.

Run them with `Rscript tests/testthat.R` from the repository root.

4.9 Further reading

Pearson (Pearson, 1901) and Hotelling (Hotelling, 1933) are the two original readings of PCA and are still the clearest statement of why they are the same thing; Eckart and Young (Eckart and Young, 1936) supply the optimality that makes it a baseline rather than a heuristic. Torgerson (Torgerson, 1952) is classical scaling's source and Kruskal (Kruskal, 1964) the stress-minimising alternative it is routinely confused with. Gower (Gower, 1975) is the reference for the alignment the score is built on; Chapter 9 owns that convention and is where to go for it.

The bridge into the rest of Part II is kernel PCA (Schölkopf et al., 1998). Replacing the Gram matrix in classical scaling with a kernel makes the same eigenproblem non-linear, and Isomap — Chapter 5 — is exactly that substitution with a kernel built from geodesic distances. Lee and Verleysen (Lee and Verleysen, 2007) set out the whole family in those terms.

Chapter 5

Geodesic methods

R. Heller · Date: 2026-07-29 · Last changed: 2026-10-30

If the data lie on a folded sheet, how do you flatten it back out?

5.1 What this chapter answers

Isomap's premise is this book's premise. Estimate the distances measured *along* the surface, embed those instead of the distances measured *through* the space around it, and a sheet that has been bent without being stretched should come back flat. `R/methods.R` records what every method consumes, and Isomap is the only entry in the registry whose answer to that question is "geodesic distance".

That makes this the one chapter where the method's input and the answer key are the same object. A rigid folding is an intrinsic isometry, so the geodesic distance between two sampled points **is** the Euclidean distance between their rows of `truth`. Not an estimate of it, and not a limit approached as the sample grows: the exact number, in closed form, before any graph is built.

The consequence is that Isomap's error decomposes into two terms that can be measured apart rather than argued about. A **graph term**, from approximating geodesics by shortest paths on a k-nearest-neighbour graph. An **embedding term**, from putting the resulting distance matrix into two dimensions with classical scaling. Hand `cmdscale()` the exact geodesics and the second term vanishes to floating point. Everything that is left belongs to the first.

This chapter measures that first term across the fold sweep, predicts where it turns from a bias into a break, and reports what the break actually is on this object — which is not the mechanism the plan of record predicted.

Diffusion maps is here as the contrast. It sits in the same registry family and consumes *ambient* distance, and the gap between the two curves is the value of consuming the right thing.

5.2 Setup

```
grid   <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")
sweeps <- read_run("data/processed/part2-sweeps.rds",
 ↪  "16d43ff4992c0e58")
p <- miura_ori(nx = 6L, ny = 6L, alpha = pi / 3)
s <- sample_manifold(p, theta = 0.5, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
```

The main grid fixes k at `K_DEFAULT`; `part2-sweeps.rds` is the artefact that moves it, over 7 values at 3 fold angles and all 20 benchmark seeds. Chapter 4 explains what `read_run()` refuses.

5.3 Background

5.3.1 Isomap, in three stages

Isomap (Tenenbaum et al., 2000) is classical multidimensional scaling (Torgerson, 1952) with the distance matrix replaced. Build a k-nearest-neighbour graph on the ambient points, weight each edge by its ambient length, take all-pairs shortest paths, and hand the result to `cmdscale()`. The first two stages estimate the geodesic metric; the third is the same eigenproblem Chapter 4 already ran. Read as a kernel method (Schölkopf et al., 1998; Lee and Verleysen, 2007) it is classical scaling with a geodesic kernel, which is exactly how Chapter 4 hands the reader over.

The estimate is justified asymptotically. Graph shortest paths converge to true geodesics as the sample density rises, at a rate controlled by the sampling radius and the surface's curvature (Bernstein et al., 2000). Both stages of that guarantee are worth naming here, because both fail on a crease pattern: the convergence is stated for a manifold of positive reach, and a crease has reach exactly zero (Chapter 2), and it is asymptotic in a density this book fixes at `N_DEFAULT` points.

```
reg <- method_table()
fam <- reg[reg$family == "geodesic", c("method", "label",
 ↪  "consumes", "stochastic")]
knitr::kable(fam, row.names = FALSE,
             col.names = c("registry name", "label", "consumes",
              ↪  "stochastic"),
             caption = "The geodesic family of the embedding
              ↪  registry.")
```

Table 5.1 is the chapter in one row each. Isomap consumes geodesic distance and is deterministic. Diffusion maps consumes **ambient** distance and is stochastic, and both of those are easy to get wrong.

Table 5.1: The geodesic family of the embedding registry.

registry name	label	consumes	stochastic
isomap	Isomap	geodesic	FALSE
diffusion	Diffusion map	ambient	TRUE

5.3.2 Diffusion maps, and why it is filed here

Diffusion maps (Coifman and Lafon, 2006) puts a Gaussian kernel on the ambient distances, normalises it into a Markov chain, and embeds the points so that Euclidean distance in the embedding approximates *diffusion distance* — the difference between the transition distributions of a random walk started at each point. Diffusion distance is a geodesic-flavoured quantity, which is why the family label is what it is, but nothing in the construction estimates a geodesic. The kernel sees ambient distance and only ambient distance, and the registry's `consumes` field says so. The same holds for Laplacian eigenmaps (Belkin and Niyogi, 2003), which Chapter 6 takes up.

Two implementation facts travel with it. It is **stochastic** — `diffusionMap::diffuse()` uses a randomised eigensolver — so `embed()` refuses to run it without a seed. And its bandwidth is set from the k-th nearest-neighbour distance rather than the median pairwise distance, because the outlier noise model inflates the second and not the first; `R/methods.R` records what that cost when it was done the obvious way.

> **Two changes from the plan of record.** The outline for this chapter named `igraph::distances()` for the shortest paths and locally linear embedding as a third method. Neither survived. `igraph` is not installable in this project — a missing system library — so `R/metrics.R` computes all-pairs shortest paths by min-plus relaxation with a work list, which is Bellman–Ford rather than Dijkstra [@dijkstra1959note] and was timed against two alternatives at the book's sample size before it was kept. LLE consumes neighbourhood structure, not geodesics, and is in the neighbor embeddings chapter with the other methods that do.

5.4 The geodesic you do not have to estimate

5.4.1 The claim, and its one condition

Folding preserves distance along the surface exactly (Chapter 2). So the length of a path on the folded sheet equals the length of its preimage in the flat chart, and the shortest path in the chart between two points is the straight segment between them. Therefore `dist(s$truth)` is the geodesic distance matrix — not a discretisation of it.

The condition is that the straight segment stays on the paper. A Miura's unfolded outline is not convex, so this is a real condition and not a formality; `sample_manifold()`'s default `boundary = FALSE` trims the sample to a convex sub-rectangle, and the trimming is what makes the equality hold rather than a

bound.

```
dU   <- reference_dist(s, "chart")
dA   <- reference_dist(s, "ambient")
same <- outer(s$facet, s$facet, "==")
up   <- upper.tri(dU)

within  <- max(abs(dA - dU)[same & up])          # isometry,
 ↪  inside a facet
across  <- max((dA - dU)[!same & up])            # contraction,
 ↪  across a crease
exit    <- chart_exit_fraction(s, p)             # segments that
 ↪  leave the sheet
```

Three checks, and each one is a different half of the claim. Inside a facet ambient and chart distance agree to 1.1e-15, which is rounding on a closed form. Across a crease the ambient distance is strictly smaller — the largest signed difference over all 308,887 such pairs is -6.2e-07, a negative number. And `chart_exit_fraction()` reports that 0 of sampled segments leave the sheet, so the chart distance is the geodesic for every pair in this sample and not merely a lower bound on it.

5.4.2 The decomposition

```
Y_exact <- cmdscale(as.dist(dU), k = EMBED_DIM)           # exact
 ↪  geodesics in
dG      <- reference_dist(s, "graph", k = K_DEFAULT)      # what
 ↪  Isomap estimates
Y_iso   <- embed("isomap", s, k = K_DEFAULT)

err_embed <- reconstruction_error(Y_exact, s$truth)
err_total <- reconstruction_error(Y_iso,   s$truth)
agree     <- max(abs(unname(Y_iso) - unname(cmdscale(as.dist(dG),
 ↪  k = EMBED_DIM))))
graph_dev <- sqrt(mean(((dG - dU)[up])^2)) /
 ↪  sqrt(mean((dU[up])^2))
```

Feed classical scaling the exact geodesic distances and it returns the chart, at a Procrustes error of 5.7e-16. That is the embedding term, and it is zero: the chart is a two-dimensional Euclidean configuration, so double-centring its squared distances and taking two eigenvectors inverts the construction exactly.

Isomap on the same sample scores 0.0266. It differs from `cmdscale()` on the graph distances by 7.9e-15, because it *is* `cmdscale()` on the graph distances. So the whole of that 0.0266 is the graph term, and the graph term is small in the distances themselves — the shortest-path matrix departs from the exact geodesics by 2.95 per cent in relative root-mean-square.

This is the sense in which the chapter's title is not a metaphor. Isomap's error on a crease pattern has one cause, it is a property of a graph rather than of an optimiser, and it can be looked at directly.

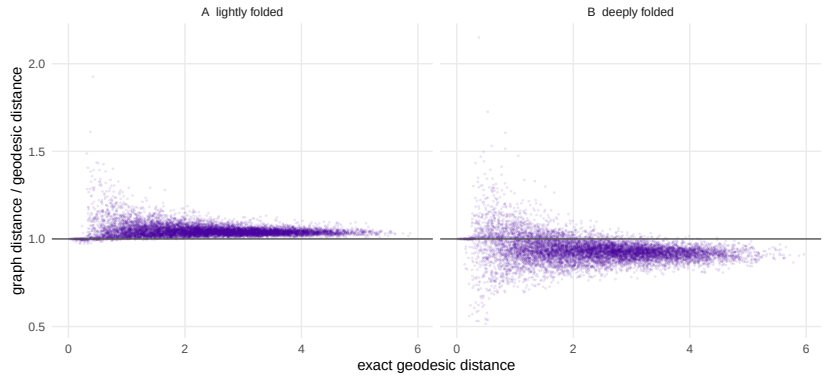


Figure 5.1: Shortest-path distance on the ambient neighbourhood graph against the exact geodesic distance, as a ratio, for a lightly folded sheet and a deeply folded one. The horizontal line at one is exactness. Points above it are the discretisation bias of walking along graph edges; points below it are distances the fold has contracted. A random ten thousand pairs are drawn in each panel.

Figure 5.1 is the graph term seen twice. At a small fold angle it is a *positive* bias: a path made of graph edges zigzags where a geodesic runs straight, so shortest paths come out too long, and the scatter collapses onto a tight band just above one as the pair gets further apart and the zigzag averages out. At a large fold angle that band has dropped below the line. Edges that cross a crease are shortened by the $\sin(\rho/2)$ factor Chapter 2 derives, shortest paths route through as many creases as they can, and the estimate falls below the truth. The two biases are different mechanisms with opposite signs, and `geo-diagnostics` is where they cancel.

5.5 Results

```
clean <- subset(grid, noise == "none" & ran)
cur   <- aggregate(cbind(rmse, qnx) ~ method + theta, clean,
 ↪  mean)
band  <- aggregate(rmse ~ method + theta, clean,
                   function(x) c(lo = min(x), hi = max(x)))
band  <- data.frame(band[, c("method", "theta")], band$rmse)
cur   <- merge(cur, band, by = c("method", "theta"))
cur   <- subset(cur, method %in% c("isomap", "diffusion", "pca"))

at <- function(m, t, col = "rmse") cur[[col]][cur$method == m &
 ↪  cur$theta == t]
wide <- reshape(cur[, c("theta", "method", "rmse", "qnx")],
                idvar = "theta", timevar = "method", direction =
                 ↪  "wide")
wide <- wide[order(wide$theta), ]
cross_rmse <- wide$theta[which(wide$rmse.isomap <
 ↪  wide$rmse.pca)[1]]
```

```
cross_qnx  <- wide$theta[which(wide$qnx.isomap  >
 ↪  wide$qnx.pca)[1]]
noisy <- aggregate(rmse ~ noise + theta, subset(grid, method ==
 ↪  "isomap" & ran),
                   mean)
```

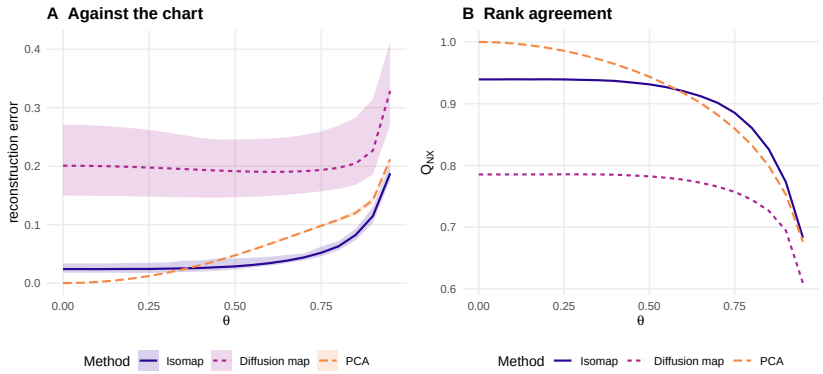


Figure 5.2: Isomap and diffusion maps over the fold sweep, with PCA from Chapter 4 as the reference. Panel A is reconstruction error against the chart, panel B is Q_{NX} at K_{QNX} against the chart. Lines are means over the twenty benchmark seeds and bands in panel A are their full range. Lower is better in A, higher in B.

Three readings of Figure 5.2, in order of how much they cost the reader to accept.

Isomap wins, eventually, and pays an entry fee for it. On the flat sheet its error is 0.024 where PCA's is zero: the graph term is there at $\theta = 0$, because a neighbourhood graph is a discretisation whether or not anything has been folded. Isomap does not overtake PCA on reconstruction error until $\theta = 0.4$, and on Q_{NX} not until $\theta = 0.6$. From 0.4 to the end of the sweep it is the best method in the book on this grid, and at the last fold angle it scores 0.188 against PCA's 0.212.

That the two metrics put the crossing in different places is not a discrepancy to be resolved. Reconstruction error reads distances and Q_{NX} reads rank order, and Chapter 9 is where the disagreement is the subject rather than a nuisance.

The curve has a shape, and the shape is the point. Isomap's error is essentially flat through the first half of the sweep — it grows by a factor of 1.19 between the flat sheet and the middle — and then rises by a factor of 6.58 over the second half. A benchmark reporting one fold angle would have called this method robust or fragile depending entirely on which angle it chose.

Diffusion maps does not improve on PCA anywhere. Its error is close to flat across the whole sweep, from 0.201 to 0.329, and it is the *worst* of the three at 20 of the 20 fold angles in the sweep. A method that consumes ambient distance and embeds a random walk on it has no mechanism for undoing a fold; it is in this chapter to show that the family label is not what does the work.

Noise moves Isomap's curve without changing it. At $\theta = 0$ the error is already 0.124 under ambient Gaussian noise and 0.416 under outliers, on a sheet that is not folded at all. Displaced points are new neighbours, and a graph method's whole input is who is a neighbour.

5.6 When does the graph bridge?

A **fold jump** is a k-nearest-neighbour edge between two points the sheet holds far apart — the classical short circuit that destroys Isomap on tightly wound surfaces (Balasubramanian and Schwartz, 2002). It is the failure this chapter was specified to predict, and the prediction can be made without running anything.

Two quantities cross. `facet_gap(p, theta)` is the exact minimum ambient distance between two facets that share no vertex: a property of the surface alone, no sample and no density (Chapter 2 uses it to check that the sheet never passes through itself). Against it stands the k-nearest-neighbour radius r_k in the chart. That radius is a constant of the sweep, not a function of θ: `sample_manifold()` draws the chart points first and folds them after, so for a fixed seed `s$truth` is the *same matrix* at every fold angle. The graph acquires its first fold jump when the surface's own clearance falls below the radius at which the sample connects points.

```
KS   <- c(10L, 20L, 30L, 45L)
s0   <- sample_manifold(p, theta = 0, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
dU0  <- as.matrix(dist(s0$truth))
Lfar <- 0.25 * max(dU0)                          # branch_gap()'s
 ↪  "far apart"
d0   <- dU0; diag(d0) <- Inf
r_k  <- vapply(KS, function(k)
  median(apply(d0, 1L, function(r) sort.int(r, partial = k)[k])),
   ↪  numeric(1))

th_fine <- seq(0.70, 0.99, by = 0.01)            # deliberately past
 ↪  THETA_GRID
gap     <- vapply(th_fine, function(t) facet_gap(p, t)$gap,
 ↪  numeric(1))
gapfun  <- stats::approxfun(th_fine, gap)
predicted <- vapply(r_k, function(r) tryCatch(
  stats::uniroot(function(t) gapfun(t) - r, range(th_fine))$root,
  error = function(e) NA_real_), numeric(1))

jumps <- function(theta, seed) {
  st <- sample_manifold(p, theta = theta, n = N_DEFAULT, seed =
 ↪  seed)
  nn <- FNN::get.knn(st$X, k = max(KS))$nn.index
  du <- as.matrix(dist(st$truth))
  vapply(KS, function(k) sum(du[cbind(rep(seq_len(nrow(nn)),
   ↪  times = k),
                                      as.vector(nn[,
                                       ↪  seq_len(k)]))] > Lfar),
```

Table 5.2: Predicted and measured onset of the first fold jump.

k	chart radius	predicted onset	measured onset	edges at flat
10	0.292	0.974	0.971	0
20	0.413	0.945	0.944	0
30	0.507	0.909	0.914	0
45	0.629	0.832	0.855	0

```
        numeric(1))
}
counts <- t(vapply(th_fine, function(t)
  rowMeans(vapply(BENCH_SEEDS, function(sd) jumps(t, sd),
   ↪  numeric(length(KS)))),
  numeric(length(KS))))
control <- rowMeans(vapply(BENCH_SEEDS, function(sd) jumps(0,
 ↪  sd),
                           numeric(length(KS))))
measured <- vapply(seq_along(KS), function(a) {
  i <- which(counts[, a] >= 1)[1]
  if (is.na(i)) NA_real_
  else if (i == 1L) th_fine[1]
  else stats::approx(counts[(i - 1):i, a], th_fine[(i - 1):i],
   ↪  xout = 1)$y
}, numeric(1))
onset <- data.frame(k = KS, r_k = r_k, predicted = predicted,
                    measured = measured, at_flat = control)
```

An edge counts as a fold jump when its endpoints are more than `Lfar` apart in the chart, which is `branch_gap()`'s own definition of "far apart along the surface" reused so that the chapter and the difficulty statistic mean the same thing. The onset is where the *expected* number of such edges reaches one, because the first jump is a tail event and one edge in one seed is not an onset.

The fold sweep here runs past `max(THETA_GRID)`, deliberately and uniquely in this book. Whether the short circuit arrives before or after the benchmark's own ceiling is exactly the question, so the measurement may not be confined to the answer.

The criterion calibrates itself. At $\theta = 0$ the sheet is flat, ambient and chart distance coincide, and any edge counted there is sparse sampling rather than folding. That control is 0 at every k tested, so the counts below have no false positives in them.

Table 5.2 is the chapter's headline promise, kept. The analytic onset and the measured one agree to 0.023 in θ at worst — under half a step of the benchmark's own grid — over four neighbourhood sizes spanning a factor of 4.5. The residual has the sign it should: `facet_gap()` is a minimum over the whole of two facets, and a finite sample does not put points exactly at the closest-approach spot, so the measurement is late.

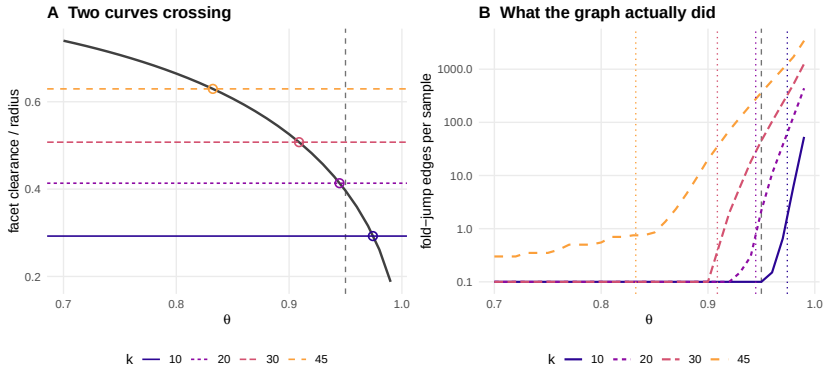


Figure 5.3: The analytic onset against the measured one, over a fold sweep deliberately extended past the benchmark's own ceiling — the vertical dashed line at the largest fold angle the grid uses. Panel A is the exact clearance between non-touching facets as the sheet closes, with the chart k-nearest-neighbour radius for four neighbourhood sizes drawn as horizontal lines; each crossing is a predicted onset, marked with an open circle. Panel B is the measured number of fold-jump edges per sample, averaged over the twenty benchmark seeds, with the same predictions marked on the horizontal axis.

The row that matters is the first one, and it is a finding about the benchmark rather than about Isomap.

> **The committed grid never enters the short-circuit regime.** At k = `K_DEFAULT` — the neighbourhood size every cell of `benchmark-grid.rds` runs at — the predicted onset lies past `max(THETA_GRID)`, and the measured count inside the sweep never leaves zero. The classical failure that geodesic methods are known for is not what this benchmark, as committed, measures.

The count inside the committed sweep is 0 at every fold angle in it, over all 20 seeds. The two boundaries are independent, which is what makes that worth saying rather than assuming. `THETA_GRID` stops where it does because at $\theta = 1$ the sheet has no extent in one direction and every method would be embedding a collapsed object — a reason with nothing to do with neighbourhood graphs (`R/constants.R` states it). The short circuit at k = `K_DEFAULT` arrives at 0.974 predicted and 0.971 measured, which is 0.021 past that ceiling. Nothing was arranged; the grid simply stops just short.

So everything in Figure 5.2 — the whole of Isomap's degradation from 0.024 to 0.188 across the sweep — happened without a single fold jump. Raising k is what crosses the boundary inside the grid, and that is the subject of the next section.

5.7 Diagnostics

If the short circuit is absent, what is doing the damage? The k sweep answers it, and the answer is that a neighbourhood size is not a setting to be tuned once.

```
sw   <- aggregate(cbind(rmse, qnx) ~ method + k_tck + theta,
                  subset(sweeps, ran), mean)
best <- do.call(rbind, lapply(split(subset(sw, method ==
 ↪  "isomap"),
                                     subset(sw, method ==
                                      ↪  "isomap")$theta),
                               function(d) d[which.min(d$rmse),
                                ↪  ]))
dif  <- subset(sw, method == "diffusion")
iso  <- subset(sw, method == "isomap")
ths  <- sort(unique(sweeps$theta))
kmax <- max(sweeps$k_tck)
kbest <- function(t) best$k_tck[best$theta == t]
err   <- function(t, k) iso$rmse[iso$theta == t & iso$k_tck == k]
carry <- function(from, to) err(to, kbest(from)) / err(to,
 ↪  kbest(to))
```

The neighbourhood size that minimises Isomap's reconstruction error falls as the sheet closes: 45 at $\theta = 0.2$, 20 at 0.5, and 12 at 0.8. Neither end's choice survives the other. Carrying the shallow optimum to the deepest fold multiplies the error by 1.51; carrying the deep optimum back to the shallowest multiplies it by 3.81. There is no one k for this pattern, and the relative cost of the wrong one is larger at the shallow end — where the absolute error is smaller by an order of magnitude, so the two ends are not simply harder and easier versions of one problem.

```
KB   <- c(5L, 10L, 20L, 45L)
bias <- do.call(rbind, lapply(ths, function(t) {
  st <- sample_manifold(p, theta = t, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
  du <- reference_dist(st, "chart"); u <- upper.tri(du)
  do.call(rbind, lapply(KB, function(k)
    data.frame(theta = t, k = k,
               bias = median(((reference_dist(st, "graph", k = k)
                ↪  - du) / du)[u])))))
}))
zero <- vapply(split(bias, bias$theta), function(d)
  tryCatch(exp(stats::approx(d$bias, log(d$k), xout = 0)$y),
   ↪  error = function(e) NA_real_),
  numeric(1))
```

Figure 5.4 puts the mechanism next to its consequence. Panel B is the two biases of Figure 5.1 as functions of k. Raising k shortens the zigzag and kills the positive bias; it also lengthens every edge, and a longer edge crosses more creases and is contracted by more of them. The crossing point — where the median shortest path is neither long nor short — falls from $k \approx 35$ at $\theta = 0.2$ to 14 at 0.5 and 6.8 at 0.8, in the same order and over nearly the same range as the error minima in panel A. The two do not coincide, and should not: panel A minimises a Procrustes fit that pays for variance as well as bias.

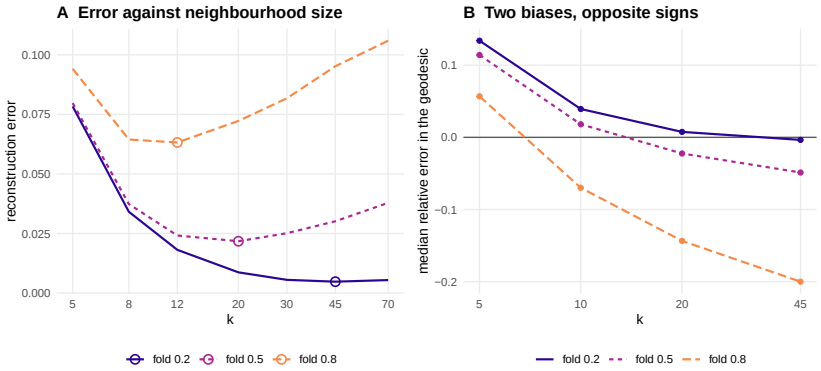


Figure 5.4: Why the optimum moves. Panel A is Isomap's reconstruction error against neighbourhood size at three fold angles, from the committed sweep; the open circles mark each curve's minimum. Panel B is the median relative departure of the shortest-path distance from the exact geodesic, measured at the same three fold angles: positive is the discretisation bias of walking along edges, negative is the fold contracting them, and each curve crosses zero at a smaller neighbourhood size than the last.

Diffusion maps has no such trade-off. Its error falls monotonically in k at all three fold angles, from 0.208 at $k = 5$ to 0.132 at $k = 70$ in the middle of the sweep, because k there sets a kernel bandwidth and not a graph. It never approaches Isomap's optimum, and its best setting is the largest one offered — a curve still falling at the edge of the sweep is a sweep that stopped too early, and this one is reported as such.

Two housekeeping facts from the artefacts, both of which would have been warnings scrolling past in a long run. Isomap's `k_effective` column equals the requested k in every one of the 1200 main-grid rows and every sweep row, so the neighbourhood graph never disconnected and never needed repairing. Diffusion maps needed a widened kernel in 399 of 1200 cells and declined outright in 1; those are recorded in the `tuning` and `status` columns rather than folded into the mean.

5.8 Where it fails

The predicted failure is not the observed one. This chapter was specified to pair an analytic short-circuit onset with a measured one, and it does — to 0.023 in θ. What the specification also assumed, and what is false, is that the short circuit is what breaks Isomap on this object. At the grid's neighbourhood size it never occurs inside the sweep, and Isomap's error still rises by a factor of 7.82 across it. The mechanism that actually operates is edge contraction: present at every $\theta > 0$, graded rather than catastrophic, and invisible to any diagnostic built to look for a bridge. A reader who arrives expecting Isomap to collapse here will not see it collapse — it is the best method on this grid above $\theta = 0.4$ — and the reason is that the benchmark's difficulty axis stops short of the regime that would do it.

When the bridge does form, it is not graceful. Shortest path takes a *minimum* over routes, so a single bad edge is not averaged against the good ones — it is preferred by every pair whose route it shortens.

```
z   <- pair[[2]]                                  # the deeply
 ↪  folded sample
dAf <- as.matrix(dist(z$s$X))
adj <- .knn_graph(dAf, K_DEFAULT)                 #
 ↪  reference_dist()'s internals,
G0  <- .apsp_minplus(adj, dAf)                    #   reached
 ↪  directly so that the
far <- as.matrix(dist(z$s$truth)) > Lfar          #   graph can
 ↪  be edited between

worst <- dAf; worst[!far] <- Inf
ij  <- arrayInd(which.min(worst), dim(worst))
edges <- sum(adj) / 2
adj[ij[1], ij[2]] <- adj[ij[2], ij[1]] <- TRUE
G1  <- .apsp_minplus(adj, dAf)

u2      <- upper.tri(G0)
touched <- mean((G1 < G0 - TOL$default)[u2])
worst_p <- max(((G0 - G1) / G0)[u2])
before  <- reconstruction_error(cmdscale(as.dist(G0), k =
 ↪  EMBED_DIM), z$s$truth)
after   <- reconstruction_error(cmdscale(as.dist(G1), k =
 ↪  EMBED_DIM), z$s$truth)
```

One edge is added to a graph that already has 4,692 of them — the single closest ambient approach between two points the chart holds far apart, which is exactly the pair `branch_gap()` reports. It shortens the shortest path for 5.15 per cent of all pairs, by up to 38.9 per cent, and raises the reconstruction error from 0.0621 to 0.0767. Nothing about the sample changed; one entry of an adjacency matrix did.

Figure 5.5 is what those errors look like. Neither panel is a failure a reader would spot by eye, which is the argument for having an answer key: the deeply folded embedding is 2.6 times worse than the lightly folded one and looks like a slightly wobblier version of it.

The estimate has no guarantee here. The convergence of graph distances to geodesics is stated for a manifold of positive reach and a density that grows (Bernstein et al., 2000; Niyogi et al., 2008). A crease has reach zero and this book fixes n at `N_DEFAULT`. What replaces the guarantee is the measurement above, and that is a weaker thing than a theorem.

One family, one size, one density. Every onset in Table 5.2 is a statement about a six-by-six Miura at `N_DEFAULT` points. The radius r_k falls as $\sqrt{k/n}$, so the same prediction at four times the sample size moves every onset later, and the sweep that would establish that has not been run.

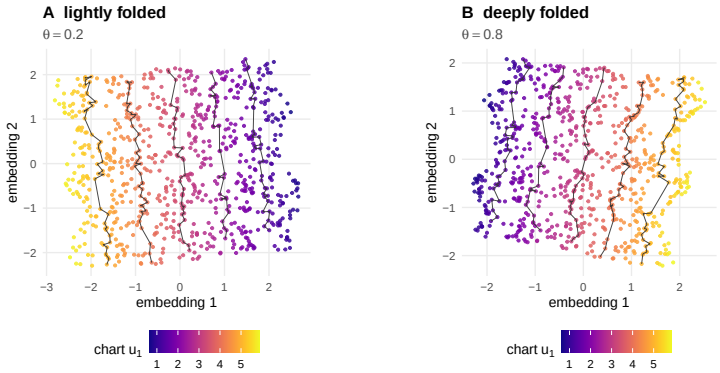


Figure 5.5: What Isomap returns, coloured by the exact chart coordinate along the corrugation, with grey level sets of that coordinate drawn over the points. Panel A is the lightly folded sheet and panel B the deeply folded one. Both recover the chart; the damage in B is a crowding of the level sets, not a tear.

5.9 Reproduce this

Both artefacts are committed and neither is generated at render time. `scripts/run-benchmark-grid.R` produces `benchmark-grid.rds` and `scripts/run-part2-sweeps.R` produces `part2-sweeps.rds`; each declares its artefact path in its header, and `scripts/check-artefact-producers.R` checks that every artefact the book specifies has one. The digests in `geo-setup` are what stop these sentences from being read against a different run.

The claims the chapter rests on are assertions in the test suite:

- `tests/testthat/test-folding.R` — facet isometry across the sweep, and `facet_gap()` against a brute-force point sample.
- `tests/testthat/test-metrics.R` — `reference_dist()`'s three geometries, the neighbourhood graph's connectivity repair, and the shortest paths against a reference implementation.
- `tests/testthat/test-methods.R` — the registry contract, including that a stochastic method called without a seed is an error.

Run them with `Rscript tests/testthat.R` from the repository root.

5.10 Further reading

Tenenbaum, de Silva and Langford (Tenenbaum et al., 2000) is Isomap's source and still the clearest statement of the three-stage construction. The convergence of graph distances to geodesics is Bernstein and co-authors (Bernstein et al., 2000); read it together with Balasubramanian and Schwartz's comment (Balasubramanian and Schwartz, 2002), which is where the topological instability this chapter measures was first named, and with Niyogi, Smale and Weinberger (Niyogi et al., 2008) for what a sampling guarantee needs in order to exist.

Coifman and Lafon (Coifman and Lafon, 2006) introduce diffusion maps and the diffusion distance the embedding approximates; Belkin and Niyogi (Belkin and Niyogi, 2003) is the closely related Laplacian construction Chapter 6 takes up. Lee and Verleysen (Lee and Verleysen, 2007) set out the whole family as one kernel eigenproblem, which is the framing that makes classical scaling (Torgerson, 1952), kernel PCA (Schölkopf et al., 1998) and Isomap three settings of one method rather than three methods. Dijkstra (Dijkstra, 1959) is where shortest paths on a weighted graph begin, though `R/metrics.R` does not use his algorithm and says why.

Chapter 6

Neighbour embeddings

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

Why do t-SNE and UMAP disagree about the same data?

6.1 What this chapter answers

The four methods in this chapter — LLE, Laplacian eigenmaps, t-SNE and UMAP — do not try to recover the unfolding. That sentence has to come before any number, because the rest of the chapter scores them against the unfolding and a reader who meets the scores first will read them as a verdict on the methods rather than on the pairing.

What each of them minimises is a discrepancy between **neighbourhoods**: which points are near which, in the ambient cloud and in the picture. None of them carries a term that asks for a distance to be reproduced. The registry records this as a field rather than as a remark — `consumes = "neighbourhood"` — so a later chapter cannot forget it. `reconstruction_error()` is a Procrustes distance to the chart, and asking these four to be small on it is asking them for something they were never built to deliver.

Then this chapter judges them by it anyway, for two reasons. The reader will: every one of these methods is used in practice as though its output were a map, and a map is a claim about distance. And the benchmark can, which is the rarer thing — the answer key exists in closed form, so the question "how much of the geometry survived?" has an exact answer rather than an argument. What comes out is not that the methods are bad. It is that the setting on which they are most often tuned, and the picture that tuning produces, move in the *opposite* direction to accuracy.

6.2 Setup

```
grid   <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")
```

Table 6.1: The four registry entries this chapter describes.

label	consumes	stochastic
LLE	neighbourhood	FALSE
Laplacian eigenmaps	neighbourhood	FALSE
t-SNE	neighbourhood	TRUE
UMAP	neighbourhood	TRUE

```
sweeps <- read_run("data/processed/part2-sweeps.rds",
 ↪  "16d43ff4992c0e58")
budget <- read_run("data/processed/seed-budget.rds",
 ↪  "e5f73dce72d53066")

NBR  <- c("lle", "laplacian", "tsne", "umap")
pat  <- miura_ori(nx = 6L, ny = 6L)
cell <- sample_manifold(pat, theta = 0.5, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
```

Three committed artefacts. `benchmark-grid.rds` sweeps θ at fixed neighbourhood size; `part2-sweeps.rds` sweeps the neighbourhood size at three fixed θ; `seed-budget.rds` is the measurement standing rule 1's seed floor is checked against. `cell` is one sample, drawn once, used for every figure that shows an embedding rather than a curve.

6.3 Background

```
knitr::kable(
  method_table()[method_table()$chapter == 6L,
                 c("label", "consumes", "stochastic")],
  row.names = FALSE,
  caption = "The four registry entries this chapter describes.")
```

LLE (Roweis and Saul, 2000) writes each point as a weighted combination of its k nearest neighbours and then finds the low-dimensional configuration that keeps those weights. **Laplacian eigenmaps** (Belkin and Niyogi, 2003) takes the bottom eigenvectors of the graph Laplacian of a heat-kernel-weighted neighbourhood graph. Both are spectral, deterministic, and solve a single eigenproblem.

t-SNE (van der Maaten and Hinton, 2008) converts ambient distances into a neighbour-probability distribution, converts embedding distances into a second one with a heavy-tailed kernel, and minimises the Kullback–Leibler divergence between them by gradient descent from a random initialisation. **UMAP** (McInnes et al., 2018) builds a fuzzy simplicial set on the neighbourhood graph and minimises a cross-entropy, also from a random start. Both are stochastic, and the registry refuses to run them unseeded:

```
unseeded <- tryCatch(embed("tsne", cell), error =
 ↪  conditionMessage)
cat(strwrap(unseeded, width = 72), sep = "\n")
```

```
#> tsne is stochastic and was called without a seed. Standing rule 1:
#> every stochastic result in this book is reported across BENCH_SEEDS,
#> and an unseeded fit cannot be one of them.
```

That error is deliberate. Standing rule 1 says every stochastic result in this book is reported across `BENCH_SEEDS`, and an unseeded fit cannot be one of them — it is an anecdote with no way of being labelled as one afterwards. The cheapest place to enforce a reporting rule is at the call that would break it.

> Two neighbourhood parameters, two names, one quantity. t-SNE's is *perplexity*, a smooth measure of the effective number of neighbours; UMAP's is `n_neighbors`. `R/methods.R` maps the registry's `k` onto both, so a sweep over `k` is a sweep over perplexity for one and over `n_neighbors` for the other, and the curves are comparable. A previous version of `embed_tsne()` declared `k` and passed a fixed perplexity — every `k` returned the same embedding, and a flat curve would have been read as insensitivity to neighbourhood size.

6.4 What the methods produce, and what it costs

```
fit <- lapply(NBR, function(m)
  embed(m, cell, k = K_DEFAULT,
        seed = if (isTRUE(METHOD_REGISTRY[[m]]$stochastic))
         ↪  BENCH_SEEDS[1]))
names(fit) <- NBR

Dchart   <- reference_dist(cell, "chart")
cell_err <- vapply(fit, reconstruction_error, numeric(1), truth
 ↪  = cell$truth)
cell_qnx <- vapply(fit, qnx, numeric(1), reference = Dchart, K =
 ↪  K_QNX)
```

Figure 6.1 is one cell of the grid — a Miura at $\theta = 0.5$, 800 points, no noise — seen four ways. LLE returns the chart to within 0.092; UMAP to 0.18; Laplacian eigenmaps to 0.22; t-SNE, at this neighbourhood size, to 0.63, which is most of the way to the score a random configuration earns.

The Q_{NX} column tells a different story about the same four pictures. Scored at $K = 20$ **against the chart**, they run 0.911, 0.757, 0.767 and 0.755. t-SNE, last by Procrustes distance by a factor of 3.5, is *ahead* of UMAP on rank preservation. That is not a contradiction and not a defect in either metric: one reads distances and the other reads rank order, and an embedding that shatters the sheet into islands can keep every local rank while destroying every distance between islands.

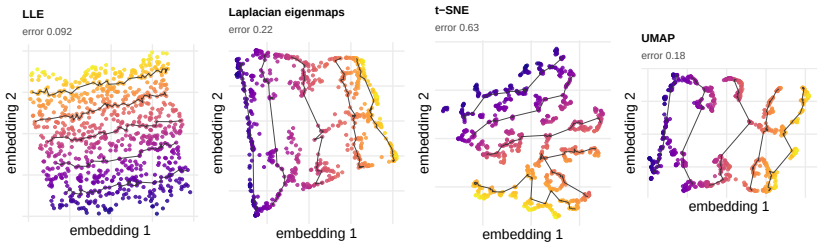


Figure 6.1: One folded sample, embedded four ways. Colour is the first chart coordinate and the black paths are level sets of the second, so a faithful embedding shows a smooth colour ramp crossed by non-intersecting paths. The Procrustes error against the chart is printed under each panel. The axes carry no numbers because an embedding is defined only up to a similarity and its units mean nothing.

6.4.1 The deception

The parameter these methods are tuned on is the neighbourhood size, and tuning is done by looking. That is the part exact truth is able to expose.

```
KS       <- c(5L, 12L, 30L, 70L)
tperp    <- lapply(KS, function(kk) embed("tsne", cell, k = kk,
                                          seed = BENCH_SEEDS[1]))
perp_err <- vapply(tperp, reconstruction_error, numeric(1), truth
 ↪  = cell$truth)
perp_qnx <- vapply(tperp, qnx, numeric(1), reference = Dchart, K
 ↪  = K_QNX)
```

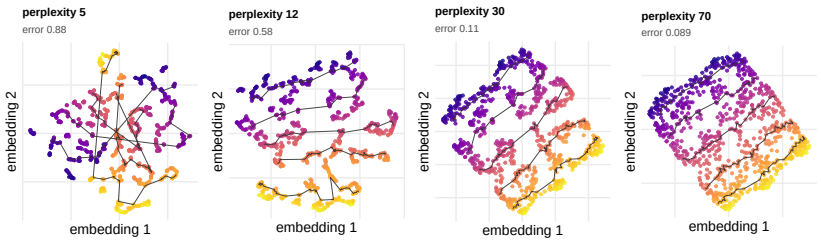


Figure 6.2: t-SNE on one sample at four perplexities. Left to right the picture loses every visible feature and gains an order of magnitude in accuracy. Colour is the first chart coordinate; the black paths are level sets of the second, and the axes are unnumbered for the same reason as in the previous figure.

The left-hand panel of Figure 6.2 is the picture practitioners recognise and reward: crisp, well-separated islands, obvious structure, something to point at. It scores 0.88. The right-hand panel is a featureless filled quadrilateral with no structure to point at whatsoever. It scores 0.089, a factor of 9.9 better, and it is featureless because **the answer is a flat rectangular patch**. The truth

here has no clusters in it. Every island in the left-hand panel is an artefact of the setting.

Q_{NX} against the chart moves the same way — 0.727 to 0.875 — so this is not one metric's idiosyncrasy. What disagrees with both is the eye. The tuning advice that circulates for these methods is largely advice about how the picture looks (Wattenberg et al., 2016; Kobak and Berens, 2019), and on this benchmark that advice points away from the truth, monotonically, across the whole swept range.

6.5 Results

```
cl <- grid[grid$noise == "none" & grid$status == "ok" &
           grid$method %in% c(NBR, "isomap"), ]
sp <- do.call(data.frame,
              aggregate(rmse ~ method + theta, cl,
                        function(x) c(lo = min(x), med =
                         ↪  median(x),
                                      hi = max(x))))
names(sp) <- sub("^rmse\\.", "", names(sp))

th_max <- max(THETA_GRID)
degrade <- vapply(c(NBR, "isomap"), function(m) {
  a <- sp[sp$method == m, ]
  a$med[a$theta == th_max] / a$med[a$theta == 0]
}, numeric(1))

widths <- transform(sp, w = hi - lo)
worst  <- widths[widths$method == "tsne", ]
worst  <- worst[which.max(worst$w), ]

out_hi <- aggregate(rmse ~ method,
                    grid[grid$noise == "outlier" & grid$status ==
 ↪  "ok" &
                         grid$theta == th_max, ], median)
out_hi <- setNames(out_hi$rmse, out_hi$method)

out_hi_mid <- aggregate(rmse ~ method,
                        grid[grid$noise == "outlier" &
 ↪  grid$status == "ok" &
                             grid$theta >= 0.5, ], median)
out_hi_mid <- setNames(out_hi_mid$rmse, out_hi_mid$method)

tsn  <- sp[sp$method == "tsne", ]; tsn <- tsn[order(tsn$theta), ]
ump  <- sp[sp$method == "umap", ]; ump <- ump[order(ump$theta), ]
gap  <- abs(tsn$med - ump$med)
over <- pmin(tsn$hi, ump$hi) > pmax(tsn$lo, ump$lo)

jump <- function(noise) {
  a <- aggregate(qnx ~ theta,
```

```
                grid[grid$method == "lle" & grid$noise == noise
↪  &
                     grid$status == "ok", ], median)
  max(diff(a$qnx[order(a$theta)]))
}
lle_jump <- c(none = jump("none"), outlier = jump("outlier"))
```

The prediction, stated before the figure because a flat curve otherwise reads as nothing happening: **these curves should be nearly flat.** Fold angle is a difficulty parameter for a method that consumes distance, because folding contracts ambient distance and leaves the chart alone (Chapter 2). It is a much weaker difficulty parameter for a method that consumes rank order, because within a facet the fold changes no distance at all and across a crease it changes ranks only where two facets pass close. A method that never trusted global structure has less to lose as the global structure degrades.

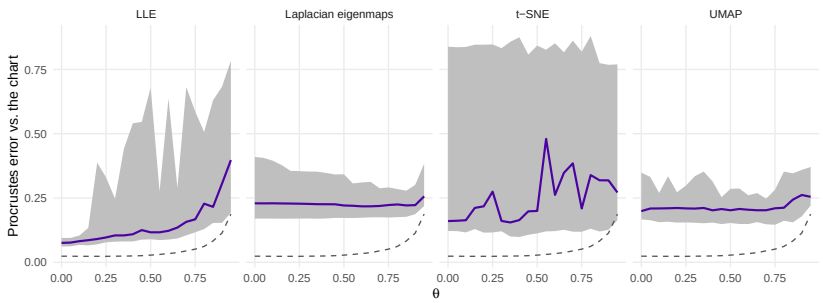


Figure 6.3: Procrustes error against the chart across the fold sweep, no noise. The band is the full spread across the twenty benchmark seeds and is the result; the solid line is its median, drawn to locate the band rather than to summarise it. The dashed grey line in each panel is Isomap's median, for scale.

Figure 6.3 is the prediction met. From flat sheet to $\theta = 0.95$, Isomap's median error grows by a factor of 7.9; Laplacian eigenmaps' by 1.1, UMAP's by 1.3, t-SNE's by 1.7. Only LLE behaves like a distance method, at 5.3, which is what one would expect of a method whose weights are least-squares reconstructions in the ambient space.

Flatness is not a compliment here. Isomap is better than all four at *every* θ in this regime, including the last one, and it gets there by consuming a geometry the crease pattern supplies exactly. A flat curve at a mediocre level says the error is dominated by something the fold does not touch — the objective mismatch itself — not that the method is robust.

Where the flatness does buy something is off the clean diagonal. Under the outlier noise model at $\theta = 0.95$, the ordering inverts: t-SNE lands at 0.402 and UMAP at 0.407, against 0.608 for Isomap, 0.626 for PCA and 0.888 for LLE. A handful of displaced points lengthens a shortest path through the neighbourhood graph and moves an ambient distance; it changes few ranks. The methods that were never reading distances are the ones that do not notice.

6.5.1 Neighbourhood size, and where Isomap breaks

```
sw <- sweeps[sweeps$status == "ok" & sweeps$method %in% c(NBR,
 ↪  "isomap"), ]
ks <- aggregate(rmse ~ method + k_tck + theta, sw, median)
k_max <- max(ks$k_tck)

best_k <- do.call(rbind, lapply(split(ks, list(ks$method,
 ↪  ks$theta)),
  function(a) data.frame(method = a$method[1], theta =
   ↪  a$theta[1],
                         k_star  = a$k_tck[which.min(a$rmse)],
                         penalty = a$rmse[a$k_tck == k_max] /
                          ↪  min(a$rmse))))
iso_k <- best_k[best_k$method == "isomap", ]
iso_k <- iso_k[order(iso_k$theta), ]
nbr_k <- best_k[best_k$method %in% NBR, ]
```

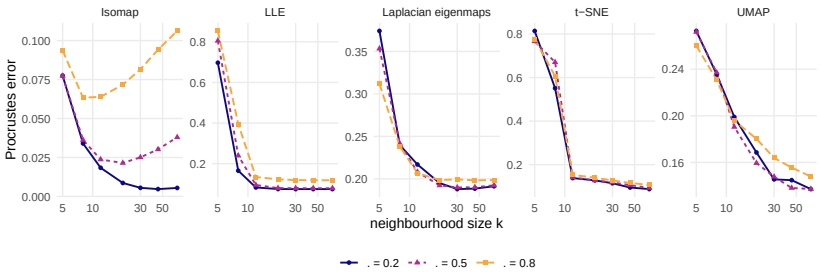


Figure 6.4: Procrustes error against neighbourhood size, at three fold angles. Each panel has its own vertical scale, because the levels differ by more than an order of magnitude and the shape of each curve is the point. Isomap has an interior optimum that moves to smaller neighbourhoods as the sheet folds; every neighbour method is still improving at the largest size swept.

Figure 6.4 is the sharpest thing in the chapter. Isomap's best neighbourhood size falls from 45 at $\theta = 0.2$, to 20 at 0.5, to 8 at 0.8. That is the short-circuit onset of Chapter 5 seen from the other side: as the sheet closes, the neighbourhood size at which the graph starts bridging between facets comes down, and past it Isomap's error rises. Choosing 70 — an unremarkable choice — costs it a factor of 1.68 over its own optimum at the tightest fold.

Not one of the four neighbour methods has an interior optimum anywhere in this sweep. Their worst penalty for taking the largest neighbourhood swept is 1.02, across all twelve method–angle combinations; for most of them it is exactly one, because the largest neighbourhood *is* the best. They cannot be short-circuited by a neighbourhood that is too large, because they never built a global object out of local ones. This is the genuine win, and it is narrow: it is robustness to a tuning mistake and to contamination, not accuracy.

6.5.2 Do they actually disagree?

The chapter's title asks why t-SNE and UMAP disagree about the same data. On this benchmark the honest answer is that mostly they do not, and what looks like disagreement is one draw from each of two wide distributions.

Their median errors differ by 0.048 across the sweep, against a t-SNE seed band whose median width is 0.72. The two bands overlap at 20 of the 20 fold angles in the grid — all of them. Two practitioners running one t-SNE and one UMAP on the same sample will get pictures that differ, and will usually be looking at within-method seed variation rather than at a difference between the methods. Where the two do separate is not in the median but in the spread: UMAP's band is 4.9 times narrower than t-SNE's, so a UMAP run is a much better estimate of what UMAP does than a t-SNE run is of what t-SNE does.

6.6 Diagnostics

6.6.1 Two conventions, and which claim needs which

Every claim above about recovery is made against the **chart** — the answer key — using either the Procrustes error or Q_{NX} at $K = 20$. The grid also carries trustworthiness, continuity and k-NN preservation, and those are scored at $k = 10$ against **ambient** distance, which is a different question: whether the embedding preserved the neighbourhoods a method could actually see. Both are defensible and they are not interchangeable. In `part2-sweeps.rds` the ambient-referenced trio is scored at the *swept* k, so its rows are not comparable with each other; Q_{NX} stays pinned at 20 against the chart and is. Every number in this chapter's figures is one of the two chart-referenced ones.

```
Kgrid <- unique(round(exp(seq(log(2), log(400), length.out =
 ↪  18L))))
ref   <- c(fit, list(pca    = embed("pca", cell),
                     isomap = embed("isomap", cell, k =
                      ↪  K_DEFAULT)))
rx <- do.call(rbind, lapply(names(ref), function(m)
  data.frame(method = m, K = Kgrid, rnx = rnx(ref[[m]], Dchart, K
   ↪  = Kgrid))))

margin <- min(vapply(Kgrid, function(kk) {
  a <- rx[rx$K == kk, ]
  a$rnx[a$method == "pca"] - max(a$rnx[a$method %in% NBR])
}, numeric(1)))
ts <- rx[rx$method == "tsne", ]
```

Figure 6.5 puts the global loss on one axis. $R_{NX}(K)$ is Q_{NX} corrected for the K/n baseline, so it is comparable across neighbourhood sizes (Lee and Verleysen, 2009); small K is local structure and large K is global. PCA — a shadow, and the weakest thing in Part II by construction — sits above all four neighbour methods at **every** K from 2 to 400, by at least 0.032. There is no neighbourhood size at which any of these four is the better description of the chart's rank order.

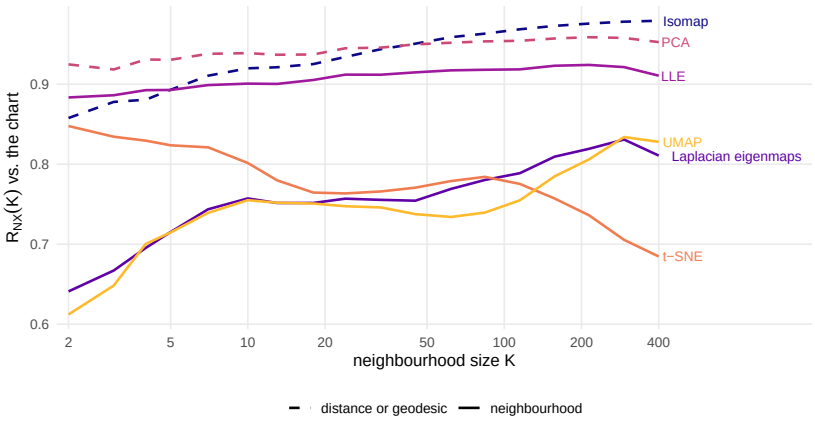


Figure 6.5: Rank preservation against the chart across every neighbourhood size, on one sample. The two methods that consume distance or geodesics are drawn dashed. No neighbour method reaches PCA at any neighbourhood size, and t-SNE is the only curve that falls as the neighbourhood grows.

Isomap is the exception to note rather than to bury: at the smallest neighbourhood it sits below LLE, at 0.858 against 0.883, and only overtakes it a few neighbours later. A geodesic method has no special claim on the very local rank order, which is exactly the part LLE is built out of.

t-SNE is the only curve that falls: 0.848 at $K = 2$, 0.685 at $K = 400$. It buys the local structure it is best at by spending the global structure, which is the trade its critics describe and its defenders accept (Chari and Pachter, 2023; Lause et al., 2024). Here the trade is priced exactly, because the thing being spent is known.

6.6.2 What twenty seeds is, and what it is not

```
bud <- budget$budget[budget$budget$method %in% NBR, ]
bud <- bud[order(-bud$seeds_needed), c("method", "stochastic",
 ↪  "sd",
                                       "seeds_needed")]
knitr::kable(bud, row.names = FALSE, digits = 4,
             col.names = c("method", "stochastic", "worst-cell
              ↪  sd",
                           "seeds needed"),
             caption = "Seeds required to resolve a reportable
              ↪  difference, from
              ↪  data/processed/seed-budget.rds.")
```

```
declined  <- grid[grid$method == "laplacian" & grid$status ==
 ↪  "declined", ]
dec       <- table(declined$noise)
dec_share <- nrow(declined) / sum(grid$method == "laplacian")
```

Table 6.2: Seeds required to resolve a reportable difference, from data/processed/seed-budget.rds.

method	stochastic	worst-cell sd	seeds needed
tsne	TRUE	0.2873	3241
lle	FALSE	0.2034	1624
laplacian	FALSE	0.0446	79
umap	TRUE	0.0438	76

Standing rule 1's floor is twenty seeds. `seed-budget.rds` measures what that floor is worth: for each method it draws 40 samples per cell at three fold angles and two noise models, takes the within-cell standard deviation of excess error over the floor, and inverts the two-sample expression for the seeds needed to resolve a difference of 0.02 at 80% power and the 5% level.

None of the four passes. t-SNE would need 3241 seeds, LLE 1624, Laplacian eigenmaps 79 and UMAP 76. The floor is between one and two orders of magnitude short for the first two. That is why the bands in Figure 6.3 are the result and no mean line is drawn anywhere in this chapter: at $\theta = 0.4$ t-SNE's twenty seeds span 0.098 to 0.875, a range of 0.777 on a metric bounded by one. A mean of that is a number with no cell in it.

Note which methods are on that list. LLE and Laplacian eigenmaps are deterministic: the same input gives the same output every time. What varies across seeds for them is the *sample*, redrawn from the same surface, and LLE's sensitivity to it is larger than either stochastic method's sensitivity to its own initialisation. "Seed variance" is two different quantities wearing one name, and only for t-SNE and UMAP does it mean what the phrase suggests.

6.7 Where it fails

6.7.1 `preserve.seed`, and a correction to the correction

```
small <- sample_manifold(pat, theta = 0.5, n = 200L, seed =
 ↪  BENCH_SEEDS[1])
runs  <- function(preserve) {
  set.seed(1L)
  vapply(BENCH_SEEDS, function(s)
    umap::umap(small$X, n_neighbors = K_DEFAULT, n_epochs = 60L,
               preserve.seed = preserve)$layout[1L, 1L],
                ↪  numeric(1))
}
has_arg  <- "preserve.seed" %in% names(formals(umap::umap))
default  <- formals(umap::umap)$preserve.seed
distinct <- c(kept = length(unique(runs(TRUE))),
              released = length(unique(runs(FALSE))))
```

The natural idiom for standing rule 1 is to seed once and loop. With `umap::umap()` that idiom silently collapses the whole loop into one run. The

pinned version takes a `preserve.seed` argument — present: TRUE, default: TRUE — and that default is the mechanism. Twenty calls with the argument at its default return 1 distinct answer; the same twenty with it switched off return 20. The function restores R's random state on exit, so consecutive calls all start from the same place, and the reported seed-to-seed spread of every UMAP result in this book would have been exactly zero — a violation of standing rule 1 that produces no warning, no error and numbers that look fine.

> The provenance of this finding needs stating, because the repository twice recorded it wrongly and in opposite directions. `PLAN.md` S1-6 predicted the defect and named `preserve.seed` as its cause. `R/methods.R` and `tests/testthat/test-methods.R` then recorded that the prediction had named the wrong mechanism, on the ground that the pinned version has no such argument, and asserted the behaviour instead. The prediction was right: the argument is present, it defaults to `TRUE`, and switching it off restores the variation — all three checked in the chunk above rather than asserted here. The test is still a good test, because it pins the behaviour rather than an explanation of it; its comment is wrong and the registry's is too.

The registry uses `uwot::umap()` instead, which consumes R's random stream, so a caller who forgets to seed gets visibly different answers rather than twenty identical ones. The choice of implementation moves the answer, and it moves it by changing what a seed means rather than by changing an algorithm.

6.7.2 Laplacian eigenmaps declines a whole noise model

`embed_laplacian()` returned nothing on 400 of the 1200 cells it was offered, which is 33% of them, and every one of those cells carries the outlier noise model: outlier and nothing else. The heat kernel underflows on a spread sample, rows of the weight matrix go to zero, the graph disconnects at working precision, and the second eigenvector is then read out of a degenerate space where every direction is equally valid. The helper detects that and refuses. A refusal is a result and the grid records it as one; what it is not is a score, and Chapter 10 must not average over the cells that are missing.

6.7.3 LLE under outliers stops being a curve

LLE never declines a cell, and that is worse than declining one. Under the outlier model its median error over the upper half of the fold sweep is 0.832, second only to the diffusion map, and its Q_{NX} against the chart stops behaving like a function of θ: between adjacent grid points it steps *upward* by as much as 0.18, against 0.003 on the clean sweep. Standing rule 2 asks for a curve; what this cell array produces is a scatter that happens to be indexed by θ. The reconstruction weights are least-squares fits in the ambient space, so one displaced neighbour rewrites a row of the weight matrix, and the whole embedding is a null space of that matrix. Nothing in the method notices.

6.7.4 What this chapter does not establish

One pattern family, one tessellation size, one ambient dimension, and 800 points. The neighbourhood sweep runs to 70, which is 9% of the sample — the curves are still falling there and this chapter cannot say where they stop. Nothing here measures whether these methods are good at the task they were built for, which is a question about cluster recovery that a folded sheet with no clusters cannot ask (Espadoto et al., 2021).

6.8 Reproduce this

- `scripts/run-benchmark-grid.R` → `data/processed/benchmark-grid.rds`, the θ sweep at fixed k.
- `scripts/run-part2-sweeps.R` → `data/processed/part2-sweeps.rds`, the k sweep at three fixed θ.
- `scripts/measure-seed-budget.R` → `data/processed/seed-budget.rds`.
- `tests/testthat/test-methods.R` asserts the unseeded refusal, the seed-to-seed variation of `Rtsne` and `uwot`, and the `umap::umap` behaviour above.

Each script's header declares the artefact it writes, and `scripts/check-artefact-producers.R` maps every artefact the book reads back to one.

6.9 Further reading

Roweis and Saul (Roweis and Saul, 2000) and Belkin and Niyogi (Belkin and Niyogi, 2003) are the primary sources for the two spectral methods, and both are usually read beside Isomap (Tenenbaum et al., 2000), which Chapter 5 covers and which is the one method here that consumes a geometry a crease pattern supplies exactly. Van der Maaten and Hinton (van der Maaten and Hinton, 2008) and McInnes, Healy and Melville (McInnes et al., 2018) are the primary sources for the other two.

On what the pictures are worth, Wattenberg, Viégas and Johnson (Wattenberg et al., 2016) is the clearest demonstration that the perplexity setting manufactures structure; Kobak and Berens (Kobak and Berens, 2019) is the practical counterpart. Kobak and Linderman (Kobak and Linderman, 2021) show that the global structure both methods are accused of losing is substantially a matter of initialisation rather than of the objective. Chari and Pachter (Chari and Pachter, 2023) press the case against reading these embeddings as maps and Lause, Berens and Kobak (Lause et al., 2024) answer it. For the rank-based criteria used here, Lee and Verleysen (Lee and Verleysen, 2009) is the reference, and Machado, Behrisch and Telea (Machado et al., 2025) make the point this chapter's two-metric disagreement illustrates: a projection quality metric is necessary and not sufficient.

Chapter 7

Learned folds

R. Heller · Date: 2026-07-29 · Last changed: 2026-11-09

Can a network learn the fold and the unfold at once?

Chapter status: planned. The items below describe work still to be implemented; this chapter does not provide a trained autoencoder or benchmark results.

- A `torch` autoencoder with a two-unit bottleneck. (`keras` would pull TensorFlow and break clean-clone reproducibility.)
- The only method here that learns an explicit inverse: encoder = low-dimensional representation, decoder = reconstruction in the input space.
- Reconstruction loss is the fold–unfold identity made computable.
- Compare the learned decoder against the true unfolding, not just against the input.

Part III

The benchmark

Chapter 8

Building crease-pattern manifolds

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

How do you build a manifold whose true geometry you already know?

8.1 What this chapter answers

A benchmark for dimension reduction has to supply four things. Three of them are routine. Data: points in an ambient space that a method can be handed. A difficulty axis: some parameter that makes the same object harder without making it a different object. An answer key: the coordinates the method should have recovered. The Swiss roll and its relatives supply all three, and so does every folded sheet in this book.

The fourth is the one that is almost never supplied, and it is what this chapter builds. **How much of the error was unavoidable?** A method that scores an error of one half on a dataset has either thrown away most of the structure or reached the limit of what two dimensions can hold, and the score alone does not say which. Without that number a benchmark can rank methods against each other but cannot say whether any of them did well, and a ranking with no scale under it is how a survey ends up reporting that every method is mediocre on every dataset (Espadoto et al., 2021).

The number exists and is computable in closed form, but computing it needs a manifold whose intrinsic dimension exceeds the target. That is this chapter's argument: the constructions in `R/constructions.R` exist to raise intrinsic dimension while keeping the answer key exact, because a chart of dimension two embedded into dimension two has an irreducible loss of exactly zero and nothing to say.

The chapter also records what the benchmark does *not* contain. Two pattern families were specified for it and both were withdrawn on evidence, and they

are reported here as results — what it takes for a pattern to qualify is worth more to a reader than a list of the ones that did.

8.2 Setup

```
grid <- read_run("data/processed/product-grid.rds",
 ↪  "bb8ade76d88b95a3")
fac_a <- miura_ori(4L, 4L, alpha = pi / 3)
fac_b <- miura_ori(3L, 5L, b = 1.4, alpha = pi / 4)
```

`grid` is the product suite, 12960 rows produced by `scripts/run-product-grid.R`; `fac_a` and `fac_b` are the two factors it is built from. Everything else this chapter calls is in `R/`, sourced when the book renders.

8.3 Background

8.3.1 Four objects, four verbs

The helper API is small on purpose. Three S3 classes carry everything between files, each a plain list, and `R/README.md` states the contract field by field.

A **`crease_pattern`** is a flat, unfolded pattern: vertices in the plane, facets as vertex-index cycles, and a crease table carrying a mountain/valley/boundary assignment. Its `vertices` matrix is the ground truth — not an approximation of the correct two-dimensional embedding, the embedding itself.

A **`folded_pattern`** is the same pattern placed in $\mathbb{R}^3$ at folding parameter θ, with one dihedral angle per crease derived from the placement rather than assumed.

A **`manifold_sample`** is n points drawn from a folded pattern: `X` in the ambient space, `truth` in the chart, row-aligned. `X[i, ]` and `truth[i, ]` are the same point seen two ways, and every measurement in this book compares an embedding of the first against the second. The verbs run one line each.

```
p  <- miura_ori(nx = 6L, ny = 6L, alpha = pi / 3)
s  <- sample_manifold(p, theta = 0.6, n = 400L, seed =
 ↪  BENCH_SEEDS[1])
Y  <- embed("isomap", s, d = EMBED_DIM, k = K_DEFAULT)
data.frame(vertices = nrow(p$vertices), creases =
 ↪  nrow(p$creases),
           points = nrow(s$X), ambient = ncol(s$X), chart =
            ↪  ncol(s$truth),
           error = round(reconstruction_error(Y, s$truth), 4))
```

```
#>   vertices creases points ambient chart  error
#> 1       49      84    400       3     2 0.0551
```

Four things about that call are load-bearing and easy to get wrong.

θ is a parameter of the folding map, not an angle. It runs on $[0, 1]$, zero being the flat sheet and one the flat-folded state, and `fold()` refuses anything

outside it; the dihedral angles are derived from it. Chapter 2 gives the derivation.

`embed()` takes the sample object, not a matrix. The registry holds 9 entries and 8 of them run: the autoencoder declares itself unavailable rather than failing at fit time, and `method_table()` prints the registry with what each method consumes.

Stochastic methods refuse to run unseeded. Standing rule 1 reports every stochastic result across `BENCH_SEEDS`, and an unseeded fit cannot be one of them, so `embed()` errors rather than quietly producing an anecdote.

Sampling trims the boundary by default. The chart is the geodesic only while the straight chart segment between two points stays on the paper, and a Miura unfolds to a region with a zigzag edge. `boundary = FALSE` is not tidiness — it is the condition under which the answer key is the answer.

8.4 Constructions that keep the answer key

8.4.1 One pattern family, and two honest gaps

The book ships one family that folds. That is fewer than were specified, and the two that were withdrawn were withdrawn for different reasons, both of which are worth more than the families would have been.

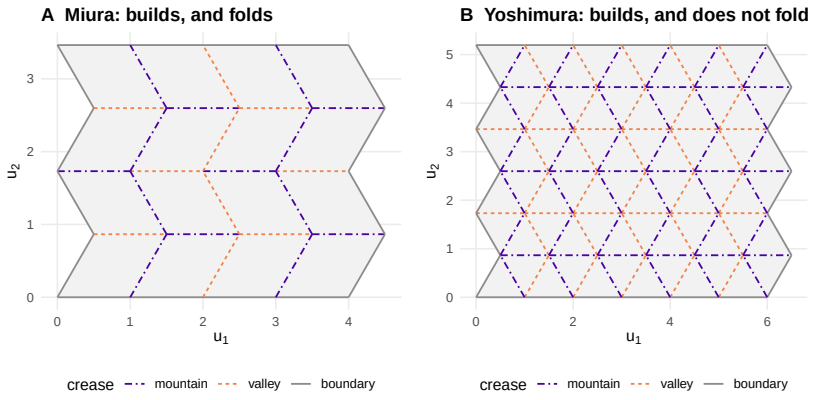


Figure 8.1: Two valid flat crease patterns. Only the left one has a certified rigid folding; the right one is drawn here and never folded anywhere in this book. Mountain and valley are distinguished by colour and by line style together.

Figure 8.1 shows the Miura beside the Yoshimura. Both panels are legitimate crease patterns. `yoshimura()` returns a `crease_pattern` like any other, its facets are congruent, and its interior vertices satisfy Kawasaki's condition — the alternating sum of the sector angles vanishes — to numerical noise. What it does not have is a folding.

```
yo <- yoshimura(6L, 6L)
kaw <- max(vapply(interior_vertices(yo), function(v) {
  a <- sector_angles(yo, v)
```

```
  abs(sum(a[c(TRUE, FALSE)]) - sum(a[c(FALSE, TRUE)]))
}, numeric(1)))
first_line <- function(e) sub("\\. See .*", ".",
 ↪  conditionMessage(e))
refusals <- c(
  yoshimura = tryCatch(fold(yo, theta = 0.5), error =
   ↪  first_line),
  waterbomb = tryCatch(waterbomb(), error = first_line))
cat(strwrap(refusals, width = 76, prefix = "  ", initial = ""),
 ↪  sep = "\n")
```

```
#> no rigid folding of the Yoshimura pattern is certified in which all three
#>   crease families fold. Two were derived and each leaves one family flat,
#>   which makes the folded object a parallelogram tessellation rather than a
#>   diamond one -- a Miura in this pattern's clothing.
#>   waterbomb() is not implemented: whether this tessellation admits a
#>   one-parameter rigid folding is an open question in this project, not a
#>   coding task.
```

Kawasaki's alternating sum is satisfied at all 25 interior vertices of that patch, worst departure 1.8e-15, and that buys nothing. Kawasaki is a necessary condition for flat-foldability at a vertex; it is not sufficient, and flat-foldability of a pattern is not the same question as the existence of a one-parameter rigid folding of a tessellation (Hull, 2020; Demaine and O'Rourke, 2007).

The Yoshimura. Two rigid foldings of the planar patch were derived, implemented and verified in this repository. Each of them is exact — facet isometry at the same order as the Miura's, flat at $\theta = 0$ — and each leaves one of the three crease families entirely unfolded, at a dihedral angle of π along its whole length. A folding that leaves a crease family flat is a folding of a coarser pattern: fuse the facets across the creases that never move, and what remains is a parallelogram tessellation. It is a Miura wearing the Yoshimura's crease lines. Shipping it as a second, independent family would have been a claim about pattern variety that the geometry does not support, so `fold()` refuses it and says why.

The waterbomb. `waterbomb()` stops before building anything. A degree-six vertex with sectors of forty-five, forty-five and ninety degrees repeated satisfies Kawasaki and is flat-foldable *as an isolated vertex*, which does not imply that the tessellation admits a one-parameter rigid folding. Whether it does is an open question in this project, not a coding task. A continuation moved it only under an imposed symmetry, and three things disqualified that result: the embedding is unproved, the fold amplitude is not monotone in θ, and degrees of freedom remain on the free boundary, so θ does not determine the configuration.

Both withdrawals share one moral and it generalises well past origami. **A test that a correct implementation passes is not the same as a test that a wrong one fails.** An isometry check cannot tell a fold from a pleat, because a pleat is a rigid folding and passes perfectly. A first-order flex at the flat state proves nothing either: at $\theta = 0$ every z column of the rigidity Jacobian vanishes, so the count of non-trivial infinitesimal flexes is $V - 3 = 46$ for a pattern of this

size and for *any* planar pattern whatsoever, foldable or not. Each trap here was sprung, and each was caught only by a second, independent check.

8.4.2 The isometry certificate

What a family has to produce, then, is a closed form, a fold that reaches every crease family, and an isometry that is exact rather than converged. The third is certified by measurement, not by the derivation that produced it.

```
alphas <- seq(MIURA_ALPHA_RANGE[1], MIURA_ALPHA_RANGE[2],
 ↪  length.out = 6L)
thetas <- seq(0, 1, by = 0.05)
cert <- do.call(rbind, lapply(alphas, function(al) {
  q <- miura_ori(4L, 4L, alpha = al)
  data.frame(alpha = round(al * 180 / pi), theta = thetas,
             err = vapply(thetas, function(t)
              ↪  facet_isometry_error(q, t),
                          numeric(1)))
}))
worst <- max(cert$err)
```

`facet_isometry_error()` compares **all pairwise distances** between a facet's vertices, folded against flat, and not merely its edges: a placement that preserved every edge of a parallelogram while shearing it would pass an edge test and would not be a rigid folding. Over the whole θ sweep and the full α range the worst departure is 1.1e-15, against a contract of 1e-10. That is floating-point noise on a closed form, not a tolerance anyone chose, and it is the certificate the product construction below inherits.

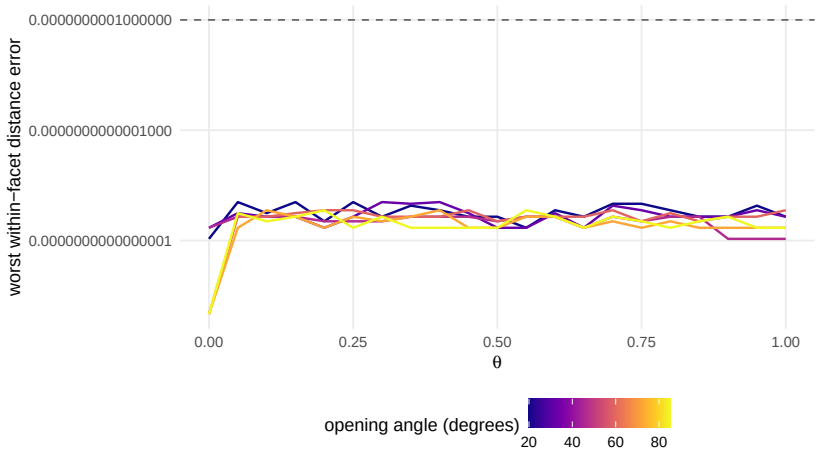


Figure 8.2: Worst within-facet distance error across the folding sweep, at six opening angles spanning the range the contract asserts. The dashed line is the contract; every curve sits five orders of magnitude below it and shows no trend in either the folding parameter or the opening angle.

Figure 8.2 is that measurement across the sweep. The flatness matters as much

as the level: an error growing with θ would mean the closed form was an approximation that happened to be good near the flat state, which is what a derivation by facet rotation tends to produce.

8.4.3 Products: raising intrinsic dimension without losing truth

A folded sheet has intrinsic dimension two and every embedding in this book has target dimension two. That is a benchmark with no headroom, and the product construction supplies some.

`product_manifold(a, b)` takes two samples of the same size and pairs them row-wise: the ambient coordinates concatenate to $\mathbb{R}^{3+3}$, the charts to $\mathbb{R}^{2+2}$. Nothing is fitted and nothing approximated, for a reason worth stating because everything downstream rests on it.

A product of isometries is an isometry. In a Riemannian product the squared distances add: for points (u, v) and (u', v'), $d^2 = d_1(u, u')^2 + d_2(v, v')^2$, and the same identity holds in the ambient product because concatenating coordinate blocks adds squared differences. Each factor map preserves its own term by construction, so the product map preserves the sum. The chart of the product is therefore the product of the charts, exactly, and its intrinsic dimension is the sum of the factors' — $2k$ in ambient $3k$ for k sheets.

```
sa <- sample_manifold(fac_a, theta = 0.55, n = 400L, seed =
 ↪  BENCH_SEEDS[1])
sb <- sample_manifold(fac_b, theta = 0.60, n = 400L, seed =
 ↪  BENCH_SEEDS[1] + 500L)
pr <- product_manifold(sa, sb)

pythag <- max(abs(as.matrix(dist(pr$truth)) -
                  sqrt(as.matrix(dist(sa$truth))^2 +
                   ↪  as.matrix(dist(sb$truth))^2)))
floors <- vapply(1:4, function(d) irreducible_loss(pr, d),
 ↪  numeric(1))
data.frame(ambient = ncol(pr$X), chart = ncol(pr$truth),
           pythagoras = formatC(pythag, format = "e", digits =
            ↪  1),
           one_sheet_floor = irreducible_loss(sa, 2L))
```

```
#>   ambient chart pythagoras one_sheet_floor
#> 1       6     4    8.9e-16               0
```

The Pythagorean identity holds to 8.9e-16 across all pairs, which is the numerical form of the argument above. The last entry is the sentence the whole construction exists to escape: a single folded sheet has a chart of dimension two, and the smallest error any two-dimensional embedding of it could achieve is 0. Exactly zero, in every cell, at every θ, under every noise model.

The bound. For an $n \times p$ chart U and the normalised Procrustes error of `R/metrics.R` — which optimises over translation, rotation, reflection and isotropic scale (Schönemann, 1966; Gower, 1975) — every $n \times d$ configuration Z

satisfies

$$\mathrm{err}(Z, U) \;\geq\; \mathrm{floor}(d) \;=\; \sqrt{\frac{\sum_{i>d} \lambda_i}{\sum_i \lambda_i}},$$

where the λ_i are the classical-MDS eigenvalues of the chart (Torgerson, 1952), equivalently the squared singular values of its centred coordinates. The proof is Eckart–Young (Eckart and Young, 1936) and it is short. As Z ranges over all configurations and the similarity group ranges over its elements, the fitted matrix ranges over exactly the centred $n \times p$ matrices of rank at most d; the best such matrix is the rank-d truncated singular value decomposition; its residual is the tail sum of squared singular values; normalising gives the expression above.

It is exact, it depends on the data alone, and no method appears anywhere in it.

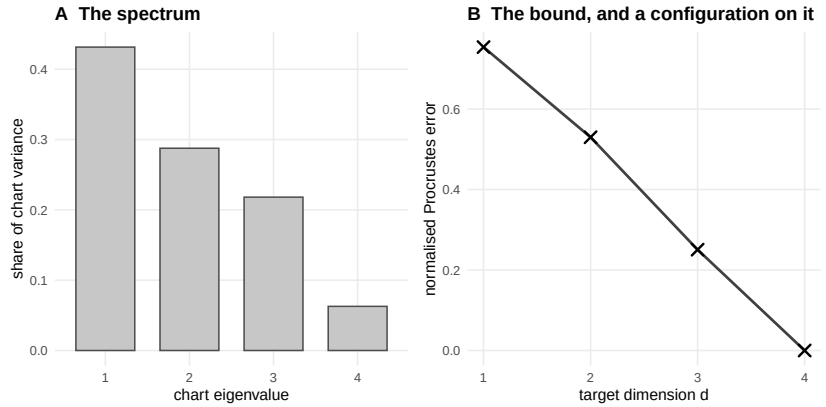


Figure 8.3: The chart spectrum of one product cell, and the bound it produces. Panel A: the four eigenvalues of the four-dimensional chart, as shares of its variance. Panel B: the bound at each target dimension, as a line, with the rank-d projection of the chart drawn on top of it as crosses. The tail of A is the square of B.

Figure 8.3 shows why it is a bound rather than a guess: **it is attained.** The crosses in panel B are the chart projected onto its own top d principal directions, scored by the same function every method in the book is scored by. They sit on the line, to 7.4e-16. A minimum, not an infimum — which also makes clear what the bound is not. The minimiser is constructed from the answer key, so nothing that has not been given the answer can be expected to reach it.

The floor depends on the chart and on nothing else. That is the property that makes it useful rather than a limitation: it does not move with θ, with the noise model or with the sample's difficulty, so "every method landed within a few per cent of the floor" is a statement about a regime rather than about a lucky draw.

8.4.4 Lifting to arbitrary ambient dimension

Real data does not arrive in three dimensions, and a method that only works when the ambient dimension is small should be caught by that rather than flattered by it. `lift()` maps a sample into $\mathbb{R}^D$ through a random orthonormal matrix.

```
lf <- lift(pr, D = 50L, seed = 20260901L)
dist_change <- max(abs(as.matrix(dist(lf$X)) -
 ↪  as.matrix(dist(pr$X))))
data.frame(ambient = ncol(lf$X),
           dist_change = formatC(dist_change, format = "e",
            ↪  digits = 1),
           floor_before = round(irreducible_loss(pr, 2L), 4),
           floor_after  = round(irreducible_loss(lf, 2L), 4))
```

```
#>   ambient dist_change floor_before floor_after
#> 1      50     2.2e-15         0.53        0.53
```

Orthonormality is load-bearing and `lift()` asserts it rather than assuming it: a merely random matrix would distort distances and invalidate the answer key silently. Because $Q^\top Q = I$ the map is exactly distance-preserving — the worst change over all 79800 pairs is 2.2e-15 — and the chart is untouched, so the floor is the same number before and after.

That decoupling is the point. Ambient dimension and intrinsic difficulty are two axes, and a benchmark that moves them together cannot say which a method is responding to. `lift()` moves the first alone.

8.5 Results

The product suite is 12960 rows: two arms, each swept over nine values of its own difficulty parameter, 20 seeds, four target dimensions and 9 registry entries at $n = 400$.

The second arm is a concession turned into a finding. `PROJECT_CONCEPT.md` listed crease-specificity among the book's differentiators, and `product_manifold()` imposes no family constraint: two isometric Swiss rolls give intrinsic dimension four, an exact chart and a computable floor exactly as two Miura sheets do. The differentiator as written is refuted, and reporting both arms side by side is the stronger statement. What crease-specificity rests on afterwards is zero reach (Federer, 1959; Niyogi et al., 2008) and an answer key that is not smooth, not the bound.

8.5.1 The floor bites, and nothing goes under it

```
fits <- nrow(ok)
under <- sum(ok$excess < -1e-9)
crease2 <- ok[ok$d == 2L & ok$family == "crease", ]
roll2   <- ok[ok$d == 2L & ok$family == "roll", ]
by_con <- aggregate(excess ~ consumes, crease2, mean)
```

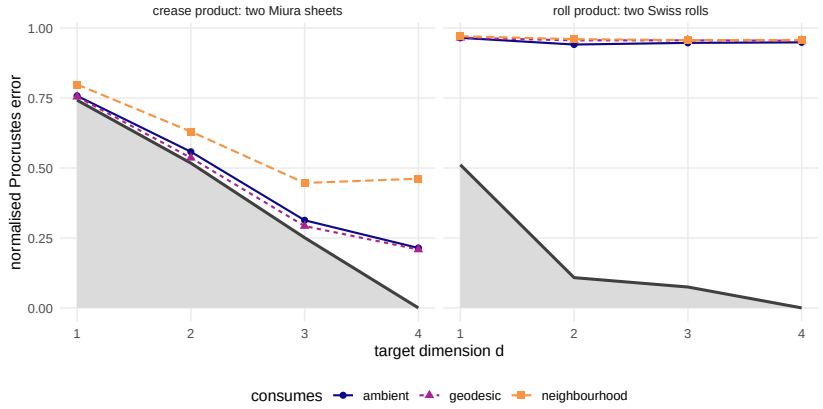


Figure 8.4: Achieved error against target dimension, by what each method consumes, with the irreducible-loss bound as the heavy line and the region below it shaded. Both arms have a four-dimensional chart, so the bound reaches zero at target dimension four. The shaded region is unreachable by any configuration whatsoever.

Across all 11160 fits in the suite, the number that went under the floor is 0. That is a regression check and not evidence, and the distinction matters: the bound is *proved*, so a method that beat it would mean the implementation is wrong, not that the claim is. It is the assertion the producer exits non-zero on, and it should be read the way a failing test is read, not the way a corroborated conjecture is.

Figure 8.4 is what the bound buys. At target dimension two the crease arm's floor is 0.517: the best configuration that exists still leaves a residual half the size of the chart's own spread, before any method is chosen. Isomap's raw error there averages 0.536, which against zero reads as failure and against the floor is an excess of 0.019 — it gives away 3.9 per cent of the range actually available to it. **That contrast is this book's thesis in one sentence.** Read the first way the table says everything failed; read the second it says the loss belongs to the data.

Excess sorts by what a method consumes. On the crease arm at target dimension two the mean is 0.019 for the one geodesic method, 0.041 for the ambient ones and 0.113 for the neighbourhood ones. Isomap consumes exactly what a crease pattern supplies exactly (Tenenbaum et al., 2000), and it is the method closest to the bound. That ordering is measured over the whole sweep and all 20 seeds. The provisional version of it came from two values of θ in a quick run that was never saved and put the ambient methods first — which is exactly the reading standing rule 2 exists to prevent.

8.5.2 Excess as a curve

Figure 8.5 is the whole sweep.

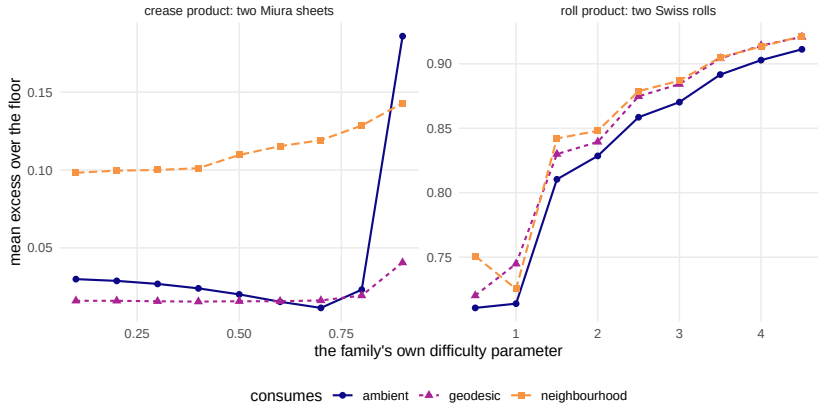


Figure 8.5: Mean excess over the floor at target dimension two, against each family's own difficulty parameter — the folding parameter on the left, the number of turns on the right. The vertical scales differ between panels and the two arms are matched on nothing, so this is a statement about the construction and not about which family is harder.

```
crease_curve <- aggregate(cbind(excess, floor) ~ par, crease2,
 ↪  mean)
roll_curve   <- aggregate(cbind(excess, floor) ~ par, roll2,
 ↪  mean)
```

Two things read off it. On the crease arm the floor is 0.517 at every one of the nine folding parameters — spread across the sweep exactly 0 — while the excess climbs from 0.062 to 0.146, almost all of it in the last two cells. The bound is a property of the answer key and folding does not touch it; the difficulty is entirely in the recovery. On the roll arm the floor *does* move, from 0.168 down to 0.068, because winding more turns lengthens the roll's arc-length coordinate and elongates its chart. Two arms, two different relationships between difficulty and the bound, and neither is a defect.

8.5.3 Why a benchmark stuck at intrinsic dimension 2 is not a serious benchmark

Everything above is unavailable on the book's main grid, and the reason is one line of arithmetic. That grid's chart is two-dimensional and its target is two-dimensional, so the `floor` column in it is identically zero — as `irreducible_loss(sa, 2L)` returned above, for the same reason, in closed form.

That is not a defect in the grid and it is worth one sentence rather than a footnote: **on the main grid every error is loss the method is responsible for.** It is the cleanest possible reading, and it is available precisely because the construction has no headroom in it.

But a benchmark that *only* ever has no headroom cannot ask the question this

chapter opened with. It cannot distinguish a method that discarded structure from one that ran out of room, because there is never any room to run out of. It cannot show what Figure 8.4 shows — the crease arm's methods hugging the bound at target dimensions one and two and pulling away from it at three and four, which says where each method's loss comes from rather than how large it is. And it cannot report a result as a ratio, because a ratio needs both ends fixed and one end is nailed to zero.

The classical benchmarks are all in this position. A Swiss roll, an S-curve, a severed sphere: two-dimensional charts, two-dimensional targets, floor zero everywhere (Lee and Verleysen, 2007). Raising intrinsic dimension is the one move that makes the bound bite, and doing it while keeping the answer key exact is what `product_manifold()` is for. The product suite, not the main grid, carries the argument.

8.6 Diagnostics

Do not read the two arms as a difficulty comparison. The crease arm is swept over the folding parameter and the roll arm over turns; the two are matched on nothing, and their floors differ by a factor of about 5 for reasons about how each chart is parameterised rather than about how hard each object is. What the pair supports is that the floor exists, is computable, is attained and is not beaten, in both families. A claim that one family is harder needs matched difficulty and does not live here.

Rows that did not run are results. Of the 12960 rows, 1440 are the autoencoder declaring itself unavailable and 360 are t-SNE refusing target dimension four — the Barnes–Hut implementation accepts one, two or three, and the `reason` column says so. Both are recorded rather than dropped, because a missing row and a refused row look identical in an aggregate and mean opposite things.

The bound has no opinion about rank metrics. `metric_floor()` bounds the normalised Procrustes error and nothing else. Trustworthiness, continuity and Q_{NX} read rank order rather than distance, and an embedding sitting on this floor may score better or worse on any of them than one well above it. That is Chapter 9's subject, not a defect here.

One stale message. `waterbomb()`'s error text still says the book ships two families plus one negative result; it predates the Yoshimura's withdrawal and was never revised. The count in this chapter is the correct one — one family, two negative results — and the discrepancy is left visible rather than quietly patched, because it is exactly the kind of stale count a registry keyed by family name is supposed to catch and did not.

8.7 Where it fails

A product is not a curved four-manifold. `product_manifold()` gives intrinsic dimension four with an exact chart, and it gives it in the easiest possible way: the factors are independent, their metrics do not interact, and the chart's

four coordinates come in two decoupled pairs. Nothing here models a manifold whose curvature entangles its directions. The construction is for the bound, and the bound needs only a chart of dimension greater than the target.

The lift is linear. What makes it a clean axis is also its limit. A random orthonormal lift adds no correlated nuisance structure, no heteroscedastic noise and no nearly redundant coordinates — none of the pathologies that make real high-dimensional data hard (Chari and Pachter, 2023; Lause et al., 2024). A method that survives `lift()` has been shown insensitive to ambient dimension alone.

The chart must be Euclidean. The bound is an Eckart–Young residual and needs the doubly-centred squared-distance matrix to be positive semi-definite. It is here, by construction. A distance matrix handed in from elsewhere may not be, and `chart_spectrum()` refuses that case rather than reporting a tail sum that bounds nothing.

The answer key is the chart only inside the paper. Chart distance equals geodesic distance only while the straight segment between two chart points stays on the sheet, and a Miura's outline is not convex. The producer asserts a chart-exit fraction of zero on every cell rather than trusting the sampling default; where that assertion fails the answer key is wrong in the direction that flatters every method.

One family. A benchmark built on a single pattern family cannot support claims about pattern variety, and this book makes none. It can support claims about a construction, and the suite's two arms are the evidence that the construction is what the claims are about.

8.8 Reproduce this

To add a family, write the constructor in `R/patterns.R` and the folding map in `R/folding.R`, and let `tests/testthat/test-folding.R` decide whether it qualifies: within-facet pairwise distances preserved to 1e-10 across the sweep, flatness at $\theta = 0$, an independent identity that was not built into the derivation, and a mountain/valley assignment derived from the folded geometry rather than asserted. A family that leaves a crease family flat fails the last of those, which is how the Yoshimura was caught.

To build a product of your own, hand `product_manifold()` two `manifold_sample` objects of equal size — they need not come from the same family, or from a crease pattern at all — and score against `irreducible_loss()` rather than against zero. `against_floor()` returns error, floor, excess and ratio together, which is the form every result in Part III is reported in.

The suite itself is `scripts/run-product-grid.R`, which is run neither in CI nor at render time: run it locally, commit the `.rds`, and let `read_run()` refuse any chapter whose prose was written against a different one.

8.9 Further reading

Miura's report (Miura, 1985) introduces the one family this benchmark folds, and Schenk and Guest (Schenk and Guest, 2013) give the mechanics of folded shells. Hull (Hull, 2020) is the reference for what Kawasaki's and Maekawa's conditions do and do not guarantee, Demaine and O'Rourke (Demaine and O'Rourke, 2007) for rigid foldability as a computational question, and Lang (Lang, 2011) for the design side this chapter does not need.

On the measurement side, Eckart and Young (Eckart and Young, 1936) is the approximation theorem the bound instantiates, Schönemann (Schönemann, 1966) and Gower (Gower, 1975) fix the alignment group it is stated over, and Torgerson (Torgerson, 1952) is where the chart spectrum comes from. Balasubramanian and Schwartz (Balasubramanian and Schwartz, 2002) state the failure mode the folding parameter drives, and Lee and Verleysen (Lee and Verleysen, 2007) is the textbook treatment of the methods being measured. Espadoto and colleagues (Espadoto et al., 2021) survey the benchmarking practice this chapter argues with, and Machado and colleagues (Machado et al., 2025) make the case for evaluating the evaluators, which Chapter 9 takes up.

Chapter 9

Ground truth and the evaluators

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

Your embedding looks good. Is it correct?

9.1 What "correct" means here

Machado, Behrisch and Telea built projections that score well on trustworthiness and continuity while displaying structure the data does not have (Machado et al., 2025). That is the false-*positive* direction, and it needs no ground truth: to show that a bad embedding scores high, it is enough to know that the embedding is bad. This chapter runs the other direction. It hands each evaluator the exact answer as a candidate embedding and asks whether the evaluator recognises it. That cannot be done without an answer key, which is why it is not usually done, and it is the one thing a crease pattern buys that no real dataset can supply.

The complement matters because the two failures have different consequences. A metric that rewards a wrong embedding wastes a reader's attention. A metric that punishes the right one changes which method gets shipped.

"Correct" is fixed here before anything is measured. An embedding is correct when it *is* the chart, up to a similarity of the plane. Chapter 2 established that the flat pattern is not an approximation to the folded sheet's intrinsic structure but that structure itself, so there is an answer and it is exact. Everything in this chapter is a comparison against that answer, and the object under test is not an embedding method. It is the scoring function.

The disagreement about two-dimensional embeddings of single-cell data has largely migrated from the embeddings to the evaluators. That framing is not this book's: Lause, Berens and Kobak state it as their published position, concluding that the case against those embeddings (Chari and Pachter, 2023) rested on metrics that "focused on preservation of distances, where 2D PCA was unsurprisingly the best" (Lause et al., 2024). This book takes no position on what

an embedding is for. It measures how far each evaluator sits from the evaluator that has the answer key.

Three things are established below, and one that the plan of record expected is not. The evaluators disagree with the answer key, systematically and in a direction that depends on which reference geometry they are given. Continuity in particular cannot tell a perfect embedding from a plainly wrong one at any fold angle. And the four metrics that a paper reports side by side are not four pieces of evidence. What does *not* survive is the shape of the law: the inversion threshold is not a property of the fold angle alone, and saying so is the point of having run the experiment rather than described it.

9.2 Setup

```
audit <- read_run("data/processed/evaluator-audit.rds",
 ↪  "d7982cf7eb3afe7d")
pilot <- read_run("data/processed/evaluator-audit-pilot.rds",
 ↪  "736906d41b87b8b3")
grid  <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")

prov <- attr(audit, "provenance")

# knitr rounds an inline numeric to options(digits), which turns
 ↪  4e-16 into 0
# and 0.0104 into 0.01. These return character, so a small number
 ↪  survives the
# trip into prose intact.
fmt <- function(x, d = 3L) formatC(signif(x, d), format = "fg",
 ↪  digits = d)
sci <- function(x, d = 2L) formatC(x, format = "e", digits = d -
 ↪  1L)

p    <- miura_ori(nx = 6L, ny = 6L)
m    <- sample_manifold(p, theta = 0.8, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
U    <- m$truth
pca2 <- stats::prcomp(m$X, rank. = EMBED_DIM)$x[,
 ↪  seq_len(EMBED_DIM)]
```

`audit` is the chapter's evidence: 19200 scored rows over 20 values of θ, 4 values of k, 2 sample sizes, 20 seeds, 3 reference geometries and 2 candidate embeddings, produced by `scripts/run-evaluator-audit.R` in 2.1 hours. `pilot` is the narrower design that decided this chapter existed at all, and `grid` is the main benchmark, read here for the metrics it already reports.

9.3 Background

9.3.1 The headline metric: normalised Procrustes RMSE

There is exactly one alignment convention in this book, and it lives in `procrustes_align()`. It is the **full** Procrustes fit (Schönemann, 1966; Gower, 1975): the similarity group of translation, rotation, **reflection** and one isotropic scale, and nothing else.

Each admission is a decision about what "the same chart" means.

Translation and rotation are free because a chart has no origin and no preferred axis. *Isotropic scale* is free because two of the methods under test have no scale to speak of — a t-SNE embedding's extent is a function of its perplexity and its iteration count (van der Maaten and Hinton, 2008) — and scoring them on one would measure a setting rather than a geometry.

Reflection is the one that must be argued. A mirrored unfolding is a correct unfolding. Turn the flat sheet over and the crease pattern is the same pattern, its mountains and valleys exchanged; no method in Part II is given anything from which it could recover a handedness, and none claims to. The implementation detail carries the whole argument: the rotation is taken as UV' from the singular value decomposition of the cross-product matrix, with **no determinant correction**. A version that forced $\det R = 1$ would score an exactly mirrored recovery as a failure.

```
rot    <- function(a) matrix(c(cos(a), sin(a), -sin(a), cos(a)),
 ↪  2L, 2L)
moved  <- 2.7 * U %*% rot(1.1) + matrix(c(4, -9), nrow(U), 2L,
 ↪  byrow = TRUE)
mirror <- moved %*% matrix(c(-1, 0, 0, 1), 2L, 2L)

err_moved  <- reconstruction_error(moved,  U)
err_mirror <- reconstruction_error(mirror, U)
det_moved  <- det(attr(procrustes_align(moved,  U), "rotation"))
det_mirror <- det(attr(procrustes_align(mirror, U), "rotation"))

zb       <- complex(real = U[, 1L], imaginary = U[, 2L])^2
bend     <- cbind(Re(zb), Im(zb))
err_exact <- reconstruction_error(U, U)
err_bend  <- reconstruction_error(bend, U)
qnx_bend <- qnx(bend, reference_dist(m, "chart"), K = K_QNX)
```

A rotated, rescaled, translated copy of the chart scores 4.7e-16. Its mirror image scores 4.7e-16 — the same number to the last digit that means anything — and the fitted rotation has determinant -1 against 1 for the unmirrored copy. Both are zero to floating point, which is what "quotients out" has to mean.

What is *not* quotiented out is everything else, and the boundary is sharp. Anisotropic scale and shear stay in the residual, because those are precisely the distortions the isometry claim is about. So does a conformal bend: the map $z \mapsto z^2$ is a rotation and an isotropic scale at every point, so it moves no neighbour out of any neighbourhood, while its scale factor varies across the chart.

It scores 0.255 here, and Q_{NX} at $K = 20$ against the chart calls the same configuration 0.963.

The score is normalised by the chart's own root-sum-square radius, which fixes both ends. Zero is exact recovery. One is an embedding that explains none of the chart's variance, and a random configuration reaches it to $O(1/n)$. A bounded metric matters because Chapter 10 reports every result as an excess over a floor, and an excess needs both ends nailed down.

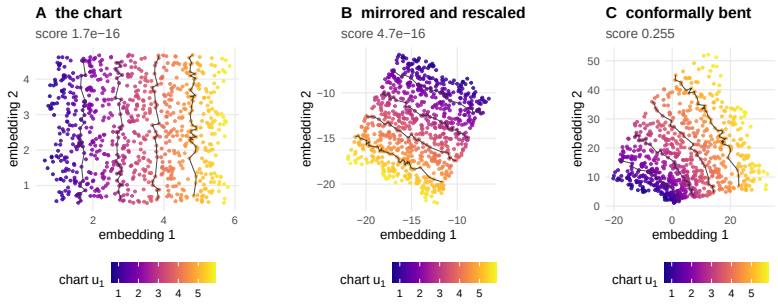


Figure 9.1: What the alignment quotients out, and what it does not. All three panels are scored against the same chart. Point colour is the first chart coordinate and the grey paths are its level sets, drawn so the structure survives greyscale printing.

Figure 9.1 is the convention in one row. Panel B is the same chart and scores as such. Panel C is not, and the rank metrics will disagree.

9.3.2 The four metrics under audit

Four scores are computed for every cell of the book's grid, and each is defined here against a **reference geometry** D and a neighbourhood size k. Write r_{ij} for the rank of point j in point i's ordering under D, and $\hat{r}_{ij}$ for the same rank in the embedding.

Trustworthiness and **continuity** (Venna and Kaski, 2001; Kaski et al., 2003) charge the embedding for the neighbours it invents and the neighbours it loses:

$$M_1(k) = 1 - A(k) \sum_i \sum_{j \in U_k(i)} (r_{ij} - k), \qquad M_2(k) = 1 - A(k) \sum_i \sum_{j \in V_k(i)} (\hat{r}_{ij} - k),$$

with $U_k(i)$ the points in the display's k-neighbourhood but not the reference's, $V_k(i)$ the reverse, and $A(k) = 2/(Nk(2N - 3k - 1))$ the scaling transcribed from Kaski et al. Two things about $A(k)$ travel with it and most re-implementations drop them. It is the reciprocal of the largest penalty sum any embedding can produce, which is why $M_1 = 0$ is a floor rather than merely a small number — and the authors are explicit that it is a *scaling*, not a proven bound on what a real projection can attain, so a value slightly below zero is not necessarily a bug and nothing here clamps one. Read as an information-retrieval pair they are precision and recall of the neighbourhood (Venna et al., 2010).

$Q_{NX}(K)$ is read off the co-ranking matrix of Lee and Verleysen (Lee and Verleysen, 2009), $Q[k, l]$ counting the ordered pairs at rank k in the reference and l in the embedding. Its top-left block is the fraction of K-neighbourhoods preserved, $Q_{NX}(K) = \frac{1}{Kn} \sum_{k \le K} \sum_{l \le K} Q[k, l]$, which is $K/(n-1)$ for a random embedding rather than zero; `rnx()` applies that correction and `auc_qnx()` integrates it over K on a log scale.

kNN preservation is the mean agreement of the two k-nearest-neighbour sets. One identity is worth stating rather than leaving to be rediscovered: its overlap form is *exactly* $Q_{NX}(k)$. `knn_preservation()` therefore defaults to the Jaccard form, so that the book's four metrics are four and not three.

Kruskal stress (Kruskal, 1964) is the fifth member of the usual set and is absent here on purpose. Chapter 2 proves as a one-line proposition that ambient-referenced stress must rank the exact unfolding below a two-component PCA of the folded cloud, for every $\theta > 0$; a simulation of a theorem is not evidence. What is not entailed, and what this chapter measures, is what the *rank* metrics do, because they are invariant to a monotone contraction of distances until neighbour identity changes.

9.3.3 Three reference geometries

Every one of the four takes a distance matrix and does not care where it came from. That is the lever.

- **ambient** — Euclidean distance in $\mathbb{R}^3$ between the points as observed. This is what PCA, MDS, t-SNE and UMAP actually consume, and what essentially every published evaluation uses (Espadoto et al., 2021).
- **chart** — Euclidean distance in the exact two-dimensional truth. On this construction that *is* the geodesic distance on the folded surface, exactly and not as an estimate.
- **graph** — the estimate a method has to work with: a symmetric k-nearest-neighbour graph on the ambient points with all-pairs shortest paths, which is Isomap's internal geometry (Tenenbaum et al., 2000).

Nothing in `R/metrics.R` prefers one, and the argument is named `reference` rather than `data` to keep that visible at the call site.

The book's own grid uses two of them at once, and the fact is worth stating plainly because it is invisible in the output.

```
D_amb <- reference_dist(m, "ambient")
D_cha <- reference_dist(m, "chart")

conventions <- data.frame(
  candidate  = c("the exact chart", "two-component PCA"),
  rmse       = c(reconstruction_error(U, U),
   ↪  reconstruction_error(pca2, U)),
  qnx_chart  = c(qnx(U, D_cha, K_QNX), qnx(pca2, D_cha, K_QNX)),
  knn_ambient = c(knn_preservation(D_amb, U,    K_DEFAULT),
                  knn_preservation(D_amb, pca2, K_DEFAULT)))
```

`grid` scores Q_{NX} at $K = 20$ against the chart, and trustworthiness, continu-

ity and kNN preservation at $k = 10$ against ambient distance. Two neighbourhood sizes and two reference geometries, in adjacent columns of one table. Both choices are defensible and `R/constants.R` records why: the first three ask whether the embedding preserved the neighbourhoods a method could *see*, so ambient is the honest reference; Q_{NX} is asked against the answer key, because that is the question a benchmark with an answer key exists to be able to ask.

The cost of leaving that unsaid is in the table. On the exact chart at $\theta = 0.8$, the Q_{NX} column reads 1 and the kNN column reads 0.789. One says the answer key is perfect. The other says it lost a fifth of its neighbourhoods. Both are correct, and they are not answering the same question.

9.4 The known-answer test

The design is one sentence. Hand each evaluator the exact chart as though it were a submission, alongside a two-component PCA of the folded cloud, and record which one it prefers.

PCA is the sharp choice of wrong answer rather than a straw man. It is the wrong chart — it flattens the corrugation instead of unfolding it — and it is simultaneously the least-squares-optimal *linear* reproduction of the ambient configuration it was given (Pearson, 1901). An evaluator that scores reproduction of ambient distance is being handed the best available competitor at exactly the thing it measures.

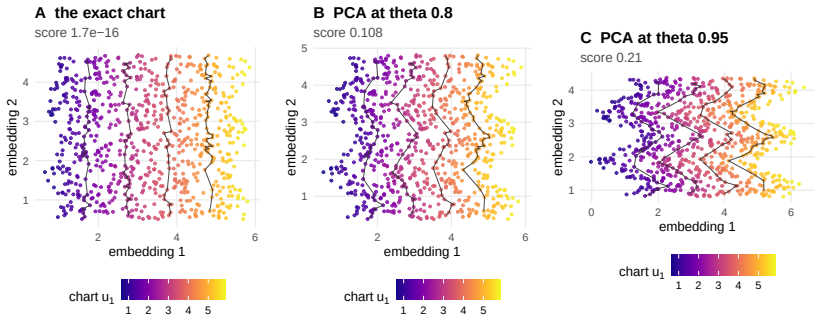


Figure 9.2: The candidates, Procrustes-aligned to the same answer key and scored against it. Point colour is the first chart coordinate and the grey paths are its level sets, which the exact chart draws as straight nested lines and a wrong chart does not. The wrongness is a function of the fold angle, which is why the audit sweeps it rather than fixing it.

Figure 9.2 is the comparison the audit runs at every cell. The exact chart scores 1.7e-16. PCA scores 0.108 at the fold angle used throughout this chapter and 0.21 at the last angle of the sweep, and the level sets it draws are kinked where the creases ran. An **inversion** is an evaluator ranking panel A below panel B or C.

Panel B is worth pausing on, because it is the reason this is a hard test rather than an easy one. At a moderate fold a Miura is still close to planar, so its

principal-component projection is not a grotesque failure — it is a mildly wrong answer that looks reasonable. An evaluator that cannot separate panel A from panel B is failing at a discrimination that matters, not at a trick.

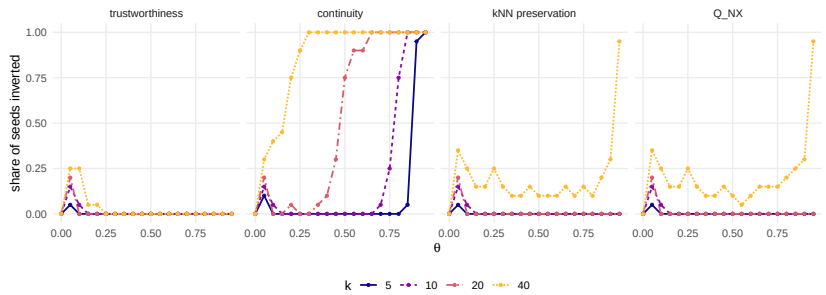


Figure 9.3: Inversion rate against the folding parameter: the share of seeds in which the evaluator ranks the exact chart below a two-component PCA of the same cloud. Ambient reference, n = 800. Neighbourhood size is encoded by colour and line style together. The panels do not pool over sample size, because the two sample sizes are two different phenomena.

Figure 9.3 is the result at the larger sample size, and it is four different pictures rather than one. Continuity inverts on a clean sigmoid whose location slides left as k grows. Trustworthiness, kNN preservation and Q_{NX} mostly do not invert at all: they rank the exact chart first almost everywhere, with a bump near $\theta = 0.05$ where the two candidates are barely distinguishable, and a low ragged band at the largest k that only becomes a majority in the last cell of the sweep.

```
inv <- audit$inversions
flat_inversions  <- sum(inv$inverted[inv$theta == 0])
chart_inversions <- sum(inv$inverted[inv$reference == "chart"])
chart_rows       <- sum(inv$reference == "chart")

sc <- audit$scores
chart_truth <- sc[sc$reference == "chart" & sc$candidate ==
 ↪  "truth", ]
truth_exact <- all(chart_truth$trust == 1 & chart_truth$cont == 1
 ↪  &
                     chart_truth$knn == 1 & chart_truth$qnx == 1)
```

Two controls make this a mechanism rather than a rate, and both come out clean.

At $\theta = 0$ the inversion count over the whole artefact is 0. A flat sheet contracts nothing, ambient distance *is* chart distance, and there is nothing to invert — which is the prediction Chapter 2's contraction result makes, evaluated at the one point where the contraction vanishes.

Against the chart reference the count is 0 in 12800 rows. Not rare: never. And the reason is arithmetic rather than luck — handed the chart as its reference, every metric scores the chart as a candidate at exactly one, which is checked

here rather than asserted (`truth_exact` is TRUE). No candidate can beat a perfect score, so no inversion is available.

The inversion is therefore not a property of the metric. It is a property of the pair (metric, reference geometry), and the reference is doing most of the work.

An inversion is also not a *bug*. It is the evaluator answering the question it was actually asked. An ambient-referenced metric measures reproduction of ambient distance; folding contracts ambient distance strictly; so the exact unfolding reproduces ambient distance badly by construction, and something that was fitted to reproduce it does better. That much is Chapter 2's stress proposition and needed no experiment.

What did need one is that a *rank* metric inverts at all. Rank metrics are invariant to any monotone transformation of distances, and the argument that they should therefore be safe here is the argument this chapter was budgeted to test. It fails on a detail: folding is not a monotone map on ambient distance. The contraction factor is $\sin(\rho/2)$ *per crease crossed*, so a pair straddling a fold contracts while a pair of the same chart distance inside one facet does not, and their order swaps. Neighbour identity therefore changes, and the amount it changes is measurable.

```
rk <- function(D) { diag(D) <- -Inf
                    t(apply(D, 1L, rank, ties.method = "first"))
                     ↪  - 1L }
agree <- function(r1, r2, K) mean(rowSums(r1 >= 1L & r1 <= K & r2
 ↪  >= 1L & r2 <= K)) / K

m_flat  <- sample_manifold(p, theta = 0, n = N_DEFAULT, seed =
 ↪  BENCH_SEEDS[1])
ra_flat <- rk(reference_dist(m_flat, "ambient"))
rc_flat <- rk(reference_dist(m_flat, "chart"))
ra <- rk(D_amb); rc <- rk(D_cha)

moved_flat <- 1 - agree(ra_flat, rc_flat, K_DEFAULT)
moved_fold <- 1 - agree(ra, rc, K_DEFAULT)
```

At $\theta = 0$ a point's ambient neighbours and its chart neighbours are the same set: the share that differ is 0. At $\theta = 0.8$ it is 0.13. That number is the whole mechanism. An eighth of every neighbourhood has been rewritten by the fold, the exact chart is scored on reproducing the rewritten version, and it does not reproduce it — because it is right.

9.4.1 Why the obvious fix fails

The obvious repair is to keep the metric and replace the reference: score against estimated geodesics instead of ambient distance, which is what a practitioner who has read Isomap would reach for. It does nothing, and it cannot.

```
D_gra <- reference_dist(m, "graph", k = max(prov$k))
rg    <- rk(D_gra)

Ks          <- prov$k
```

```
agree_graph <- vapply(Ks, function(K) agree(ra, rg, K),
 ↪  numeric(1))
agree_chart <- vapply(Ks, function(K) agree(ra, rc, K),
 ↪  numeric(1))
graph_above <- all(D_gra >= D_amb - TOL$default)

key      <- c("k", "theta", "n", "seed", "candidate")
paired   <- merge(sc[sc$reference == "ambient", ],
 ↪  sc[sc$reference == "graph", ],
                  by = key, suffixes = c(".a", ".g"))
gap_graph <- vapply(names(METRIC_LABEL), function(v)
  max(abs(paired[[paste0(v, ".a")]] - paired[[paste0(v,
   ↪  ".g")]])), numeric(1))
```

The graph reference agrees with the ambient reference on 1, 1, 1, 1 of the neighbour sets at k of 5, 10, 20, 40 — that is, on all of them. The chart reference, over the same points, agrees on only 0.908, 0.873, 0.865, 0.879.

The identity is exact and has a one-line proof. Any path through a Euclidean graph is at least as long as the straight line between its endpoints, so graph distance is never below ambient distance (`graph_above` is TRUE); and the graph contains an edge to each of a point's k nearest ambient neighbours, whose shortest path is therefore that edge. So for every $K \leq k$ the two geometries have the *same* K nearest neighbours in the *same* order. A rank metric evaluated at $K \leq k$ cannot see the difference.

The audit says the same thing across all 19200 scored rows: the largest gap between an ambient-referenced and a graph-referenced score is 0 for continuity, kNN preservation and Q_{NX}, and 0.0022 for trustworthiness. Trustworthiness is the only one that moves, and that too follows: it is the one metric that asks for the reference rank of a point the *embedding* placed nearby, which can lie far outside the graph's own k.

An estimated geodesic is not a different question from an ambient distance at the scale a rank metric looks at. Only an exact chart is, and only a benchmark with an answer key has one. This is a stronger statement than "graph distances are fragile" — the short-circuit failures of Chapter 5 (Balasubramanian and Schwartz, 2002) are about the long-range structure, and the point here is that even a *perfectly* connected graph changes nothing locally.

9.5 Results

9.5.1 How optimistic, and in what units

"Optimistic by X" is not a statement the evidence supports, and the reason is visible before any inversion is counted.

Figure 9.4 puts all five quantities on one axis, which they share because the normalised Procrustes score also lives on the unit interval. Trustworthiness and continuity are two flat lines at the top of the panel. They do not separate a

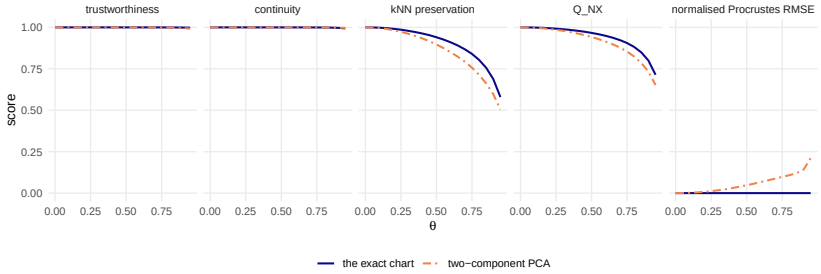


Figure 9.4: The two candidates scored against ambient distance across the whole fold sweep, at the book's default neighbourhood size and n = 800. All five quantities lie on the unit interval and share one axis, so the comparison is not an artefact of scaling. Candidate is encoded by colour and line style together.

configuration whose error is zero from one whose error reaches a fifth of the chart's radius, at any fold angle in the sweep.

```
amb <- inv[inv$reference == "ambient", ]
amb$margin <- amb$pca2 - amb$truth      # positive when the
 ↪  evaluator prefers PCA
big <- max(prov$n)

optimism <- do.call(rbind, lapply(names(METRIC_LABEL),
 ↪  function(mt) {
  d <- amb[amb$metric == mt & amb$n == big, ]
  data.frame(metric         = METRIC_LABEL[[mt]],
             inverted       = mean(d$inverted),
             widest_pca     = max(d$margin),
             widest_truth   = -min(d$margin),
             pca_reportable = mean(d$margin  > MIN_REPORTABLE),
             truth_reportable = mean(-d$margin > MIN_REPORTABLE))
}))

cont_cells      <- amb[amb$metric == "cont" & amb$n == big, ]
cont_reportable <- sum(cont_cells$margin > MIN_REPORTABLE)

one_cell <- function(ref, cand) {
  d <- sc[sc$reference == ref & sc$candidate == cand &
            sc$k == K_DEFAULT & sc$n == big, ]
  aggregate(cbind(trust, cont, knn, qnx) ~ theta, data = d, FUN
   ↪  = mean)
}
ceiling   <- one_cell("ambient", "truth")   # what a perfect
 ↪  answer scores
chart_pca <- one_cell("chart",   "pca2")    # what a wrong answer
 ↪  scores

knitr::kable(
```

Table 9.1: Four metrics, four answers. Ambient reference, at the larger sample size, over every fold angle, neighbourhood size and seed in the audit. 'Reportably' means by more than the book's pre-registered reportable difference.

metric	inverted	widest PCA lead	widest truth lead	PCA wins reportably	truth wins report
trustworthiness	0.013	0.000	0.009	0.000	0
continuity	0.418	0.010	0.000	0.000	0
kNN preservation	0.054	0.029	0.197	0.003	0
Q_NX	0.056	0.023	0.157	0.001	0

```
optimism, digits = 3, row.names = FALSE,
col.names = c("metric", "inverted", "widest PCA lead", "widest
 ↪  truth lead",
               "PCA wins reportably", "truth wins reportably"),
caption = "Four metrics, four answers. Ambient reference, at
 ↪  the larger sample size, over every fold angle,
 ↪  neighbourhood size and seed in the audit. 'Reportably'
 ↪  means by more than the book's pre-registered reportable
 ↪  difference.")
```

Table 9.1 is the honest summary and it does not reduce to a number.

Continuity inverts on 41.8 per cent of cells, and its widest lead over the truth anywhere at this sample size is 0.0104. It clears the book's pre-registered reportable difference of 0.02 in 0 of 1600 cells. So the claim the numbers support is *not* "continuity prefers a wrong embedding by X". It is that **continuity cannot distinguish a perfect embedding from a plainly wrong one at all**, and reports both as excellent. The weaker version sounds bolder and is not what happened.

Trustworthiness is inert too, and more nearly symmetrically so: it inverts on 1.3 per cent of cells and never moves by more than 0.0087 either way.

kNN preservation and Q_{NX} do the job asked of them. They rank the exact chart first, by a margin clearing the reportable difference in 45.9 and 32.9 per cent of cells, and their widest lead for the truth is 0.197. Ranking the right answer first is the whole of what is asked, and two of four metrics do it.

They do it while calling the right answer mediocre, which is the second half of the finding and the one with a practical consequence. The `ceiling` object holds the score the answer key itself earns under the standard recipe. At the last cell of the sweep, ambient-referenced kNN preservation at $k = 10$ rates the exact unfolding 0.58 and Q_{NX} rates it 0.714. Any reader shown 0.58 would call that embedding a partial failure. It is the answer.

One more reading is available here and it corrects a tempting summary. The result so far invites "use the right reference geometry and the metrics behave", and that is only half true. Swapping ambient for the exact chart fixes the *ordering*: with the answer key as reference, nothing inverts anywhere. It does not give trustworthiness and continuity a range to answer in. Scored against

the exact chart at the hardest cell of the sweep, the two-component PCA — the configuration whose true error is 0.212 — still earns trustworthiness 0.995 and continuity 0.995. A perfect reference geometry and a wrong embedding, and the metric reports near-perfection. The reference decides who wins; the metric decides whether the win means anything, and for these two it does not.

9.5.2 Four metrics, one number

The four are reported side by side in almost every evaluation, and read as four independent lines of evidence. On the audit they gave four different answers. On the benchmark grid — nine methods, three noise models, the whole θ sweep, 9199 scored cells — they largely do not.

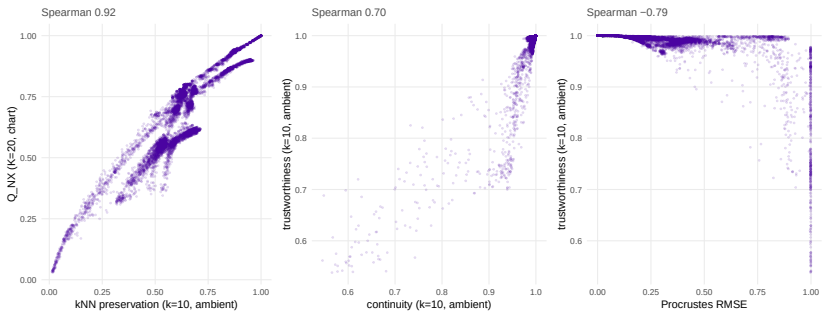


Figure 9.5: Metric agreement across the benchmark grid. Each point is one scored cell. The two rank metrics in the left panel are computed at different neighbourhood sizes against different reference geometries and still move together; the right panel is the gap this chapter is about.

Figure 9.5, left panel: Q_{NX} and kNN preservation rank the grid's cells at Spearman 0.921 — across 9199 cells, at *different* neighbourhood sizes, against *different* reference geometries. They are near-redundant despite being configured to ask two different questions, because both count neighbour-set overlap and neighbour sets are what varies. Trustworthiness and continuity agree at 0.695. Reporting three of these is not three pieces of evidence; it is one piece of evidence and two restatements, and the practice of tabling them together (Espadoto et al., 2021) hides that.

The four fall into two families rather than four instruments, and the split is the one the definitions predict. Q_{NX} and kNN preservation count how much of a neighbour *set* survived, and their overlap forms are the same number, so correlation near one is the expected result rather than a discovery. Trustworthiness and continuity weight each lost or invented neighbour by *how far* it moved in ranks, which is a different functional and gives them the looser agreement above. A reader wanting two independent readings should take one from each family and a distance-based score alongside — which is what `rank_metrics()` returns in one pass and what the book's grid records per cell.

The right panel is what a chapter on evaluators is for. 74.1 per cent of the grid's scored cells have trustworthiness above 0.99, and 46.6 per cent of *those* have a normalised Procrustes error above 0.2 — an embedding that has lost a

fifth of the chart. The grid's median cell scores 0.227 on Procrustes and 0.996 on trustworthiness. A metric whose median is that close to its ceiling is not measuring the range it is being asked about.

9.6 Diagnostics

9.6.1 Does k change the answer?

It does, and this is where the chapter's plan of record has to be corrected against its own artefact.

The contribution was specified as a **threshold law in** θ: rank metrics are invariant to monotone contraction until neighbour identity changes, so they should invert only above a fold angle of about 0.8 while stress inverts everywhere. The first half is right and is why the rank metrics behave at all differently from stress. The threshold is not.

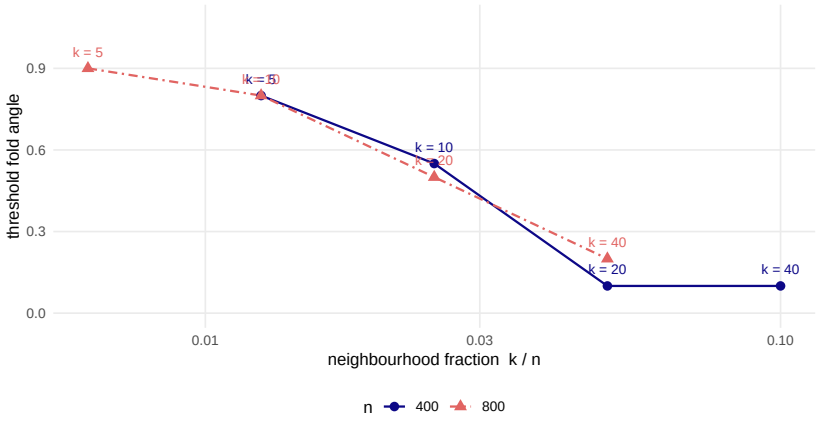


Figure 9.6: Where continuity's inversion sets in. The threshold is the smallest fold angle from which the evaluator inverts on at least half of seeds and never recovers. Plotted against the neighbourhood fraction k/n, the two sample sizes fall on one curve; plotted against k alone they would not. Sample size is encoded by colour, shape and line style together.

Figure 9.6 locates it. At the smallest neighbourhood the threshold is 0.9, which is roughly where the plan expected it. At the largest it is 0.2 — near the flat sheet, fourteen grid steps earlier. A single threshold in θ does not exist, because k moves it across most of the available range.

What the two sample sizes do is settle what k is a proxy for. Plotted against the neighbourhood *fraction* k/n the two curves nearly coincide: at $k/n = 0.0125$ the threshold is 0.8 at the smaller sample and 0.8 at the larger — the same value, at half the neighbourhood size and twice the points. The mechanism follows: a k-neighbourhood covers a chart radius that scales as $\sqrt{k/n}$, and continuity inverts once that radius is large enough that a typical neighbourhood spans a crease. Fold angle and neighbourhood fraction are two ways of buying the same thing.

So the law the chapter can state is a threshold surface in $(\theta, k/n)$ for **continuity**, and for continuity alone: at the larger sample size no other metric has a $\theta_{1/2}$ at any k in the design. The coincidence with Chapter 5's analytic short-circuit onset that the plan anticipated is *not* established here — this artefact does not measure the short circuit, and the two would have to be run against each other before either could be said to predict the other.

9.6.2 What to report when you have no answer key

Three recommendations follow, and they cost nothing.

Report the reference geometry and the neighbourhood size with the score. "Trustworthiness 0.99" is not a result. The audit's whole content is that the same metric on the same embedding answers differently depending on which distance matrix it was handed, and by more than the difference between a perfect embedding and a bad one.

Report the score against its floor. `metric_floor()` returns the smallest normalised Procrustes error *any* d-dimensional configuration of these points could achieve — the tail of the chart's own spectrum, by Eckart–Young (Eckart and Young, 1936).

```
floor_2d <- metric_floor(m, 2L)
floor_1d <- metric_floor(m, 1L)
prod4    <- product_manifold(m, sample_manifold(p, theta = 0.5, n
 ↪  = N_DEFAULT,
                                                 seed =
                                                  ↪  BENCH_SEEDS[
                                                  ↪  2]))
floor_prod <- metric_floor(prod4, EMBED_DIM)
grid_floor <- unique(grid$floor[!is.na(grid$floor)])
```

On this sample it returns 0, and the grid's own `floor` column is 0 in every cell it holds. That is worth one sentence rather than a column: a two-dimensional chart embedded in two dimensions has nothing unreachable in it, so every error reported in Chapter 10 is loss the method is responsible for. Force the same chart into one dimension and the floor is 0.652; take Chapter 8's product of two folded sheets, an intrinsically four-dimensional object, and it is 0.649. A method scoring 0.649 has thrown away everything on the main grid and hit the bound exactly on the product. Reporting the number without its floor distinguishes neither.

Report the answer key's own score. The floor bounds the Procrustes metric and nothing else — there is no analogous bound for Q_{NX} or trustworthiness, and this chapter does not supply one. What it does supply is the empirical substitute that only a known-answer benchmark can produce: the `ceiling` object above is the score a *perfect* embedding attains under each convention. Where that number is 0.58, an embedding scoring 0.58 has not half failed. It has succeeded, and the shortfall belongs to the metric.

```
DU <- as.matrix(stats::dist(U))
set.seed(11L)
```

```
J <- matrix(stats::rnorm(2L * nrow(U)), ncol = 2L)
distortion <- list(
  `conformal bend` = function(t) {
    z <- complex(real = U[, 1L], imaginary = U[, 2L])^(1 + t)
    cbind(Re(z), Im(z))
  },
  shear  = function(t) cbind(U[, 1L] + t * U[, 2L], U[, 2L]),
  jitter = function(t) U + t * J)

match_error <- function(f, target, hi = 6) {          # bisect on
 ↪  the strength
  lo <- 1e-6
  for (i in seq_len(50L)) {
    mid <- (lo + hi) / 2
    if (reconstruction_error(f(mid), U) < target) lo <- mid else
     ↪  hi <- mid
  }
  f((lo + hi) / 2)
}

battery <- do.call(rbind, lapply(names(distortion), function(nm)
 ↪  {
  Y <- match_error(distortion[[nm]], 0.39)
  data.frame(distortion = nm,
             rmse  = reconstruction_error(Y, U),
             trust = trustworthiness(DU, Y, K_DEFAULT),
             cont  = continuity(DU, Y, K_DEFAULT),
             knn   = knn_preservation(DU, Y, K_DEFAULT),
             qnx   = qnx(Y, DU, K_QNX))
}))

knitr::kable(
  battery, digits = 3, row.names = FALSE,
  col.names = c("distortion", "Procrustes RMSE",
   ↪  "trustworthiness",
               "continuity", "kNN preservation", "Q_NX"),
  caption = "Three distortions of one chart, each tuned by
   ↪  bisection to the same normalised Procrustes error. The rank
   ↪  metrics do not agree about them, and the disagreement is
   ↪  not small.")
```

And a fourth, which is really a warning about the first three. Optimism is a band, not a number. Table 9.2 applies three distortions to the same chart and tunes each to an identical true error of 0.39. Trustworthiness rates them 1, 0.998, 0.889. kNN preservation rates the same three configurations 0.927, 0.589, 0.0593 — a spread of 0.87 at fixed true error. There is no correction factor to apply, because there is no factor: what the rank metrics see is the *kind* of distortion, and the Procrustes error does not determine it.

Table 9.2: Three distortions of one chart, each tuned by bisection to the same normalised Procrustes error. The rank metrics do not agree about them, and the disagreement is not small.

distortion	Procrustes RMSE	trustworthiness	continuity	kNN preservation	Q_NX
conformal bend	0.39	1.000	1.000	0.927	0.943
shear	0.39	0.998	0.998	0.589	0.739
jitter	0.39	0.889	0.887	0.059	0.178

9.7 Where it fails

One family, one wrong candidate. Every number above is a Miura at one tessellation size against one competitor. PCA is the right competitor for an ambient-referenced metric and a poor one for anything else, and an evaluator that survives this test has survived one test. The audit's inversion is a property of the pair, and a different wrong embedding — a short-circuited Isomap, say — would produce a different table.

Two sample sizes are not a sample-size law. The k/n collapse in Figure 9.6 rests on two values of n that differ by a factor of two, and on a threshold statistic estimated from 20 seeds per cell on a grid of 20 fold angles. It is consistent with $\sqrt{k/n}$ and it does not exclude much else. A third sample size is the cheapest experiment this chapter suggests and it has not been run.

The pilot's own headline pooled over n, and should not have. The narrow run that decided this chapter existed reported one table in k and θ, averaged over both of its 2 sample sizes. On the full sweep kNN preservation inverts on 30.4 per cent of cells at the smaller sample size and 5.44 per cent at the larger; the pooled figure is 17.9 per cent, which describes neither. That is recorded here rather than quietly fixed, because it is the same error this chapter warns about in other people's tables.

Nothing here is a claim about methods. The known-answer test scores evaluators. Which method to use, and whether this benchmark can tell them apart at all, is Chapter 10's question and is settled by a pre-registered rule rather than by these curves.

What this chapter does not settle. Whether an evaluator that mishandles a crease also mishandles a real dataset is not established and cannot be, because the property being exploited — an exact chart — is the property real data lacks. The defensible reading is the disqualification argument: a metric that ranks the right answer below a wrong one where the right answer is *known* has not earned trust where it is not. That is a necessary condition, it asserts nothing about biology, and it is all this chapter claims.

9.8 Reproduce this

The artefact is `data/processed/evaluator-audit.rds`, produced by `scripts/run-evaluator-audit.R`, which also writes the pilot under `--pilot`. It carries its own provenance and this chapter refuses to render against a

different run.

```
Rscript scripts/run-evaluator-audit.R --pilot   # the narrow
 ↪  decision run
Rscript scripts/run-evaluator-audit.R           # the full sweep
```

The metric definitions are in `R/metrics.R` and every identity used above is a test in `tests/testthat/test-metrics.R`: that reflection alone is free, which is what forbids a determinant-one fit; that $A(k)$ agrees with a brute-forced worst case derived without reference to the paper; that the normalised Procrustes statistic agrees with `vegan`'s symmetric form; that `knn_preservation(measure = "overlap")` is exactly $Q_{NX}(k)$; that on a flat sheet the chart *is* the ambient geometry and on a folded one it is not; and that `metric_floor()` is attained by the truncated chart and nothing gets below it. Run them with `Rscript tests/testthat.R`.

9.9 Further reading

Lee and Verleysen's textbook (Lee and Verleysen, 2007) and their co-ranking paper (Lee and Verleysen, 2009) are the reference for everything in the rank-metric section, and Kaski et al. (Kaski et al., 2003) is the primary for trustworthiness and continuity — with Venna and Kaski (Venna and Kaski, 2001) for priority and the information-retrieval reading (Venna et al., 2010) for why the pair is a pair. Espadoto et al. (Espadoto et al., 2021) survey what is actually reported in practice, which is the target of the agreement result above.

On the adversarial side, Machado et al. (Machado et al., 2025) is the direct complement to this chapter and should be read alongside it. Schoeneman et al. (Schoeneman et al., 2017) make the same move in a different direction: they build a benchmark whose answer key is exact because the parameterisation is unit speed, and Chapter 11 measures this book's construction against theirs.

Chapter 10

Benchmark results

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

Which method survives — and can the benchmark tell?

10.1 What this chapter answers

Part II fitted each method to a folded sheet one configuration at a time. This chapter fits all of them to the whole sweep at once — every method against every fold angle under every noise model, twenty seeds deep — and reads the table.

It asks two questions, in a fixed order, and the order is the argument. The first is whether the benchmark can tell the methods apart at all: whether the spread *across* methods exceeds the spread across repeated runs *within* one method. Only if it does is the second question — which method wins — about methods rather than about seeds. A benchmark that answers the second without checking the first publishes a ranking of its own noise.

The short answer to the first is yes, everywhere on this grid, by a margin that narrows sharply where the sheet is most folded. The answer to the second is that it depends on θ and on the noise model, so that "method X is best" is not a well-formed sentence here unless both are stated. The rest is those two answers with their conditions attached, and a rule — fixed in code before the grid existed — for turning them into a decision.

10.2 Setup

```
grid <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")
prov <- attr(grid, "provenance")
ok   <- grid[grid$ran, ]
```

The grid is committed, not computed at render time. `read_run()` refuses any

Table 10.1: The benchmark grid, read off the artefact rather than asserted.

Factor	Levels	Value
pattern	1	miura
θ	20	0 to 0.95 in steps of 0.05
noise	3	ambient, none, outlier
method	9	autoencoder, diffusion, isomap, laplacian, lle, mds, pca, tsne, umap
seed	20	1001 to 1020
cells	1200	pattern $\times$ θ $\times$ noise $\times$ seed
rows	10800	one per method per cell

file whose contents differ from the run these numbers were read off, so a regenerated artefact is a build failure rather than a silently changed sentence. Producing it took 4.9 hours of wall clock across 4 forked workers — 19.6 core-hours for 9,199 fits, or about 59 core-seconds per sampled cell, metrics included.

`grid` keeps every row the producer wrote, including the ones with no metrics; `ok` is the subset that fitted. Both are used below, and which one a number came from is stated, because the difference between them is itself a result.

10.3 What the grid holds

```
step <- unique(round(diff(sort(unique(grid$theta))), 10))
design <- data.frame(
  Factor = c("pattern", "$\\theta$", "noise", "method", "seed",
   ↪  "cells", "rows"),
  Levels = c(length(unique(grid$pattern)),
   ↪  length(unique(grid$theta)),
             length(unique(grid$noise)),
              ↪  length(unique(grid$method)),
             length(unique(grid$seed)),
             nrow(unique(grid[, c("pattern", "theta", "noise",
              ↪  "seed")])),
             nrow(grid)),
  Value  = c(paste(unique(grid$pattern), collapse = ", "),
             sprintf("%s to %s in steps of %s", min(grid$theta),
              ↪  max(grid$theta), step),
             paste(sort(unique(grid$noise)), collapse = ", "),
             paste(sort(unique(grid$method)), collapse = ", "),
             sprintf("%d to %d", min(grid$seed), max(grid$seed)),
             "pattern $\\times$ $\\theta$ $\\times$ noise
              ↪  $\\times$ seed",
             "one per method per cell"))
knitr::kable(design, align = "lrl", caption = "The benchmark
 ↪  grid, read off the artefact rather than asserted.")
```

There is no pattern axis. One family folds (Chapter 8), and a factor with a single level is not a factor. The three noise conditions are a clean sheet, isotropic

ambient jitter, and a sparse outlier contamination; the θ sweep stops short of the flat-folded state, where the sheet has no extent in one direction and every method is embedding a degenerate object.

```
n_unavail  <- sum(grid$status == "unavailable")
n_declined <- sum(grid$status == "declined")
run_rate   <- tapply(grid$ran, grid$method, mean)
fate <- factor(ifelse(!grid$ran,
                      ifelse(grid$status == "unavailable",
                       ↪  "declared unavailable",
                             "declined the cell"),
                      ifelse(is.na(grid$tuning), "ran as asked",
                       ↪  "ran after retuning")),
               levels = c("ran as asked", "ran after retuning",
                          "declined the cell", "declared
                           ↪  unavailable"))
max_floor <- max(grid$floor)
```

Of 10,800 rows, 9,199 carry metrics. The remaining 1,601 are results, not gaps: 1,200 autoencoder rows declared unavailable by the registry until Chapter 7, and 401 rows where a method was offered a cell and returned nothing. `report_failures()` prints the same accounting from the producer.

The `floor` column is 0 in every row. The irreducible loss of a p-dimensional chart embedded in d dimensions is the tail of the chart's spectrum, and here the chart is two-dimensional and d is 2, so there is no tail. Every error reported below is therefore loss the method is responsible for, and "scored against what was achievable" and "scored in absolute terms" coincide on this grid. They do not coincide on the product construction of Chapter 8, which is the artefact built to make the floor positive.

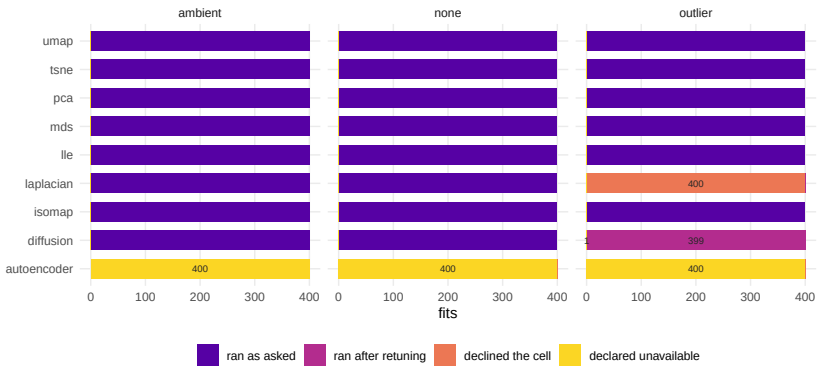


Figure 10.1: What happened to every fit, by method and noise condition. Counts are printed on each segment that is not a clean run, so nothing here is carried by colour alone.

Figure 10.1 shows where the losses sit, and they are structured rather than scattered. Laplacian eigenmaps declined every outlier cell; the diffusion map ran under outliers only after widening its own kernel bandwidth, which the

`tuning` column records.

10.4 Where the benchmark has resolving power

A ranking is worth reading only if the differences it orders are larger than the noise it was measured through. `R/selection.R` names that quantity and fixes it in code:

$$R = \frac{\text{spread across methods}}{\text{spread within a method, across seeds}},$$

with both spreads taken of the same quantity — excess over the irreducible floor — so that R is dimensionless. The numerator is the standard deviation of the per-method means in a regime; the denominator is the median, over cells, of the seed-wise standard deviation inside one cell. Above one, the methods differ by more than a rerun of the same method does. At or below one, any ordering read off the grid is seed noise with a rank imposed on it.

```
rp <- do.call(rbind, lapply(sort(unique(grid$noise)),
 ↪  function(nz)
  do.call(rbind, lapply(THETA_GRID, function(t) {
    s <- select_method(grid, theta = t, noise = nz)$resolving
    data.frame(noise = nz, theta = t, R = s$R, between =
     ↪  s$between, within = s$within)
  })))
worst     <- rp[which.min(rp$R), ]
pooled_sd <- tapply(ok$rmse, interaction(ok$method, ok$theta,
 ↪  ok$noise, drop = TRUE),
                    stats::sd)
ends <- function(nz) round(rp$R[rp$noise == nz][c(1L,
 ↪  length(THETA_GRID))], 1)
```

Three details of that definition earn their keep. The denominator is *within a cell*, not within a method across the regime, because a method's variation from one fold angle to the next is difficulty and belongs in the numerator. It is a **median** rather than a mean, because one wildly unstable method would otherwise set the noise level for all eight — the pooled within-cell spread on this grid has a median of 0.019 and a maximum of 0.31, and the gap between those two is t-SNE. And it is computed on excess over the floor rather than on raw error, which on this grid changes nothing because the floor is 0, and on Chapter 8's product suite changes a great deal.

R is above the floor of 1 in all 60 regimes of Figure 10.2. The chapter was specified to report $R < 1$ wherever it held, and on this grid it holds nowhere; that is the finding, and it is a more comfortable one than the specification expected. The weakest cell is ambient noise at $\theta = 0.95$, where $R = 1.33$ — the between-method spread of 0.031 against a within-method spread of 0.023. Two ticks of θ earlier the same condition resolves twice as well. The outlier condition resolves best of the three, by an order of magnitude, and that is not

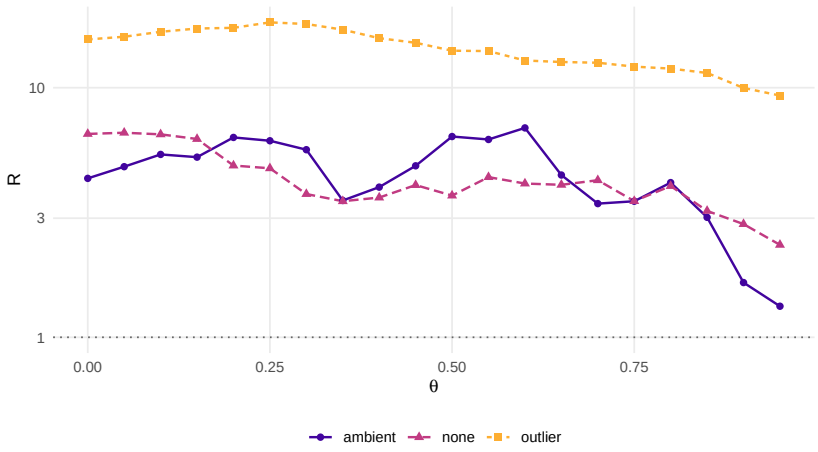


Figure 10.2: Resolving power against the folding parameter, one line per noise condition, on a logarithmic scale. The dotted line is the pre-registered floor below which the selection rule declines to rank anything.

a compliment to the benchmark: it separates the methods there because one of them fails completely, which is a wide gap rather than an informative one.

The trend across the sweep is the part worth carrying forward. Every condition loses resolution as the sheet closes — from 6.5 to 2.3 on the clean sheet, 4.3 to 1.3 under ambient noise, 15.6 to 9.3 under outliers — so the benchmark is least able to discriminate exactly where its geometry is most distinctive. That is not a paradox: at large θ every method is failing, and methods that are all failing look alike. It does mean that a regime chosen for being difficult is a regime chosen for weak resolving power, and Chapter 12 has to pick one knowing that.

The number to resist is the seed count. R is a ratio of two spreads and neither is a standard error, so running forty seeds instead of twenty estimates both more precisely and moves the ratio nowhere; `tests/testthat/test-selection.R` demonstrates exactly that. Raising the seed budget until something separates would be the same abuse as raising it until a p-value falls, with the advantage of looking like diligence. The budget was fixed in advance and is reported in the design table above.

Two things this section does **not** establish. It does not show that a crease pattern discriminates better than a classical benchmark: Chapter 11 measures that directly and finds the opposite, which is why this book's spine is the irreducible-loss bound and not the claim that folded sheets are a harder test. And R is a property of a regime, not of the generator — it is computed here at fixed n, fixed k and one pattern size, and each of those moves it.

10.5 Results

```
mm <- aggregate(rmse ~ method + theta + noise, ok, mean)
```

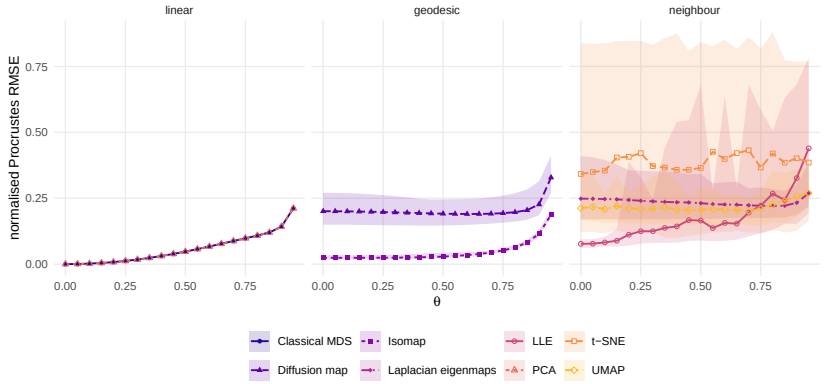


Figure 10.3: Normalised Procrustes reconstruction error against the folding parameter on the clean sheet, grouped by what each method consumes. The line is the mean over twenty seeds; the band is the full seed range. PCA and classical MDS coincide exactly and plot on top of each other.

```
mm$rank  <- ave(mm$rmse, mm$theta, mm$noise, FUN = rank)
mm$label <- vapply(mm$method, function(m)
 ↪  METHOD_REGISTRY[[m]]$label, character(1))
lead <- do.call(rbind, lapply(split(mm, list(mm$theta,
 ↪  mm$noise)), function(s)
  data.frame(theta = s$theta[1], noise = s$noise[1], best =
   ↪  s$method[which.min(s$rmse)])))
cross <- min(lead$theta[lead$noise == "none" & lead$best ==
 ↪  "isomap"])
at <- function(m, i, nz = "none") {
  mm$rmse[mm$method == m & mm$noise == nz & mm$theta ==
 ↪  THETA_GRID[i]]
}
iso_rank <- mm$rank[mm$method == "isomap" & mm$noise == "outlier"
 ↪  &
                    mm$theta %in% THETA_GRID[c(1L,
 ↪  length(THETA_GRID))]]
qq <- aggregate(qnx ~ method + theta, ok[ok$noise == "none", ],
 ↪  mean)
qq <- qq[order(qq$theta), ]
q_cross <- min(qq$theta[qq$method == "isomap"][
  qq$qnx[qq$method == "isomap"] > qq$qnx[qq$method == "pca"]])
sd_by <- function(m) {
  s <- ok[ok$method == m, ]
  median(tapply(s$rmse, interaction(s$theta, s$noise, drop =
   ↪  TRUE), stats::sd))
}
```

Figure 10.3 is the headline on the clean sheet. At $\theta = 0$ the sheet *is* the chart, and the linear pair recovers it to 4.9e-16 while Isomap — which consumes

geodesics it must first estimate — pays 0.024 for the estimation (Tenenbaum et al., 2000; Torgerson, 1952). The linear error then climbs monotonically with the fold, because folding is a strict ambient contraction and a linear map cannot undo one, while Isomap's stays nearly flat: its assumption is satisfied exactly on a rigid folding, and the geodesics it estimates do not change as the sheet closes.

The two curves cross. Isomap takes the lead at $\theta = 0.4$ — 0.025 against 0.024 one step earlier, 0.026 against 0.031 at the crossing — and holds it to the end of the sweep, finishing at 0.188 against 0.212. Averaging over θ would name the linear pair, because 8 of the 20 fold angles fall on the flat side of the crossing and the errors there are small. That average is a statement about the grid's design, not about the methods.

The crossing moves when the metric does. This book reports two headline numbers per cell, deliberately: Procrustes RMSE asks whether the embedding *is* the chart, and chart-referenced Q_{NX} asks only whether it has the chart's neighbourhoods, which any diffeomorphism of the plane preserves. Read against Q_{NX}, Isomap does not pass the linear pair until $\theta = 0.6$ — 0.920 against 0.917 — four steps of the sweep later than under RMSE. The gap between the two crossings is the distance between "the linear projection is still the better *map*" and "it is still the better *neighbour list*", and naming a winner without naming the metric hides it.

The other five methods break rather than cross. LLE (Roweis and Saul, 2000) starts respectably at 0.077 and ends at 0.439, worse than anything else on the clean sheet at the deepest fold: its local weights are fitted to neighbourhoods that folding has brought into contact across facets. The diffusion map and Laplacian eigenmaps are flat and mediocre throughout, averaging 0.204 and 0.236, which is roughly what a method that recovers the chart up to an unknown nonlinear warping scores against a metric permitting only similarity. And t-SNE averages 0.387 and is last at every fold angle — not a finding about t-SNE, which was never built to reproduce distances, but the clearest case on this grid of a method scored against a target it does not consume.

Figure 10.4 puts the three noise conditions side by side, and the third one breaks the story the first two tell. Under sparse outliers UMAP (McInnes et al., 2018) leads at every fold angle, at 0.353 against 0.422 for the linear pair, while the diffusion map (Coifman and Lafon, 2006) sits at 0.999 — which is the score of an embedding that explains none of the chart's variance — and Laplacian eigenmaps (Belkin and Niyogi, 2003) does not produce one at all. Isomap's advantage at large θ survives as a climb from 4th place to 3rd, but UMAP leads at all 20 fold angles and Isomap never reaches the top under this contamination.

So the ordering is not a property of the methods. It is a property of (method, θ, noise), and any of the three sentences "the linear methods win", "Isomap wins" and "UMAP wins" can be read off this one grid.

10.6 Diagnostics

Four things in the table are worth naming before anyone reads a ranking off it.

PCA and classical MDS are the same map here. Their reconstruction

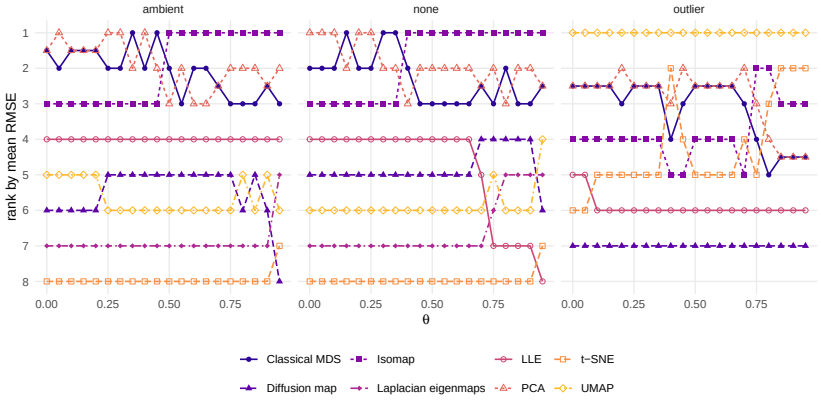


Figure 10.4: Rank by mean reconstruction error against the folding parameter, one panel per noise condition. Rank one is at the top. The leader changes with the fold angle under two conditions and is a different method entirely under the third.

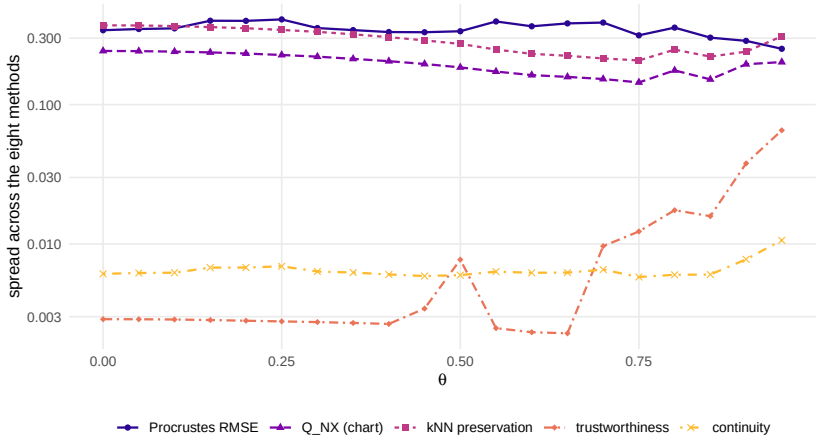


Figure 10.5: How far apart the metrics place the eight methods, on the clean sheet: the range of the per-method means at each fold angle, on a logarithmic scale.

errors differ by at most 2.8e-15 across the grid and their Q_{NX} scores are bit-identical, because classical MDS of a Euclidean distance matrix *is* the principal-component projection (Torgerson, 1952). Their alternation between ranks one and two in Figure 10.4 is floating-point tie-breaking, not a result.

Two of the five metrics cannot separate anything. Figure 10.5 is the spread across methods, metric by metric. Trustworthiness and continuity, scored at $k = 10$ against *ambient* distance (Kaski et al., 2003), place the eight methods within a median range of 0.003, while Procrustes RMSE spreads them over 0.35. Chart-referenced Q_{NX} at $K = 20$ (Lee and Verleysen, 2009) sits between the two. This is the optimism Chapter 9 quantifies; here it is enough to note that trustworthiness only begins to move at all beyond about two-thirds of the sweep.

t-SNE's seed spread swallows most of the field. Its median within-cell standard deviation across seeds is 0.26, against 0.003 for the linear pair and 0.004 for Isomap — a single t-SNE run on this grid is an anecdote (van der Maaten and Hinton, 2008), and the band in Figure 10.3 is why the book reports twenty.

Isomap never needed graph repair. `k_effective` equals the requested $k = 10$ in all 1200 fits, so the neighbourhood graph was connected everywhere in the sweep. That is not the same as saying no short circuit occurred (Balasubramanian and Schwartz, 2002): a shortcut edge between facets connects the graph rather than breaking it, and it is the rise in Isomap's error past $\theta = 0.4$ that would carry it.

10.7 Where it fails

```
armB <- read_run("data/processed/e1-armB.rds",
 ↪  "cbf9ce13782a556d")
cc   <- aggregate(rmse ~ creases + method, armB, mean)
iso_range <- diff(range(cc$rmse[cc$method == "isomap"]))
```

The grid varies four things and holds a great many fixed, and the fixed ones bound every sentence above.

One pattern, at one size. A single Miura at 6 by 6 cells. Crease count is not an axis here; it was swept separately by Chapter 11's second arm, which found Isomap ahead of the linear pair at all 3 crease counts tested at matched branch separation, with Isomap's own error moving by 0.01 across a sixteen-fold change in creases. That is a reason for holding it fixed, not a licence to generalise from one size.

One sample size, one neighbourhood, one target dimension. Every fit used $n = 800$, $k = 10$ and $d = 2$. Chapter 9 sweeps k; nothing here does, and a k-sensitive method (Roweis and Saul, 2000) is being reported at one setting of it.

No autoencoder. 1,200 rows of the grid are a declaration rather than a measurement, and every comparison above is over eight methods, not nine.

Two noise models, both synthetic. Isotropic ambient jitter and sparse

outliers are the two failure modes easiest to write down, not the two most common in data. Neither is structured, correlated with position on the sheet, or heteroscedastic, and the outlier result — a complete reordering of the field — is a warning that the noise model may matter more than the geometry does.

A floor of 0. Reporting against what was achievable is informative only when something was unachievable, and on a two-dimensional chart in two dimensions nothing is.

Only one pattern family folds at all, which is Chapter 8's subject. This grid cannot support any claim about variety across families, and none is made.

10.8 The pre-registered selection rule

Chapter 12 has to pick a method with this grid. The value of that choice rests entirely on the rule having been fixed before the answer was known, so it is committed as code — `select_method()` in `R/selection.R`, a total function of the grid's columns, registered before the grid it reads had ever been generated. Prose can be read generously afterwards; a function cannot.

Three thresholds, and nowhere else: a run-rate floor of 0.9, below which a method is not a candidate however well it scores; a reportable difference of 0.02 normalised RMSE, below which a lead is a tie; and a resolving-power floor of 1, below which the rule declines and the chapter reports a null result.

```
window <- THETA_GRID[THETA_GRID >= THETA_GRID[13]]
r_none <- select_method(grid, theta = window, noise = "none")
r_amb  <- select_method(grid, theta = window, noise = "ambient")
r_all  <- select_method(grid, theta = THETA_GRID, noise =
 ↪  unique(grid$noise))
class_excess <- sort(tapply(ok$rmse - ok$floor, ok$consumes,
 ↪  mean))
```

The rule does not choose the regime; that is Chapter 12's argument to make from its data, before it fits anything. Applied here to a deep-fold window as an illustration, it selects **isomap** on the clean sheet, by 0.037 at $R = 3.44$. Under ambient noise, in the same window, the leading margin falls to 0.003 — inside the reportable threshold — so the rule declares a tie of 3 methods and breaks it on failures, then on assuming least, and returns **mds** instead. Pooled over the whole grid it excludes autoencoder and laplacian on the run-rate floor and returns **mds** again. Three regimes, three defensible answers, one rule: which is the point, and the reason the regime must be argued for in the open.

One clause needs a caveat. Ties break toward the method that *assumes least* — ambient, then geodesic, then neighbourhood — which is an argument about assumptions, not a prediction about scores. The grid ranks the classes the other way: mean excess over the floor is 0.22 for the geodesic class, 0.30 for ambient and 0.32 for neighbourhood, in that order and in every noise condition separately, and Chapter 8 measures the same ordering on the product suite. The tie-break stands, because it applies only where the benchmark *cannot separate* — but it is not this book's estimate of which class does better.

The declining branch did not fire anywhere on this grid. It was written to fire and is tested to fire, and a rule that could not have declined would have established nothing by not declining.

10.9 Reproduce this

The grid is regenerated with `Rscript scripts/run-benchmark-grid.R --cores 4` from the repository root, and `--shard i/n` followed by `--merge` spreads it over machines. That script is the provenance record for `data/processed/benchmark-grid.rds` and the only thing that writes it; it refuses to merge shards built from different versions of `R/`, and `--quick` writes to a different filename so a smoke test can never be mistaken for the book's evidence. The cell itself is `grid_cell()` in `R/grid.R`, shared with the Part II sweeps and the classical grid so that three producers cannot drift apart. `tests/testthat/test-selection.R` asserts that each of the three thresholds changes the decision, and that raising the seed count does not.

10.10 Further reading

Espadoto and colleagues (Espadoto et al., 2021) survey the projection-quality literature this chapter's metrics come from, and Lee and Verleysen (Lee and Verleysen, 2009) give the co-ranking framework behind Q_{NX}; Kaski and colleagues (Kaski et al., 2003) are the source of trustworthiness and continuity, including the scaling constant. For the methods themselves, the original papers remain the clearest statements of what each one consumes (Tenenbaum et al., 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Coifman and Lafon, 2006; van der Maaten and Hinton, 2008; McInnes et al., 2018), which is the property that predicts the regime changes above better than any of the scores does. What none of them supplies is a scale against which a score can be called good; that is Chapter 8's subject, and it is why this chapter's floor column is a placeholder rather than a measurement.

Chapter 11

Crease patterns vs. the Swiss roll

R. Heller · Date: 2026-07-29 · Last changed: 2026-09-07

Does the new benchmark actually add anything?

11.1 What this chapter answers

Yes — and not the thing this book set out to claim.

The claim under test was that a crease pattern is a *harder* benchmark than the Swiss roll: that folding reaches difficulty the classical surfaces cannot. Experiment E1 was run before this chapter was drafted, precisely so that the prose could not outlive a refutation, and it refuted it. At matched difficulty a folded Miura is **easier** than a Swiss roll, and it tells the methods apart about five times less well.

What survives is narrower and better founded. The Swiss roll is one fixed geometry, and the answer key usually shipped with it is not its isometric chart. The severed sphere has no exact answer key and, by Gauss's theorem, cannot have one. A crease pattern is a whole family indexed by θ whose chart is exact at every point of it. The contribution is the answer key and the axis, not the difficulty — which is why Chapters 8 and 9 are where it pays, and this chapter is where it does not.

11.2 Setup

```
classic <- read_run("data/processed/classic-grid.rds",
 ↪  "d71fb3ba0ad5de68")
crease  <- read_run("data/processed/benchmark-grid.rds",
 ↪  "2483a468d82d4d64")
e1      <- read_run("data/processed/e1-controlled.rds",
 ↪  "01d82cfe8f45bb44")
```

```
armB    <- read_run("data/processed/e1-armB.rds",
 ↪  "cbf9ce13782a556d")
```

`classic-grid.rds` is the Swiss roll, the S-curve and the severed sphere scored through the same `grid_cell()` the crease grid uses: same methods, same metrics, same floor. Nothing on either side is special-cased. The two E1 artefacts are the controlled arm and the crease-count arm of the experiment that decided this chapter's claim, and `scripts/experiment-e1.R` produced both before it was drafted.

11.3 Background

11.3.1 The Swiss roll has an exact chart, and it is usually not the one shipped

A Swiss roll is a rectangle of paper wound onto a spiral. It arrived with Isomap (Tenenbaum et al., 2000), appeared the same year in the LLE paper beside the S-curve (Roweis and Saul, 2000), and has been the first figure of the subject ever since (Lee and Verleysen, 2007).

It is developable, so it *does* have an isometric chart, and the chart is cheap: arc length along the spiral paired with height. For the Archimedean spiral $r = at$ the arc length is closed form,

$$s(t) = \frac{a}{2}\left[t\sqrt{1+t^2} + \operatorname{asinh} t\right],$$

which is `archimedean_arclength()` in `R/baselines.R`. So the Swiss roll is not short of ground truth. What it is usually short of is the *right* ground truth: the key shipped with it is the generating angle, and arc length is not proportional to angle.

```
roll_ang <- swiss_roll(N_DEFAULT, turns = 2, seed =
 ↪  BENCH_SEEDS[1],
                       isometric = FALSE)
eul_ang  <- swiss_roll(N_DEFAULT, turns = 2, seed =
 ↪  BENCH_SEEDS[1],
                       isometric = FALSE, spiral = "euler")
key_gap  <- c(archimedean = reconstruction_error(roll_ang$truth,
 ↪  roll_ang$iso_chart),
              euler       = reconstruction_error(eul_ang$truth,
               ↪  eul_ang$iso_chart))
key_qnx  <- c(archimedean = qnx(roll_ang$truth,
 ↪  roll_ang$iso_chart, K = K_QNX),
              euler       = qnx(eul_ang$truth,
               ↪  eul_ang$iso_chart,  K = K_QNX))

sphere   <- severed_sphere(N_DEFAULT, cap = 0.4, seed =
 ↪  BENCH_SEEDS[1])
stretch  <- (pi - sphere$cap) / sin(sphere$cap)    # r / (R
 ↪  sin(r/R)) at the rim
```

```
n_sphere <- sum(classic$family == "severed_sphere" & classic$ran)
cat(strwrap(tryCatch(require_exact_truth(sphere, "a
 ↪  reconstruction score"),
                     error = conditionMessage), 72), sep = "\n")
```

```
a reconstruction score needs exact ground truth, and this sample does
not have it: severed_sphere carries a conventional chart (equidistant
(a convention; no isometric chart exists)). Scoring against it measures
agreement with a convention, not recovery of a structure.
```

Score the customary angle chart against the arc-length chart on one two-turn roll and the gap is 0.567 in normalised Procrustes RMSE, with Q_{NX} at 0.462. That is not a rounding difference. An evaluation against the angle chart measures recovery of a reparameterisation, and the factor by which it is wrong is a property of the roll rather than of the method under test.

The defect is known and was fixed in 2017. Schoeneman et al. proposed the Euler Isometric Swiss Roll (Schoeneman et al., 2017), replacing the Archimedean spiral with the Fresnel integrals; it has been used since (Mahapatra and Chandola, 2018). Their stated reason is that curvature proportional to arc length "ensures that isometry is preserved". The checkable reason is simpler: the Fresnel parameterisation is **unit speed**, so the parameter *is* arc length by definition. Label an Euler roll by its turning angle instead and the same error appears at 0.258 — smaller, and not zero.

Every Swiss roll in this book is scored against arc length. Figure 11.1 is why.

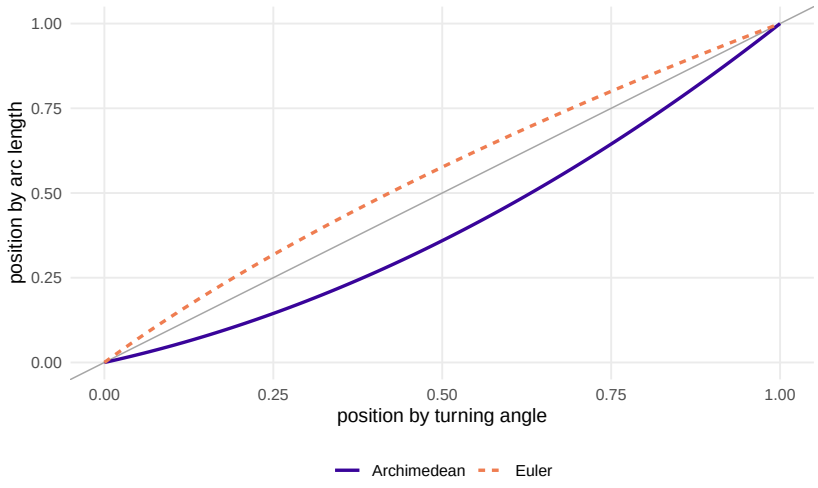


Figure 11.1: Position along the spiral measured two ways, on the same sample. The grey diagonal is where the two agree. The Archimedean roll's customary angle chart departs from arc length across the whole sweep; the Euler roll's departs far less, because its parameterisation is unit speed.

11.3.2 The severed sphere has no answer key at all

Say it bluntly. A sphere has constant positive Gaussian curvature, the plane has none, and curvature is intrinsic — Gauss's *Theorema Egregium*. No chart of any patch of a sphere preserves distances, however small the patch and however clever the chart. `severed_sphere()` therefore ships a **convention**: the azimuthal-equidistant chart about the retained pole, which gets every radial geodesic exactly right and stretches every circle around it by $r/(R\sin(r/R))$ — a factor reaching 7.04 at the rim of the largest patch in the classic grid.

The object knows this, and the refusal printed above is `require_exact_truth()` reading the flag. That guard fires only where something calls it. `grid_cell()` does not, so the classic grid scored the severed sphere 1280 times, exactly as it scored the roll. Those numbers exist, they are in the committed artefact, and this chapter reports none of them as recovery of a structure.

11.4 Difficulty has two axes, not one

11.4.1 The knob has to move the right thing

Both families have a difficulty parameter, and both parameters have to be checked, because the obvious version of each does nothing.

For the Swiss roll it is the turn count, and the obvious parameterisation — let the angle run further and the radius run with it — adds revolutions at *constant* sheet spacing, so the k-nearest-neighbour graph stays faithful however many turns are asked for. `swiss_roll()` instead holds the outer radius fixed and fits the requested revolutions inside it, so the radial gap between consecutive sheets falls as one over the turn count. That is Balasubramanian and Schwartz's gap-to-spacing ratio g/s (Balasubramanian and Schwartz, 2002), and it is the one axis both families can be plotted against.

For a crease pattern it is θ. But folding does two things at once: it brings distant sheets together *and* it lifts the sheet out of the plane. The second is what `non_planarity()` measures — the share of ambient variance no plane captures, the third principal eigenvalue over their sum.

```
axes <- local({
  seeds <- BENCH_SEEDS[1:3]
  one <- function(m, family, par) data.frame(
    family = family, par = par, np = non_planarity(m),
    gs = branch_gap(m)$ratio, sc = short_circuit_index(m))
  run <- function(vals, family, make) do.call(rbind, lapply(vals,
 ↪  function(v)
    do.call(rbind, lapply(seeds, function(s) one(make(v, s),
     ↪  family, v)))))
  rbind(
    run(sort(unique(classic$par[classic$family ==
     ↪  "swiss_roll"])), "Swiss roll",
        function(v, s) swiss_roll(N_DEFAULT, turns = v, seed =
         ↪  s)),
    run(sort(unique(classic$par[classic$family ==
     ↪  "severed_sphere"])),
```

```
        "severed sphere", function(v, s)
          severed_sphere(N_DEFAULT, cap = v, seed = s)),
    run(NA_real_, "S-curve", function(v, s) s_curve(N_DEFAULT,
      seed = s)),
    run(THETA_GRID, "Miura", function(v, s)
      sample_manifold(miura_ori(6L, 6L), theta = v, n =
        N_DEFAULT, seed = s)))
})
# The S-curve is added by hand because it has no difficulty
  parameter at all;
# aggregate() drops a group whose key is NA, and inventing a key
  would be worse.
cell <- aggregate(cbind(np, gs, sc) ~ family + par, axes, mean)
sc_rows <- axes[axes$family == "S-curve", ]
cell <- rbind(cell, data.frame(family = "S-curve", par =
  NA_real_,
                               np = mean(sc_rows$np), gs =
                                 mean(sc_rows$gs),
                               sc = mean(sc_rows$sc)))
np_top <- max(cell$np[cell$family == "Miura"])
np_bot <- min(cell$np[cell$family != "Miura"])

swept <- cell[cell$family != "S-curve", ]
ggplot2::ggplot(cell, ggplot2::aes(gs, np, colour = family, shape
  = family)) +
  # Finite bounds, not +/- Inf: under a log transform an infinite
    limit is NaN
  # and the band silently does not draw.
  ggplot2::annotate("rect", xmin = min(cell$gs) / 2, xmax =
  max(cell$gs) * 2,
                    ymin = np_top, ymax = np_bot, fill =
                      "grey70", alpha = 0.4) +
  ggplot2::geom_line(data = swept[order(swept$family, swept$par),
  ],
                     ggplot2::aes(group = family), linewidth =
  0.5) +
  ggplot2::geom_point(size = 2.3) +
  ggplot2::scale_x_continuous(transform = c("log10", "reverse"))
  +
  scale_colour_book(begin = 0.05, end = 0.78) +
  ggplot2::labs(x = "gap-to-spacing ratio (log scale, harder to
  the right)",
                y = "ambient non-planarity", colour = NULL, shape
                  = NULL) +
  theme_fold()
```

The ranges are **disjoint**. Across the whole θ sweep a six-by-six Miura covers non-planarity from 0.000 to 0.056; the Swiss rolls in the classic grid cover 0.090 to 0.124, the S-curve sits at 0.116, and the severed sphere starts at 0.218 and rises. The corridor between 0.056 and 0.090 is empty.

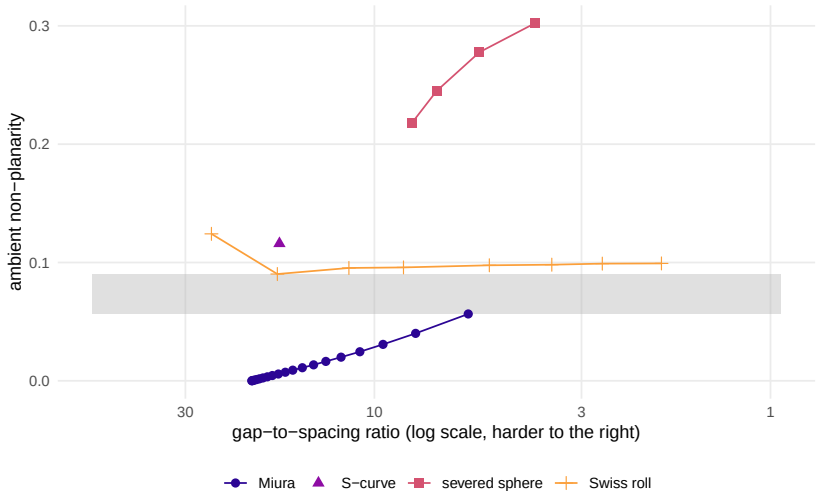


Figure 11.2: The two difficulty axes. Each family sweeps its own parameter; difficulty rises to the right. The shaded band is empty, so over this design there is no setting at which a folded Miura and a classical surface are comparable on both axes at once.

Figure 11.2 is the argument. A sweep of θ alone is not a controlled comparison, because it moves both axes together and the families never meet on the second. At the easy end of the g/s range a Miura is a flat plane, which PCA recovers exactly; a Swiss roll at the same separation is still curved, and PCA cannot.

The third column of `cell` is the reason g/s carries the first axis rather than the more obvious candidate. `short_circuit_index()` asks whether the neighbours a method would actually pick are neighbours on the surface too; it is scale-free, bounded above by one, and comparable across families, which makes it the better-behaved statistic of the two. It is also nearly constant over most of this design. Across the classic grid's turn range it falls only from 1.000 to 0.703, and almost all of that drop happens at the last two settings; across the whole θ sweep the Miura falls from 1.000 to 0.799; the severed sphere never leaves 0.999. A difficulty axis that is flat until the very end of the range cannot order the settings in between, so the comparison is made on g/s and the short-circuit index is reported beside it as the check that g/s is not measuring the sampler.

11.4.2 The fix, and what it costs

θ is not the only knob the generator has. Cell count dominates non-planarity: a small Miura at a wide opening angle and a deep fold reaches above anything the Swiss roll produces, while a six-by-six at the same angles stays an order of magnitude below. Sweeping (n_x, α, θ) decouples the axes and creates the overlap a θ sweep lacks. That is what E1's second arm did, and it is what `e1-controlled.rds` holds.

Crease count on its own does nothing. Arm B holds g/s fixed and varies the tessellation from nine facets to a hundred and forty-four; PCA's mean error

moves between 0.039 and 0.071, Isomap's between 0.028 and 0.038, and neither monotonically. Error is flat in crease count at matched separation. Whatever makes these patterns hard, it is not that they have creases.

11.4.3 One geometry against a family

This is the structural difference, and it survives everything the measurements below do to the rest of the chapter. A Swiss roll's turn count changes how tightly one shape is wound; it is a real knob, and `swiss_roll()` makes it one. A crease pattern gives a *family*: the same exact chart at every θ, so a method's score is a curve rather than a point and the shape of the curve is a property of the method. Nothing about the classical surfaces forbids such a sweep, and one is drawn here. What they cannot supply is the exact key at the far end of it.

Nor have they had to break the habit that goes with a fixed geometry. A Swiss roll's answer key is two-dimensional and its failures are legible, so the customary report is a picture: the unrolled sheet either comes out as a rectangle or it does not, and the reader is invited to agree. That is a serviceable way to show a method working and a poor way to compare two that both nearly work — which is exactly the regime the dispute over two-dimensional embeddings lives in (Chari and Pachter, 2023; Lause et al., 2024). A parameterised family forces the question into a number at every setting, and the number is what the rest of this chapter argues over.

11.5 Results

11.5.1 The head-to-head, matched on separation alone

Match each Swiss roll to the θ whose measured g/s is nearest, keep the ones landing inside the crease family's range, and score both sides from the committed grids at the same eight methods, the same twenty seeds and no noise.

```
mi <- cell[cell$family == "Miura", ]
ro <- cell[cell$family == "Swiss roll", ]
ro <- ro[ro$gs >= min(mi$gs) & ro$gs <= max(mi$gs), ]
near <- vapply(ro$gs, function(x) which.min(abs(mi$gs - x)),
 ↪  integer(1))
ro$theta    <- mi$par[near]
ro$mismatch <- abs(mi$gs[near] - ro$gs) / ro$gs

A <- classic[classic$family == "swiss_roll" & classic$noise ==
 ↪  "none" &
             classic$ran & classic$par %in% ro$par, ]
B <- crease[crease$noise == "none" & crease$ran & crease$theta
 ↪  %in% ro$theta, ]
stopifnot(all(A$seed %in% BENCH_SEEDS), all(B$seed %in%
 ↪  BENCH_SEEDS))
h <- merge(setNames(aggregate(cbind(rmse, qnx) ~ method, A,
 ↪  mean),
                    c("method", "roll_rmse", "roll_qnx")),
```

```
          setNames(aggregate(cbind(rmse, qnx) ~ method, B,
           ↪  mean),
                   c("method", "cr_rmse", "cr_qnx")))
h <- h[order(h$roll_rmse), ]
rho <- c(rmse = cor(h$roll_rmse, h$cr_rmse, method = "spearman"),
         qnx  = cor(h$roll_qnx,  h$cr_qnx,  method = "spearman"))
gap <- function(d, key) {
  w <- reshape(d[d$method %in% c("pca", "isomap"), c(key, "seed",
 ↪  "method", "rmse")],
               idvar = c(key, "seed"), timevar = "method",
                ↪  direction = "wide")
  abs(w$rmse.pca - w$rmse.isomap)
}
naive <- mean(gap(A, "par")) / mean(gap(B, "theta"))
```

That leaves 3 matched pairs, agreeing to within 6.8% in g/s.

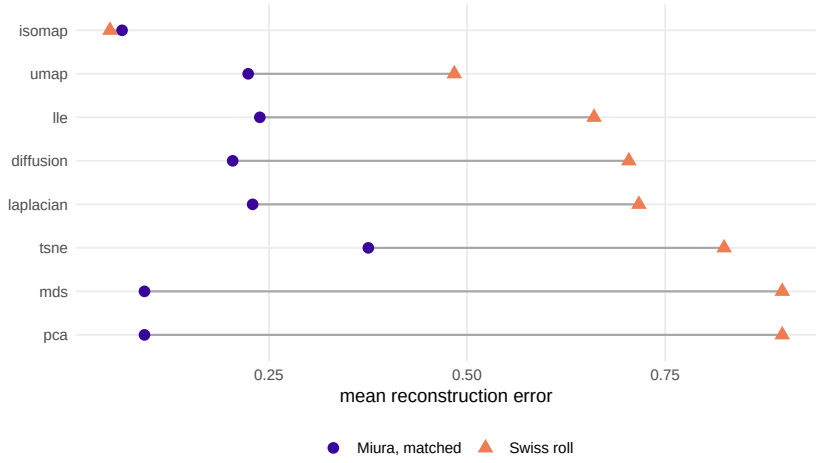


Figure 11.3: Mean reconstruction error of eight methods on a Swiss roll and on a Miura matched to it in gap-to-spacing ratio. Methods are ordered by their Swiss-roll score, and the connecting segments make the reordering visible: the two linear methods move from joint last to near the front.

The two rankings barely correlate: Spearman 0.036 on reconstruction error and -0.060 on Q_{NX}. So the classical benchmark *does* rank methods differently — and the order it produces is the one the field has internalised. Isomap first, the linear methods joint last: the picture the Swiss roll was introduced to make (Tenenbaum et al., 2000), and the one against which the neighbour embeddings have been read ever since (van der Maaten and Hinton, 2008; McInnes et al., 2018).

The reordering is not a shuffle at the margin. PCA and classical MDS score 0.898 on the roll, joint last of eight, and 0.093 on the Miura matched to it, which is second. Isomap barely moves — 0.049 against 0.065 — and neither does its Q_{NX}, 0.860 against 0.864. Every method scores better on the crease side, which

is the first thing to notice: at matched separation the folded sheet is the easier object for all eight. What changes the order is that the linear methods gain far more from the change than the geodesic one does.

11.5.2 Why that answer is the wrong one

Because the match was on one axis and there are two. PCA does well on a folded Miura for a reason that has nothing to do with creases: even at the largest θ this grid uses the sheet is still nearly planar, and a nearly planar object is what PCA is optimal for. Read off this comparison, the Swiss roll separates PCA from Isomap 30 times better than the crease pattern does — a number produced almost entirely by the axis the match failed to control.

11.5.3 The controlled comparison

`e1-controlled.rds` sweeps (n_x, α, θ) against turns, and has an overlap region in which both axes lie inside the joint range of the two families.

```
ctl <- local({
  s <- e1[is.finite(e1$rmse), ]; s$lg <- log(s$gs)
  s[s$lg >= max(tapply(s$lg, s$kind, min)) & s$lg <=
 ↪  min(tapply(s$lg, s$kind, max)) &
    s$np >= max(tapply(s$np, s$kind, min)) & s$np <=
 ↪  min(tapply(s$np, s$kind, max)), ]
})
w <- reshape(ctl[, c("kind", "label", "seed", "method", "rmse")],
             idvar = c("kind", "label", "seed"), timevar =
              ↪  "method",
             direction = "wide")
w$spread <- abs(w$rmse.pca - w$rmse.isomap)
sp    <- tapply(w$spread, w$kind, mean)
ratio <- sp[["classic"]] / sp[["crease"]]

set.seed(11)                          # cluster bootstrap,
 ↪  resampling whole settings
boot <- replicate(2000, {
  r <- vapply(split(w, w$kind), function(d) {
    l <- sample(unique(d$label), length(unique(d$label)), replace
 ↪  = TRUE)
    mean(unlist(lapply(l, function(x) d$spread[d$label == x])))
  }, numeric(1))
  r[["classic"]] / r[["crease"]]
})
ci <- quantile(boot, c(0.025, 0.975))

fit  <- lm(spread ~ I(kind == "classic"), data = w)   #
 ↪  cluster-robust by setting
X <- model.matrix(fit); u <- residuals(fit); bread <-
 ↪  solve(crossprod(X))
meat <- Reduce(`+`, lapply(unique(w$label), function(l) {
  i <- w$label == l
```

```
  tcrossprod(crossprod(X[i, , drop = FALSE], u[i]))
}))
tval <- coef(fit)[2] / sqrt((bread %*% meat %*% bread)[2, 2])
pval <- 2 * pt(-abs(tval), df = length(unique(w$label)) - 2)
mscore <- aggregate(cbind(rmse, qnx) ~ kind + method, ctl, mean)
score  <- function(k, m, col) mscore[[col]][mscore$kind == k &
 ↪  mscore$method == m]
```

9 crease settings and 5 Swiss rolls survive the double restriction. Inside it **the ranking is identical**: Isomap first on both families, PCA and classical MDS behind it on both. Nothing flips.

What differs is the spread. The mean gap between PCA and Isomap is 0.574 on the classical side against 0.113 on the crease side — a ratio of 5.06, with a cluster bootstrap interval over settings of 1.7 to 9.8 and a cluster-robust p of 0.02. The absolute scores go the same way: PCA at 0.922 on a roll against 0.286 on a crease pattern, Isomap at 0.348 against 0.183.

State it in the book's own voice. **Crease patterns are not a harder benchmark; they are an easier one that discriminates less.** A benchmark earns its keep by telling methods apart, and on that criterion the Swiss roll wins by a factor of about five.

What that costs a reader is worth being concrete about. A benchmark whose methods sit 0.113 apart needs more replicates to separate two of them than one whose methods sit 0.574 apart. Chapter 10 turns that into a statistic of its own and reports where along the θ axis this benchmark has any resolving power at all; read next to the ratio above, the two are the same observation from opposite ends. It does not make the crease pattern useless — a benchmark whose answer key is exact can be asked questions a benchmark whose answer key is a convention cannot, and Chapter 9 asks them — but anyone choosing between the two for a straight method comparison should choose the roll, and this chapter is where the book says so.

Figure 11.4 attaches a caveat to that factor of five. The Swiss roll's discriminating power is not monotone in its own difficulty. It reaches 0.909 in a middle window and then collapses to 0.0003 at the tightest rolls, where Isomap short-circuits and fails as completely as PCA does — so past that window the classical benchmark separates nothing either. The settings the head-to-head matched on sit inside the peak. That is the right place to measure a benchmark, on its best window rather than its worst, but the number is a statement about Swiss rolls there and not about Swiss rolls generally.

11.6 Diagnostics

One objection deserves an answer before a reader raises it, and it is specific enough that a footnote would not do.

E1's decisive arm sampled with `boundary = TRUE`, because a two-by-two Miura leaves no room for the interior margin `sample_manifold()` normally draws inside. A Miura's unfolded outline is not convex, so for a pair straddling one

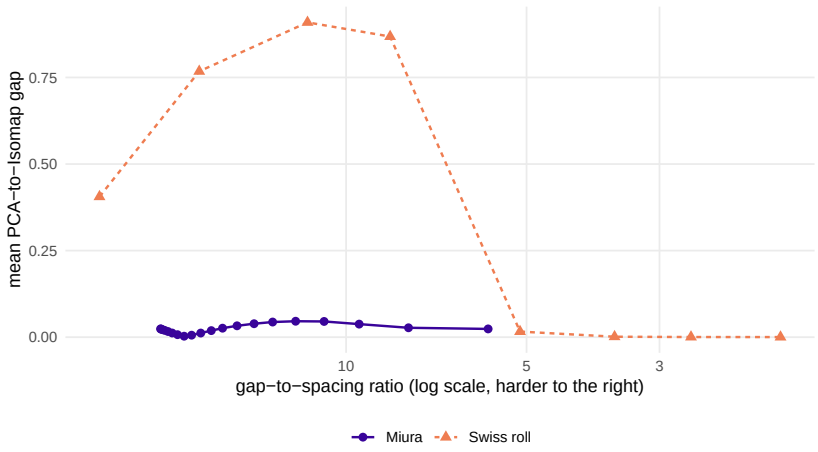


Figure 11.4: How far apart each benchmark holds PCA and Isomap, across its whole sweep and at every seed the grids carry. Difficulty rises to the right. The Miura curve stays near the floor throughout. The Swiss roll separates the two methods widely over a middle window and then stops separating them at all, because past that window Isomap short-circuits and fails as completely as PCA.

of its teeth the straight chart segment leaves the sheet and chart distance is a *lower bound* on the geodesic — Chapter 2 measures this. An answer key that understates long distances penalises the geodesic method specifically, which is exactly the comparison being made.

```
labs <- unique(ctl$label[ctl$kind == "crease"])
exit <- vapply(labs, function(l) {
  q   <- as.numeric(strsplit(sub("^miura", "", l), "/")[[1]])
  pat <- miura_ori(as.integer(q[1]), as.integer(q[1]), alpha =
 ↪  q[2])
  chart_exit_fraction(sample_manifold(pat, theta = q[3], n =
   ↪  600L,
                                      seed = BENCH_SEEDS[1],
                                       ↪  boundary = TRUE), pat)
}, numeric(1))
loo <- vapply(labs, function(l) {
  k <- w[w$kind == "classic" | w$label != l, ]
  m <- tapply(k$spread, k$kind, mean)
  m[["classic"]] / m[["crease"]]
}, numeric(1))
```

Between 0.00% and 3.47% of sampled pairs are affected. Dropping the worst-affected setting moves the ratio from 5.06 to 5.11, and over the 9 leave-one-out refits it stays between 4.41 and 5.80. The approximation biases against the finding rather than for it, and no single setting carries it.

The second question a reader is entitled to ask is how much of the ratio is seed noise. Every cell above is an average over seeds — 5 in the controlled arm, 20 in

the two grids — and the within-family scatter is large: the standard deviation of the PCA-to-Isomap gap across settings and seeds is 0.393 on the classical side against 0.070 on the crease side. That scatter is why the interval quoted above comes from a bootstrap over whole settings rather than over rows, and why the p-value is cluster-robust: seeds within a setting are not independent draws of the quantity being estimated, and treating them as such would report an interval several times too narrow.

11.7 Where it fails

The overlap is a corner. It holds 9 crease settings, 5 rolls, three methods and two metrics. The bootstrap interval — 1.7 to 9.8 — is the honest width of the claim, and it is wide. The head-to-head above is broader in methods and narrower in control. Neither is the study a settled answer would need.

Under the neighbourhood metric the families nearly do collapse. Isomap scores 0.773 on creases against 0.778 on Swiss rolls. The whole difference between the families sits in how badly the linear methods fail, and that is a fact about ambient non-planarity rather than about creases.

The pre-registration got nothing right. `PLAN.md` drafted three outcomes before the run. The families did not collapse; they did not separate in the registered sense, which was *different rankings*, and the ranking is identical; and arm B's null settled neither. The replacement claim is in this book because it fits the evidence, not because its antecedent was met, and recording that is what pre-registration is for.

The Swiss roll is still the right tool for two jobs. It is the clearest picture in the subject and it belongs in a first lecture, and a pipeline that fails on it is broken. What it cannot be is the evidence, because one fixed geometry has nowhere to put a curve.

11.8 Reproduce this

`scripts/run-classic-grid.R` writes `classic-grid.rds`, `scripts/experiment-e1.R` the three E1 artefacts, `scripts/run-benchmark-grid.R` the crease grid; all are committed with provenance. `tests/testthat/test-baselines.R` asserts the arc-length closed form, the Fresnel error, the sphere's stretch factor and the refusal printed above.

11.9 Further reading

The Swiss roll and the S-curve enter with Isomap (Tenenbaum et al., 2000) and LLE (Roweis and Saul, 2000); Lee and Verleysen (Lee and Verleysen, 2007) is the textbook treatment and the place to see how settled their role became. Balasubramanian and Schwartz (Balasubramanian and Schwartz, 2002) is one page long and is the origin of the gap-to-spacing argument this chapter's first axis rests on.

On the answer key, Schoeneman et al. (Schoeneman et al., 2017) is the paper that noticed the Archimedean roll's chart was not isometric and proposed the Euler replacement, and Mahapatra and Chandola (Mahapatra and Chandola, 2018) use it. This chapter's contribution is not the observation but its size.

On what a comparison of this kind can establish, Espadoto et al. (Espadoto et al., 2021) survey the quantitative benchmarks and Machado et al. (Machado et al., 2025) argue that the projection quality metrics they rest on are necessary but not sufficient — a caution that applies with full force to the five-fold separation reported above. The dispute about reading two-dimensional embeddings at all (Chari and Pachter, 2023; Lause et al., 2024) is the reason this book measures rather than argues.

Part IV

Application

Chapter 12

From benchmark to decision

R. Heller · Date: 2026-07-29 · Last changed: 2026-11-09

What happens when the data are real?

Chapter status: planned. The application below has not yet been carried out. No single-cell comparison or improvement over a default method is claimed.

- Use the Chapter 10 grid to select a method and its hyperparameters.
- Apply that choice to a single-cell dataset.
- Compare against the field-default choice.
- Report a null result honestly if the benchmark-guided choice does not win.
- Closes with visualisation and honest-reporting standards: alt text, no colour-only encoding, seed disclosure, aspect-ratio discipline.

Appendix A

Notation

R. Heller · Date: 2026-07-29 · Last changed: 2026-08-30

Symbols

Symbol	Meaning
θ	Folding parameter; the difficulty axis. $\theta = 0$ is the flat sheet
ρ	Dihedral angle at a crease. $\rho = \pi$ when $\theta = 0$; derived from θ, not a second name for it
a, b, α	Miura-ori cell parameters
M, V	Mountain and valley creases
k	Neighbourhood size for kNN-graph methods
D	Ambient dimension after linear lift
$\mathcal{M}$	The folded manifold in $\mathbb{R}^3$
$\mathcal{U}$	The exact unfolding — ground truth in $\mathbb{R}^2$
d_A	Ambient distance, measured in $\mathbb{R}^3$ through the fold
d_U	Intrinsic (path) distance, measured along the surface — equal to Euclidean distance in $\mathcal{U}$

Appendix B

Datasets and codebook

R. Heller · Date: 2026-07-29 · Last changed: 2026-08-31

Generated data

All crease-pattern manifolds are generated, not collected. Every dataset in Parts I–III is reproducible from a pattern specification, a fold angle, a sample size, a noise specification, and a seed.

Precomputed results

`data/processed/benchmark-grid.rds` — the Chapter 10 grid. Provenance: `scripts/run-benchmark-grid.R`.

`data/processed/product-grid.rds` — the product suite behind Chapter 8's irreducible-loss bound, in two arms: a product of two Miura sheets and a product of two Swiss rolls. Provenance: `scripts/run-product-grid.R`.

`data/processed/e1-difficulty.rds`, `e1-controlled.rds`, `e1-armB.rds` — the three arms of E1, which retired the book's original headline claim. Provenance: `scripts/experiment-e1.R`.

`data/processed/part2-sweeps.rds` — the same cell as the main grid, swept over neighbourhood size instead of over noise. This is where Chapters 4, 5 and 6 answer how much of a method's result is the method and how much is the k it was handed. Provenance: `scripts/run-part2-sweeps.R`.

`data/processed/classic-grid.rds` — the Swiss roll, the S-curve and the severed sphere, scored by the same function, the same metrics and the same floor as the crease patterns. Chapter 11 needs it: a comparison in which the two sides are scored by two code paths is not a comparison. Provenance: `scripts/run-classic-grid.R`.

`data/processed/evaluator-audit-pilot.rds` — the narrow pilot that decided whether Chapter 9 had a subject. Provenance: `scripts/run-evaluator-audit.R --pilot`.

`data/processed/seed-budget.rds` — the within-cell spread of every method, and the seed count each one needs before a difference the size of the pre-registered reportable difference (`MIN_REPORTABLE`, fixed in `R/selection.R`) can be detected. This is what standing rule 1's floor is measured against rather than assumed. Provenance: `scripts/measure-seed-budget.R`.

Every one of these carries a provenance block recording the `R/` tree hash, the R version, the BLAS, the package versions, the wall clock and whether the tree was dirty when it ran. `read_run()` refuses one that does not, refuses one written by a `--quick` smoke test, and refuses one whose contents are not the contents the chapter around the call was written against.

External data

One dataset, used only in Chapter 12.

Fresh 68k PBMCs (Donor A) — peripheral blood mononuclear cells from a single healthy donor, from Zheng et al. (Zheng et al., 2017). Obtained from the 10x Genomics dataset repository, Cell Ranger 1.1.0 processing, Chromium Universal 3' chemistry.

`scripts/prepare-single-cell.R` downloads the filtered matrix, applies the preprocessing recorded in that script, and writes the artefact Chapter 12 reads. It is not run in CI and not run at render time, on the same provenance convention as the benchmark grid: run it once, commit the result, keep the script as the record of how.

Licence, and what may be redistributed

The dataset is published under **CC BY 4.0**, which permits redistribution with attribution. So the processed matrix *may* ship in this repository, and the constraint on whether it does is size rather than licence — the full filtered matrix is far too large for git, and Chapter 12 does not need it. What is committed is the reduced matrix the chapter actually reads, produced by the script above, with the reduction stated there and in the chapter.

Label provenance, and why the comparison is not circular

This matters more than it looks, and it was an open question until it was checked. Chapter 12's only external evaluation channel is the cell-type labels. If those labels had themselves been derived by clustering one of the embeddings under test, the comparison would be circular and the chapter would be measuring its own assumptions.

They were not. Zheng et al. generated reference profiles by sequencing ten bead-enriched purified subpopulations from the same donor, separately, and assigned each cell in the 68k population the identity of the purified population with which it had the highest Spearman correlation. The labels come from a separate set of experiments, and no clustering or embedding of the matrix under test enters the assignment.

Two caveats belong in the chapter rather than in a footnote, because both bound what a per-label result can mean.

- The assignment is made in gene-expression space — the same space the embeddings are computed from. It is independent of the *embedding*, which is the circularity that would matter here, but it is not an independent measurement of cell identity, and the chapter should not describe it as ground truth. The book reserves that phrase for the crease patterns, where it is earned.
- The authors flag their own assignments as unreliable in specific places: the gap between the highest and second-highest correlation is small for some cells, cytotoxic T against NK being the example they give, and the purified populations are nested — CD4+ T-helper contains all CD4+ cells — so a hand-written rule reassigns cells to the next-highest correlation within a nested family. Any per-label accuracy in Chapter 12 inherits that, and the pre-registered selection rule of Chapter 10 is what keeps it from being read as a ranking.

Glossary

R. Heller · Date: 2026-07-29 · Last changed: 2026-08-21

Developable surface A surface that can be flattened into a plane without stretching or tearing.

Dihedral angle (ρ) The angle between the two facets meeting at a crease, measured in the ambient space. A crease in an unfolded sheet has $\rho = \pi$, not 0, and ρ decreases as the sheet folds. It is *derived* from θ and is never a synonym for it.

Folding parameter (θ) The parameter of the folding map, and the difficulty axis of the whole book. $\theta = 0$ is the flat sheet, and the pattern folds monotonically as θ grows. It is a parameter, not an angle read off a figure; the map $\rho(\theta)$ from it to the dihedral angle is stated once per pattern in Chapter 2.

Isometry, intrinsic and ambient An *intrinsic* isometry preserves distances measured **along** the surface. An *ambient* isometry preserves straight-line distance in the space the surface sits in. Rigid folding is the first and emphatically not the second: it contracts ambient distance strictly for every $\theta > 0$. Which of the two a method consumes is the distinction the book turns on, and the reason Chapter 9 reports two headline numbers rather than one.

Mountain / valley assignment The direction of each crease. Written M and V; encoded by linetype as well as colour in every figure.

Reach The largest r such that every point within r of the surface has a unique nearest point on it. Most manifold-learning guarantees are stated in terms of a positive reach; a creased surface has reach exactly zero, which is what puts these objects outside the theory rather than merely at its edge.

Rigid origami Folding in which facets stay flat and all deformation happens at creases.

Short-circuit Two points close in ambient space but far apart geodesically. The failure mode that breaks kNN-graph methods.

Colophon

R. Heller · Date: 2026-07-29 · Last changed: 2026-07-29

Written in R Markdown and built with **bookdown** (Xie, 2026). Typeset in Inter and JetBrains Mono. Figures use the viridis plasma palette (option `"C"`), the series convention for this volume.

Interface icons are Font Awesome Free 6.5.1, vendored in `style/webfonts/` rather than loaded from a content delivery network. The icons are licensed CC BY 4.0, the fonts under the SIL OFL 1.1, and the accompanying code under the MIT License; see https://fontawesome.com/license/free.

```
#> - Session info ---------------------------------------------------------------
#>  setting  value
#>  version  R version 4.6.2 (2026-10-29)
#>  os       Ubuntu 24.04.5 LTS
#>  system   x86_64, linux-gnu
#>  ui       X11
#>  language (EN)
#>  collate  C.UTF-8
#>  ctype    C.UTF-8
#>  tz       UTC
#>  date     2026-11-10
#>  pandoc   3.8.3 @ /opt/hostedtoolcache/pandoc/3.8.3/x64/ (via rmarkdown)
#>  quarto   NA
#>
#> ------------------------------------------------------------------------------
```

About the author

R. Heller · Date: 2026-07-29 · Last changed: 2026-07-29

R. Heller is a physician and biomedical researcher working on trauma immunology, laboratory diagnostics, and reproducible analysis workflows. ORCID: 0000-0001-8006-9742.

Citing this book

R. Heller · Date: 2026-08-21 · Last changed: 2026-08-21

This book is published under CC BY 4.0, which asks for attribution and nothing else. Quote it, teach from it, adapt it, use its figures — the licence permits all of that, and a citation is the whole of what is expected in return.

Cite the book itself for the argument and the benchmark design. Cite the repository when what you are relying on is the code: the pattern constructors, the folding maps, the metrics, or a committed artefact under `data/processed/`. They carry the same version number, and the repository is the only one of the two that pins a commit.

> Heller, R. (2026). *Folding in R: Origami, Manifolds, and Dimension Reduction.* Version 0.1.0. https://r-heller.github.io/folding-in-r/

The BibTeX entry is in `citation.bib`, and `CITATION.cff` carries the same metadata in a form GitHub and Zenodo read directly:

```
@book{heller2026folding,
  title     = {Folding in R: Origami, Manifolds, and Dimension
   ↪  Reduction},
  author    = {Heller, R.},
  year      = {2026},
  publisher = {Self-published},
  url       = {https://r-heller.github.io/folding-in-r/},
  note      = {Version 0.1.0}
}
```

Both files are generated from the same source and should not be edited by hand; if a version number here disagrees with one of them, the files are right.

References

R. Heller · Date: 2026-07-29 · Last changed: 2026-07-29

Bibliography

Balasubramanian, M. and Schwartz, E. L. (2002). The isomap algorithm and topological stability. *Science*, 295(5552):7.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Bernstein, M., de Silva, V., Langford, J. C., and Tenenbaum, J. B. (2000). Graph approximations to geodesics on embedded manifolds. Technical report, Department of Psychology, Stanford University, Stanford, CA. Technical report. The convergence result Isomap's consistency rests on: graph shortest paths approximate geodesics when the sample is dense relative to the manifold's minimum radius of curvature and branch separation.

Chari, T. and Pachter, L. (2023). The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8):e1011288.

Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.

Demaine, E. D. and O'Rourke, J. (2007). *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Espadoto, M., Martins, R. M., Kerren, A., Hirata, N. S. T., and Telea, A. C. (2021). Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173.

Federer, H. (1959). Curvature measures. *Transactions of the American Mathematical Society*, 93(3):418–491.

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441.

Hull, T. C. (2020). *Origametry: Mathematical Methods in Paper Folding*. Cambridge University Press.

Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., and Castrén, E. (2003). Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4(1):48.

Kobak, D. and Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nature Communications*, 10:5416.

Kobak, D. and Linderman, G. C. (2021). Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, 39(2):156–157.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Lang, R. J. (2011). *Origami Design Secrets: Mathematical Methods for an Ancient Art*. A K Peters/CRC Press, 2 edition.

Lause, J., Berens, P., and Kobak, D. (2024). The art of seeing the elephant in the room: 2d embeddings of single-cell data do make sense. *PLOS Computational Biology*, 20(10):e1012403.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer New York.

Lee, J. A. and Verleysen, M. (2009). Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7-9):1431–1443.

Machado, A., Behrisch, M., and Telea, A. (2025). Necessary but not sufficient: Limitations of projection quality metrics. *Computer Graphics Forum*, 44(3):e70101.

Mahapatra, S. and Chandola, V. (2018). Learning manifolds from non-stationary streaming data.

McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction.

Miura, K. (1985). Method of packaging and deployment of large membranes in space. *The Institute of Space and Astronautical Science report*, 618:1–9. No DOI was ever assigned. Google Books provides the bibliographic record for ISAS report 618 (1985). The full-text repository is https://jaxa.repo.nii.ac.jp/records/31382; CiNii also indexes it at https://cir.nii.ac.jp/crid/1050003824960950144.

Niyogi, P., Smale, S., and Weinberger, S. (2008). Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1–3):419–441.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Schenk, M. and Guest, S. D. (2013). Geometry of miura-folded metamaterials. *Proceedings of the National Academy of Sciences*, 110(9):3276–3281.

Schoeneman, F., Mahapatra, S., Chandola, V., Napp, N., and Zola, J. (2017). Error metrics for learning reliable manifolds from streaming data. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 750–758. Society for Industrial and Applied Mathematics.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Venna, J. and Kaski, S. (2001). Neighborhood preservation in nonlinear projection methods: An experimental study. In *Artificial Neural Networks — ICANN 2001*, Lecture Notes in Computer Science, pages 485–491. Springer Berlin Heidelberg.

Venna, J., Peltonen, J., Nybo, K., Aidos, H., and Kaski, S. (2010). Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(13):451–490.

Wattenberg, M., Viégas, F., and Johnson, I. (2016). How to use t-SNE effectively. *Distill*.

Xie, Y. (2026). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.47.

Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J., and Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049.