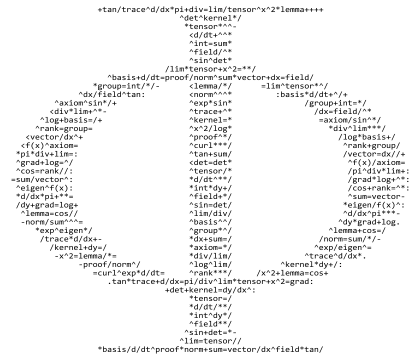
PHI TECHNICAL MONOGRAPHS

Analytic Algorithms and Statistical Computation

Comparative solutions, exact representations and quantified performance



PHI research collective

Abstract

This volume examines finite algorithms, symbolic enumeration, linear algebra, statistical estimation and numerical computation through explicit mathematical representations. The 155 bibliographically identified case studies preserve conventional solutions, structural reductions and the assumptions required for their comparisons. Further chapters develop exact root certificates, continuation, analytic approximation and finite-dimensional recovery.

Arithmetic work, operand growth, memory requirements, conditioning and verification are evaluated separately. Recorded experiments include exact enumeration, shared-factorization leave-one-out estimation, transform computation and source-defined statistical samples. Every runtime ratio belongs to its recorded implementation and workload; algebraic reductions have independently stated cost models.

The nearest-neighbour preservation study on MNIST contains its mathematical specification and an original-data procedure. The original image experiment remains unexecuted in the recorded source. Polynomial-regression and Weekly-data results retain their documented numerical conventions; native R execution is not asserted. Original textbook page reproductions and bulk question statements are absent from this public edition. Bibliographic exercise locators identify the source problems.

Reproduction material. The public source package contains the LaTeX manuscript and authored vector assets required to rebuild this volume. Its executable supplement covers selected numerical studies and certificate calculations. External textbook page reproductions and original externally acquired datasets are not distributed. References identify the scholarly works needed for independent source comparison.

Evidence convention. Established mathematical results retain their scholarly attribution. The exposition states the assumptions of each calculation and comparison. Recorded numerical results retain their original experimental scope. Formal specifications describe prospective proof developments unless an accepted proof object is explicitly identified. This public edition does not establish new proof-assistant kernel acceptance.

Contents

Abstract **i**

1 Dive into Deep Learning **1**

1.1 AC-001 | Derivatives from difference quotients 1
1.2 AC-002 | Sum, product and quotient rules 2
1.3 AC-003 | Constant scaling 3
1.4 AC-004 | The derivative of a variable power 3
1.5 AC-005 | Stationary derivatives and their interpretation 4
1.6 AC-006 | A rational function and its tangent 5
1.7 AC-007 | A separable gradient 6
1.8 AC-008 | Euclidean norm: gradient and failure at the origin 6
1.9 AC-009 | A multivariable chain rule 7
1.10 AC-010 | Derivative of an inverse function 8
1.11 AC-011 | Sine differentiated by automatic differentiation 8
1.12 AC-012 | An explicit expression dependency graph 9
1.13 AC-013 | Derivative labels on the expression graph 10
1.14 AC-014 | Forward and reverse accumulation 11
1.15 AC-015 | When each differentiation mode is preferable 12
1.16 AC-016 | Transposition is an involution 12
1.17 AC-017 | Linearity of transposition 13
1.18 AC-018 | Symmetrization 13
1.19 AC-019 | Tensor norms and library conventions 14
1.20 AC-020 | Optimal association of a three-matrix product 15
1.21 AC-021 | Transpose views and equal arithmetic counts 16
1.22 AC-022 | A stack along the third axis 16
1.23 AC-023 | Variance and concentration of a coin estimate 17
1.24 AC-024 | Shared sample means are dependent 18
1.25 AC-025 | Three-node Markov factorization 19
1.26 AC-026 | Dependent-test Bayes calculation 20
1.27 AC-027 | Mean–variance portfolio algebra 20
1.28 AC-028 | Length of a rank-three array 21
1.29 AC-029 | Length, leading axis and scalar exceptions 22
1.30 AC-030 | Shapes of axis reductions 22

2 Discrete Mathematics: An Open Introduction **23**

2.1 AC-031 | Four-colour squares and five-colour dominoes . . . 23
2.2 AC-032 | Two-colour squares and three-colour dominoes . . . 24
2.3 AC-033 | A repeated unit root . . . 24
2.4 AC-034 | A repeated root at two . . . 25
2.5 AC-035 | Seven elementary generating functions . . . 26
2.6 AC-036 | Sequences from five rational functions . . . 27
2.7 AC-037 | Triangular numbers by three constructions . . . 28
2.8 AC-038 | A triangular perturbation of a constant sequence . . . 28
2.9 AC-039 | An inhomogeneous numerator from initial data . . . 29
2.10 AC-040 | The Fibonacci generating function . . . 30
2.11 AC-041 | Partial sums of Fibonacci numbers . . . 30
2.12 AC-042 | A sum of two geometric modes . . . 31
2.13 AC-043 | A shifted geometric coefficient . . . 31
2.14 AC-044 | A seventh coefficient of a product . . . 32
2.15 AC-045 | Coefficients of a double geometric pole . . . 33
2.16 AC-046 | Four transformations of the counting sequence . . . 33
2.17 AC-047 | Four transformations of shifted Fibonacci numbers . . . 34
2.18 AC-048 | Alternating powers of two . . . 34
2.19 AC-049 | Two initial constants followed by a counting sequence . . . 35
2.20 AC-050 | Reconstructing a sequence from its differences . . . 35

3 Introduction to Algorithms **37**

3.1 AC-051 | Fibonacci numbers: generating function, rounding and . . . 37
3.2 AC-052 | Binary-tree enumeration and the Catalan asymptotic . . . 38
3.3 AC-053 | An explicit cubic polynomial product . . . 39
3.4 AC-054 | Synthetic division and Horner evaluation . . . 40
3.5 AC-055 | Reversing a polynomial in point-value form . . . 41
3.6 AC-056 | The number of values needed for interpolation . . . 41
3.7 AC-057 | Quadratic-time coefficient interpolation . . . 42
3.8 AC-058 | Why pointwise ratios need not perform polynomial . . . 42
3.9 AC-059 | Cartesian sums through coefficient convolution . . . 43
3.10 AC-060 | A four-point DFT in the textbook sign convention . . . 44
3.11 AC-061 | Inverse FFT pseudocode and correctness . . . 45
3.12 AC-062 | Radix-three Fourier recursion . . . 45
3.13 AC-063 | The chirp transform including the zero parameter . . . 46
3.14 AC-064 | All derivatives at one point by one convolution . . . 47
3.15 AC-065 | Constructing a polynomial from its roots . . . 47
3.16 AC-066 | Multipoint evaluation by a remainder tree . . . 48
3.17 AC-067 | An invertible Fourier transform over a composite ring . . . 49
3.18 AC-068 | Prime-modulus Fourier transforms . . . 50
3.19 AC-069 | The Vandermonde determinant . . . 50
3.20 AC-070 | Newton iteration: corrected local error theorem . . . 51

3.21 AC-071 | Horner evaluation with a loop invariant 53
3.22 AC-072 | Optimal bitonic Euclidean tours 54
3.23 AC-073 | Integer linear programming and duality gaps 55

4 Database System Concepts **56**
4.1 AC-074 | Counting all oriented labelled join orders 56

5 Natural Language Processing with Python **57**
5.1 AC-075 | Recursive and dynamic Catalan computation 57

6 Basics of Compiler Design **59**
6.1 AC-076 | Decimal languages with leading zeros 59
6.2 AC-077 | NFA and subset DFA for a starred prefix 60
6.3 AC-078 | A block-language NFA and DFA 61
6.4 AC-079 | Counting and modular-counting automata 62
6.5 AC-080 | Minimal binary divisibility automata 63
6.6 AC-081 | The empty regular language and dead states 64
6.7 AC-082 | Four closure constructions for regular languages 65
6.8 AC-083 | Three false closure claims 66

7 Introduction to Information Retrieval **67**
7.1 AC-084 | Transition matrix of the three-state figure 67
7.2 AC-085 | Teleportation matrices at three probabilities 68
7.3 AC-086 | A Markov state for browser backtracking 68
7.4 AC-087 | Ergodicity of a parameterized three-state chain 69
7.5 AC-088 | Full-support teleportation and ergodicity 70
7.6 AC-089 | Uniform lower bounds and the collapse of PageRank 71
7.7 AC-090 | Reproducing the seven-page PageRank example 71
7.8 AC-091 | Sequential-disk PageRank 72
7.9 AC-092 | The support of topic-specific PageRank 73
7.10 AC-093 | A topic need not reach the entire web 74
7.11 AC-094 | A counterexample to pagewise topical monotonicity 74
7.12 AC-095 | The personalized teleportation chain 75
7.13 AC-096 | Ergodicity on the reachable topic support 76
7.14 AC-097 | Coordinatewise mixture identity 76
7.15 AC-098 | The first HITS update depends on update order 77
7.16 AC-099 | Rank from a linear dependence 78
7.17 AC-100 | An eigenvalue and its full eigenspace 78
7.18 AC-101 | A normalized symmetric eigendecomposition 78
7.19 AC-102 | Term co-occurrence from an incidence matrix 79
7.20 AC-103 | Exact verification of the printed thin SVD 80
7.21 AC-104 | Document co-occurrence from a binary incidence matrix 80
7.22 AC-105 | Term-frequency inner products 81
7.23 AC-106 | The optimal rank-one approximation and its exact error 81

7.24 AC-107 | A rank-one SVD term is an outer product 82
7.25 AC-108 | Full and reduced singular-value arrays 83

8 Foundations of Machine Learning **84**
8.1 AC-109 | PCA and the Rayleigh quotient 84
8.2 AC-110 | Nyström approximation as orthogonal projection 85
8.3 AC-111 | Positive rescaling of a kernel 86
8.4 AC-112 | Eleven positive kernels: explicit feature and integral 86
8.5 AC-113 | Two-step graph walks and Gram factorization 89
8.6 AC-114 | Symmetric-difference kernel 89
8.7 AC-115 | Set-sum kernel 90
8.8 AC-116 | Two conditionally negative kernels 91
8.9 AC-117 | A two-point obstruction to positive definiteness 91
8.10 AC-118 | Fractional distance: non-PDS and NDS 92
8.11 AC-119 | Negative definiteness on a Hilbert space 93
8.12 AC-120 | All least-squares solutions and the nullspace 93
8.13 AC-121 | Kernel perturbation and the resolvent identity 94
8.14 AC-122 | Unregularized regression has no uniform stability bound 95
8.15 AC-123 | Exact leave-one-out kernel ridge regression 95
8.16 AC-124 | Birthday collisions with one favoured floor 97
8.17 AC-125 | Label-bias estimation and an exponential certificate 98
8.18 AC-126 | Optimal testing of two biased coins 99
8.19 AC-127 | Optimizing a tail-to-moment bound 100
8.20 AC-128 | Hoeffding versus Chebyshev with the missing indepen- 100
8.21 AC-129 | Random projections, PCA and MNIST nearest-neighbour overlap . . . 101

9 The Elements of Statistical Learning **104**
9.1 AC-130 | Population structure and the complete 200-observation experiment . . . 104
9.2 AC-131 | Kernel-PCA coordinates and out-of-sample centring 105
9.3 AC-132 | Variational proof of kernel principal functions 106
9.4 AC-133 | Reproducing identities and the finite representer theorem 107
9.5 AC-134 | Centring with an unpenalized intercept 108
9.6 AC-135 | Gaussian posterior and ridge regression 109
9.7 AC-136 | Ridge as augmented least squares 110
9.8 AC-137 | Orthogonal subset, ridge and lasso estimators 111
9.9 AC-138 | Norm monotonicity: ridge, lasso and PLS 111
9.10 AC-139 | Duplicate predictors and the effective ridge penalty 112
9.11 AC-140 | Feature-space and kernel-space ridge equivalence 113
9.12 AC-141 | The large-penalty ridge direction 114
9.13 AC-142 | High-dimensional ridge through the observed row space 115
9.14 AC-143 | Exact reduction for arbitrary loss and quadratic regu- 115
9.15 AC-144 | Zero-residual fits and the minimum-norm interpolant 116
9.16 AC-145 | The ridged discriminant limit and maximal data piling 116

9.17 AC-146 | One exact row-space reduction across cross-validation 117
9.18 AC-147 | Kernel logistic regression from an empirical square-root 118
9.19 AC-148 | Leave-one-out local polynomial regression 119
9.20 AC-149 | When the PRESS identity holds for a linear smoother 119

10 An Introduction to Statistical Learning: with Applications in Python 121
10.1 AC-150 | Polynomial regression, exact PRESS and the Python-seeded sample . . 121
10.2 AC-151 | Complete Weekly logistic leave-one-out calculation: Python edition . . . 123

11 An Introduction to Statistical Learning: with Applications in R 126
11.1 AC-152 | Polynomial regression, exact PRESS and the R-seeded sample 126
11.2 AC-153 | Complete Weekly logistic leave-one-out calculation: R edition 128

12 Introduction to Natural Language Processing 130
12.1 AC-154 | An exact two-ReLU XOR network 130
12.2 AC-155 | A complete certificate for a dead ReLU on the Boolean 131

13 Certified roots, continuation and partial computation 132
13.1 CSE-A: roots as typed algebraic data 132
13.2 CSE-B: an exact root-count checker for rational contours 133
13.3 CSE-C: continuation with topology and conditioning in one record 136
13.4 CSE-D: finite passports, Galois closure and state-space cost 137
13.5 CSE-E: invariant outputs can reduce computation without losing the question . . 138
13.6 CSE-F: composition of partial programs 139

14 Finite certificates and exact representations 140
14.1 CU1: a square-root API needs a branch contract 140
14.2 CU2: a three-entry representation can retain a repeated root 140
14.3 CU3: permutation data have checkable invariants 141
14.4 CU4: a root count carried by four exact inequalities 141
14.5 CU5: a covering lift becomes a finite automaton 142
14.6 CU6: group enumeration and its cost 143
14.7 CU7: guard a partial composition, then propagate its error 144
14.8 Submission and machine-checking boundary 144
14.9 Undergraduate comparison laboratory: exact input, exact claim 144

15 Algorithmic geometry and continuation 148
15.1 CG1: Bernstein coefficients are a proof representation 148
15.2 CG2: subdivision and a sound checker 149
15.3 CG3: separate root counting from branch following 149
15.4 CG4: multiplication matrices avoid unstable root labels 150
15.5 CG5: a regular closure is an automaton with algebraic invariants 151
15.6 CG6: truncation error in a Hardy-space computation 151
15.7 CG7: error balls must fit the next domain 152

15.8 Deliverable and verification standard . . . 152
15.9 Graduate comparison: algorithmic work and certificate strength . . . 152

16 Uniform certificate construction and complexity 156
16.1 CP1: a quantitative termination condition for one polynomial . . . 156
16.2 CP2: a nonzero family can defeat an axis-only witness . . . 156
16.3 CP3: joint Bernstein certificates for parameter boxes . . . 157
16.4 CP4: separating arithmetic, topology and implementation soundness . . . 158
16.5 CP5: a condition-aware complexity account . . . 159
16.6 CP6: labelled continuation and label-free observables . . . 159
16.7 CP7: a research specification with auditable novelty . . . 159
16.8 Research extension: completeness, lower bounds and the audit itself . . . 160

17 Analytic approximation and geometric computation 164
17.1 C01. Finite jets as a certified data structure . . . 164
17.2 C02. Exact minimax complexity of linear restriction . . . 165
17.3 C03. Coefficient extraction and optimal sampling radius . . . 166
17.4 C04. Finite sampling, regularity witnesses, and certificates . . . 167
17.5 C05. Certified conformal charts in computational geometry . . . 167
17.6 C06. Inverse formulas and projective numerical contracts . . . 169
17.7 C07. Continuation, identifiability, and attainable guarantees . . . 169

18 Coefficient and chart calculations 172
18.1 CU01. Horner evaluation with a remainder certificate . . . 172
18.2 CU02. Weighted coefficient tables . . . 173
18.3 CU03. Root-of-unity averaging and aliasing . . . 173
18.4 CU04. A test suite can miss an entire polynomial . . . 174
18.5 CU05. A global map invariant from one factorization . . . 175
18.6 CU06. Projective representations in a numerical interface . . . 176
18.7 CU07. Cancellation-free inverse evaluation . . . 177
18.8 CU08. Cost and verification as separate outputs . . . 177

19 Approximation operators in computation 179
19.1 CG01. Optimality over arbitrary linear measurements . . . 179
19.2 CG02. Strict convexity of the sampling-radius objective . . . 180
19.3 CG03. Derivative reconstruction from a larger-domain certificate . . . 181
19.4 CG04. Exact sample certificates with uncertain data . . . 181
19.5 CG05. Composition of approximate conformal charts . . . 183
19.6 CG06. Restriction in different output norms . . . 183
19.7 CG07. Analytic continuation as an inverse problem . . . 184
19.8 CG08. Formal contracts for numerical certificates . . . 185

20 Optimal recovery and numerical specifications **187**
20.1 CP01. The competitor class in an optimality theorem 187
20.2 CP02. A rational bracket for the optimal sample radius 188
20.3 CP03. Information-theoretic impossibility from a mode pair 188
20.4 CP04. Complexity of exact sign certificates 189
20.5 CP05. Domain guards for analytic geometry code 190
20.6 CP06. Semantic ownership of a proof corpus 191
20.7 CP07. Mutation testing of theorem interfaces 192
20.8 CP08. A layered trust record for machine-assisted mathematics 193

References **195**

Chapter 1

Dive into Deep Learning

Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. Reference problems and derived analyses.

1.1 AC-001 | Derivatives from difference quotients

Reference: [1], exercise 1.

Source task and conventions

Prove the derivatives of a constant, a power, the exponential and the logarithm directly from the defining limits.

Conventional solution and proof

The derivative at x is $\lim_{h\to 0}[f(x+h)-f(x)]/h$, when this finite limit exists. For $f(x)=c$ the numerator is zero, so $f'(x)=0$. For an integer $n\geq 1$, the finite binomial identity gives

$$\frac{(x+h)^n-x^n}{h}=nx^{n-1}+\sum_{j=2}^{n}\binom{n}{j}x^{n-j}h^{j-1}.$$

The sum tends to zero; hence $(x^n)'=nx^{n-1}$. The case $n=0$ is constant. Define $e^x=\sum_{k=0}^{\infty}x^k/k!$. Absolute convergence permits the Cauchy product, and the binomial theorem gives $e^{x+h}=e^xe^h$. For $|h|\leq 1$,

$$\left|\frac{e^h-1}{h}-1\right|\leq |h|\sum_{k=2}^{\infty}\frac{1}{k!}\longrightarrow 0.$$

It follows that $(e^x)'=e^x$. The positive, continuous, strictly increasing exponential has a continuous inverse $\log:(0,\infty)\to\mathbb{R}$. Put $u=\log(x+h)-\log x$; then $u\to 0$ and $h=x(e^u-1)$. Consequently

$$\frac{\log(x+h)-\log x}{h}=\frac{1}{x}\frac{u}{e^u-1}\longrightarrow\frac{1}{x}.$$

For an integer $n=-m<0$, $x\neq 0$ and the algebraic identity

$$\frac{(x+h)^{-m}-x^{-m}}{h}=-\frac{(x+h)^m-x^m}{h\,x^m(x+h)^m}$$

gives nx^{n-1}. For a real exponent a on $x>0$, write $x^a=e^{a\log x}$. In the difference quotient set $u=a\log(1+h/x)$. The limits $(e^u-1)/u\to 1$ and $\log(1+t)/t\to 1$ give $(x^a)'=ax^{a-1}$. This also specifies the necessary domains.

Structural or analytic derivation

The coefficient of h in $f(x+h)$ is the derivative whenever the remaining terms are $o(h)$. The preceding calculations establish the expansions

$$c=c+0h,\quad (x+h)^n=x^n+nx^{n-1}h+O(h^2),$$

$$e^{x+h}=e^x+e^xh+O(h^2),\qquad \log(x+h)=\log x+x^{-1}h+o(h).$$

Thus coefficient extraction gives exactly the same four results. For complex x, the power and exponential remain analytic on their natural domains; a logarithm requires a chosen local branch avoiding zero. This formulation is a proved extension of the real calculation, not an additional assumption about a real-valued loss.

Comparative cost and preferred method

For a single symbolic derivative the difference quotient and the local coefficient method have comparable fixed proof cost. No measured speed advantage is claimed. Once the four local rules are available, evaluation uses one power or exponential or one reciprocal, independently of a finite-difference step. A numerical difference quotient uses two function evaluations and division and supplies only an approximation; it does not meet the source's exact-proof requirement. The preferred solution here is the defining-limit proof.

1.2 AC-002 | Sum, product and quotient rules

Reference: [1], exercise 2.

Source task and conventions

Derive the sum, product and quotient differentiation rules from first principles.

Conventional solution and proof

Suppose f,g are differentiable at x. Their increments satisfy $f(x+h)=f(x)+f'(x)h+r_f(h)$ and $g(x+h)=g(x)+g'(x)h+r_g(h)$, where $r_f(h)/h, r_g(h)/h\to 0$. Addition gives $(f+g)'(x)=f'(x)+g'(x)$. For the product, use the exact identity

$$f(x+h)g(x+h)-f(x)g(x)=[f(x+h)-f(x)]g(x+h)+f(x)[g(x+h)-g(x)].$$

Divide by h. Differentiability implies $g(x+h)\to g(x)$, so the limit is $f'(x)g(x)+f(x)g'(x)$. If $g(x)\neq 0$, continuity supplies a neighbourhood in which g is nonzero, and

$$\frac{1}{h}\left(\frac{f(x+h)}{g(x+h)}-\frac{f(x)}{g(x)}\right)=\frac{g(x)\frac{f(x+h)-f(x)}{h}-f(x)\frac{g(x+h)-g(x)}{h}}{g(x+h)g(x)}.$$

The denominator tends to $g(x)^2\neq 0$, proving $(f/g)'=(f'g-fg')/g^2$.

Structural or analytic derivation

A first-order jet is the pair $(u,\dot u)$ represented by $u+\varepsilon\dot u$ with $\varepsilon^2=0$. Multiplication in this two-dimensional algebra gives

$$(u+\varepsilon\dot u)(v+\varepsilon\dot v)=uv+\varepsilon(\dot uv+u\dot v).$$

For $v \neq 0$, $(v+\varepsilon\dot{v})^{-1} = v^{-1} - \varepsilon\dot{v}/v^2$, as direct multiplication verifies. Addition, multiplication and this inverse therefore reproduce all three rules. The limit argument above proves that these formal coefficients coincide with actual derivatives.

Comparative cost and preferred method

For a straight-line calculation with L elementary operations, first-order jet evaluation performs a constant number of operations for each original operation, hence $O(L)$ arithmetic and the same dependency ordering. Repeated finite-difference calls do not prove any rule. For the requested theorem, the conventional limit proof is already efficient; the jet algebra becomes useful when the rules are reused in automatic differentiation.

1.3 AC-003 | Constant scaling

Reference: [1], exercise 3.

Source task and conventions

Deduce the constant multiple rule as a special case of the product rule.

Conventional solution and proof

Take the constant function $g(x) = c$. Its derivative is zero by the defining difference quotient. Therefore $(cf)' = g'f + gf' = 0f + cf' = cf'$. This argument also covers $c = 0$.

Structural or analytic derivation

Multiplication of a first-order jet by c sends (f, f') to (cf, cf'). This is the linearity of scalar multiplication in the jet algebra, and it gives the same identity without additional differentiation rules.

Comparative cost and preferred method

Both proofs require one application of an already proved linear or product identity. The numerical derivative evaluation requires one multiplication by c. No asymptotic improvement is available in this fixed scalar task.

1.4 AC-004 | The derivative of a variable power

Reference: [1], exercise 4.

Source task and conventions

Differentiate $f(x) = x^x$ on its real differentiable domain $x > 0$.

Conventional solution and proof

Taking logarithms gives $\log f(x) = x \log x$. Differentiating yields $f'/f = \log x + 1$, and hence

$$\boxed{f'(x) = x^x(1 + \log x), \quad x > 0.}$$

The steps are valid because $f(x) > 0$ and log is differentiable on $(0, \infty)$.

Structural or analytic derivation

For a local complex logarithm L with $L(x) = \log x$, set $F(z) = \exp(zL(z))$. Since $L'(z) = 1/z$, the chain rule gives $F' = F(L + 1)$. Restricting to positive real z recovers the displayed result. This extension also supplies analytic derivatives of every order in a disk avoiding the logarithm cut.

Comparative cost and preferred method

Logarithmic differentiation evaluates the result with one logarithm, one exponential/power and elementary arithmetic. Expanding a difference quotient is more cumbersome but is not the strongest conventional baseline. The conventional logarithmic method and the analytic continuation have the same formula and arithmetic cost. The real logarithmic solution is preferable for this question.

Source qualification

Nonpositive real arguments do not provide a single open real domain on which the usual real-valued x^x is defined. No extension through such arguments is assumed.

1.5 AC-005 | Stationary derivatives and their interpretation

Reference: [1], exercise 5.

Source task and conventions

Explain $f'(x_0) = 0$ and give a function and point with this property.

Conventional solution and proof

The defining statement is

$$f(x_0 + h) - f(x_0) = o(h).$$

Thus the first-order change vanishes and the tangent line is horizontal. For $f(x) = x^2$ at $x_0 = 0$, the quotient $h^2/h = h \to 0$, and the point is a minimum. For $g(x) = x^3$ at zero, the quotient is $h^2 \to 0$, but g changes sign: a zero derivative alone does not imply an extremum. The function $-x^2$ supplies a maximum.

Structural or analytic derivation

A local power series $f(x_0 + h) = a_0 + a_1 h + a_2 h^2 + \cdots$ has $a_1 = 0$ at a stationary point. The first nonzero coefficient after a_0 determines the local behaviour: an even power gives a local extremum according to its sign, and an odd power changes sign. This conclusion follows by factoring $h^k(a_k + o(1))$, whose second factor has the sign of a_k for sufficiently small nonzero h.

Comparative cost and preferred method

The requested example and interpretation have fixed symbolic cost. The coefficient formulation additionally classifies analytic stationary points but supplies no computational acceleration for the single example. The conventional defining-limit answer is sufficient.

1.6 AC-006 | A rational function and its tangent

Reference: [1], exercise 6.

Source task and conventions

Plot $f(x) = x^3 - x^{-1}$ and its tangent at $x = 1$.

Conventional solution and proof

The domain is $\mathbb{R} \setminus \{0\}$. Direct differentiation gives $f'(x) = 3x^2 + x^{-2}$. Since $f(1) = 0$ and $f'(1) = 4$, the tangent is

$$\boxed{y = 4(x - 1).}$$

The exact difference factors as

$$f(x) - 4(x - 1) = \frac{(x - 1)^2(x^2 + 2x - 1)}{x}.$$

It is $O((x - 1)^2)$ near one, proving tangency independently of the derivative calculation. The graph uses separate intervals on either side of zero; the vertical asymptote is not joined by a plotted line.

Structural or analytic derivation

The Laurent representation $z^3 - z^{-1}$ makes the excluded point and the derivative $3z^2 + z^{-2}$ explicit. A Taylor expansion about one has first-order coefficient four and the exact quadratic remainder displayed above. This analytic description certifies both the tangent and the singularity shown in the figure.

Comparative cost and preferred method

Both symbolic methods use a fixed number of operations. Drawing m graph samples costs $\Theta(m)$ under either representation. The Laurent formulation improves the singularity check, while direct differentiation is the simplest conventional solution.

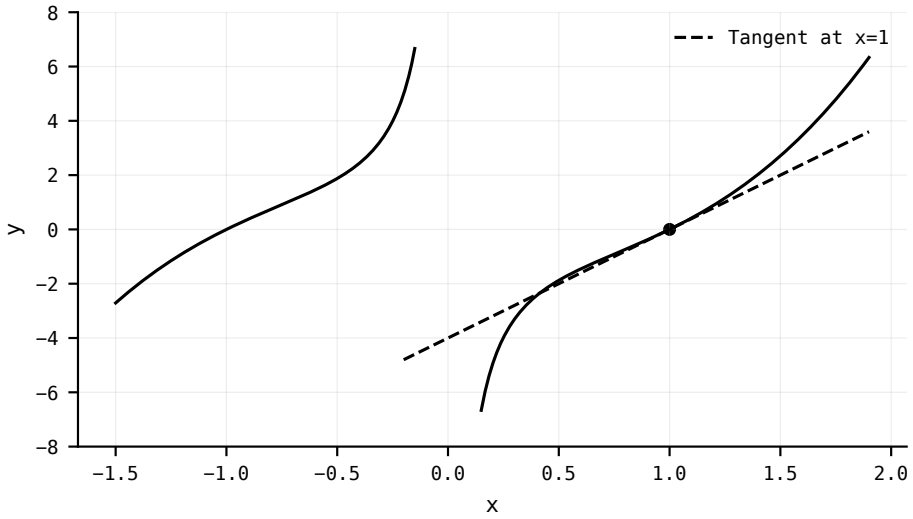


Figure 1.1: Both branches of x^3-1/x and the exact tangent 4(x-1); the plot is deliberately disconnected at the pole.

1.7 AC-007 | A separable gradient

Reference: [1], exercise 7.

Source task and conventions

Find ∇f for $f(x_1, x_2) = 3x_1^2 + 5e^{x_2}$.

Conventional solution and proof

Holding the other coordinate fixed gives

$$\boxed{\nabla f(x_1, x_2) = (6x_1, 5e^{x_2})^{\mathsf{T}}.}$$

For an increment $h = (h_1, h_2)$, the residual after subtracting the linear term is $3h_1^2+5e^{x_2}(e^{h_2}-1-h_2) = O(\|h\|^2)$. Thus this vector is the total derivative, not merely a list of formal partial derivatives.

Structural or analytic derivation

The entire extension $F(z_1, z_2) = 3z_1^2 + 5e^{z_2}$ separates into two coordinate functions. Its first-order jet is $6z_1h_1 + 5e^{z_2}h_2$. Coefficient extraction or forward automatic differentiation therefore yields the same gradient without mixed terms.

Comparative cost and preferred method

The direct formula costs one exponential and two scalar multiplications. Automatic differentiation has constant overhead on the same expression graph. Separate finite differences use at least three function evaluations and introduce truncation error. Direct symbolic differentiation is the preferable method for this two-term expression.

1.8 AC-008 | Euclidean norm: gradient and failure at the origin

Reference: [1], exercise 8.

Source task and conventions

Differentiate $f(x) = \|x\|_2$ and determine what happens at $x = 0$.

Conventional solution and proof

For $x \neq 0$, the chain rule applied to $(x^{\mathsf{T}}x)^{1/2}$ gives

$$\boxed{\nabla f(x) = x/\|x\|_2.}$$

The denominator is positive, so the expression is well defined. At zero, along any unit vector v, $f(tv) = |t|$ has unequal one-sided derivatives. Therefore no derivative exists at zero. Equivalently, a putative linear differential L would satisfy $L(v) = L(-v) = 1$, contradicting linearity.

Structural or analytic derivation

The subdifferential at zero is exactly the closed unit ball. Indeed, g is a subgradient if $\|y\|_2 \geq g^{\mathsf{T}}y$ for every y. Cauchy–Schwarz proves this for $\|g\|_2 \leq 1$; choosing $y = g$ proves the converse. At nonzero x, equality in Cauchy–Schwarz makes the subgradient unique and equal to $x/\|x\|_2$. A norm contains conjugation in complex coordinates and is not holomorphic.

Comparative cost and preferred method

Computing a dense n-coordinate norm gradient costs $\Theta(n)$: the formula reads every coordinate and writes n outputs, giving a matching input/output lower bound. Coordinatewise numerical differentiation requires $\Theta(n^2)$ work if each function call scans the full vector. Against a competent analytic or automatic-differentiation baseline there is no asymptotic improvement. Complex-step differentiation of the norm is invalid.

1.9 AC-009 | A multivariable chain rule

Reference: [1], exercise 9.

Source task and conventions

For $u = f(x, y, z)$ and $x = x(a, b)$, $y = y(a, b)$, $z = z(a, b)$, give the derivatives of u with respect to a, b.

Conventional solution and proof

Assume $g = (x, y, z) : \mathbb{R}^2 \to \mathbb{R}^3$ and $f : \mathbb{R}^3 \to \mathbb{R}$ are differentiable at the relevant points. Write $g(t + h) = g(t) + J_g h + o(\|h\|)$ and $f(g(t) + k) = f(g(t)) + \nabla f^{\mathsf{T}}k + o(\|k\|)$. Since $k = O(\|h\|)$, substitution proves $D(f \circ g) = Df\, Dg$. Hence

$$u_a = f_x x_a + f_y y_a + f_z z_a, \qquad u_b = f_x x_b + f_y y_b + f_z z_b.$$

Every partial derivative of f is evaluated at $(x(a, b), y(a, b), z(a, b))$.

Structural or analytic derivation

The reverse accumulation is $\nabla_{(a,b)} u = J_g^{\mathsf{T}} \nabla f$. It propagates the three output sensitivities through one matrix–vector product. The displayed differential proof establishes the correctness of this accumulation rule. Complex variables require the corresponding real differential or a genuinely holomorphic map.

Comparative cost and preferred method

Once the five derivative objects are supplied, the matrix–vector product uses six multiplications and four additions. Computing the two displayed scalar sums has exactly the same count. Reverse-mode notation organizes the computation; it does not reduce these counts for the 3×2 example.

1.10 AC-010 | Derivative of an inverse function

Reference: [1], exercise 10.

Source task and conventions

Find the derivative of a locally invertible differentiable function's inverse, specifying the necessary nondegeneracy condition.

Conventional solution and proof

Let f be continuous and one-to-one on an interval, differentiable at x_0, with $f'(x_0) \neq 0$. Its inverse g is continuous at $y_0 = f(x_0)$. For $y \neq y_0$, put $x = g(y)$; then $x \to x_0$ and

$$\frac{g(y) - g(y_0)}{y - y_0} = \left(\frac{f(x) - f(x_0)}{x - x_0} \right)^{-1} \longrightarrow \frac{1}{f'(x_0)}.$$

Therefore $g'(y) = 1/f'(g(y))$ wherever these hypotheses hold. Differentiating $f(g(y)) = y$ gives the same identity after existence of g' has been established.

Structural or analytic derivation

For a holomorphic F with $F'(z_0) \neq 0$, the complex inverse function theorem gives a local holomorphic inverse and the same formula. The sensitivity of the inverse is $|G'(F(z_0))| = 1/|F'(z_0)|$. If the derivative approaches zero, inversion becomes ill-conditioned. The real bijection $f(x) = x^3$ is a counterexample to differentiability of the inverse at zero.

Comparative cost and preferred method

Evaluating the derivative after the inverse point is known takes one derivative evaluation and one reciprocal. Numerically perturbing y and resolving the inverse twice adds two nonlinear solves. The direct inverse-derivative formula is conventional and already optimal at this level; the complex formulation adds branch and sensitivity information.

Source qualification

Invertibility alone does not imply a differentiable inverse. The nonzero-derivative hypothesis is added explicitly and the degenerate case is exhibited.

1.11 AC-011 | Sine differentiated by automatic differentiation

Reference: [1], exercise 4.

Source task and conventions

Plot $f(x) = \sin x$ and a derivative obtained by automatic differentiation, without using an explicitly coded cosine as the derivative.

Conventional solution and proof

The executable solution makes x a differentiable variable, evaluates the primitive sine, sums the outputs, and requests its gradient. Since every component depends only on its corresponding input, differentiating the sum returns the vector of the individual derivatives. The resulting curve is plotted together with the original sine. The evaluation procedure does not insert an explicit cosine formula.

```
x = torch.linspace(-pi, pi, 801, dtype=torch.float64,
                   requires_grad=True)
y = torch.sin(x)
g, = torch.autograd.grad(y.sum(), x)
```

The automatic-differentiation primitive implements the local derivative rule. Composition of these local rules is justified by the chain-rule proof in the preceding questions.

Structural or analytic derivation

An independent analytic check is available from $\sin z = (e^{iz} - e^{-iz})/(2i)$: its derivative is $(e^{iz} + e^{-iz})/2$. This identity is used only to check the automatic output. A complex-step diagnostic computes $\operatorname{Im} \sin(x + ih)/h$; its Taylor expansion differs from the exact derivative by $O(h^2)$, so it is not an exact replacement for the requested automatic differentiation.

Comparative cost and preferred method

For m independent points, both automatic evaluation and direct primitive differentiation cost $\Theta(m)$ and produce m values. The recorded error is a numerical diagnostic, not a proof of the library implementation. Complex-step uses one complex evaluation per point and has a nonzero truncation term. Automatic differentiation satisfies the source requirement and is retained as the principal method.

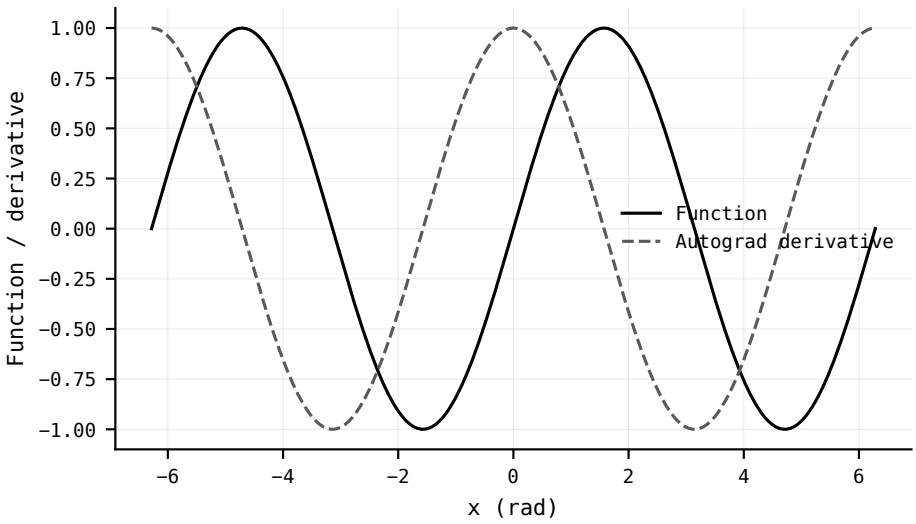


Figure 1.2: The derivative curve comes from the executed automatic-differentiation output array, not a separately inserted cosine curve.

1.12 AC-012 | An explicit expression dependency graph

Reference: [1], exercise 5.

Source task and conventions

Construct the dependency graph of $f(x) = \log(x^2)\sin x + x^{-1}$, for real $x \neq 0$.

Conventional solution and proof

Use the six intermediate quantities

$$v_1 = x^2, \quad v_2 = \log v_1, \quad v_3 = \sin x, \quad v_4 = v_2 v_3, \quad v_5 = x^{-1}, \quad v_6 = v_4 + v_5 = f.$$

The directed dependencies are $x \to v_1 \to v_2 \to v_4 \to v_6$, $x \to v_3 \to v_4$, and $x \to v_5 \to v_6$. Since $x \neq 0$, $v_1 > 0$ and all six operations are defined. The accompanying graph includes every edge and every intermediate quantity.

Structural or analytic derivation

This graph is a straight-line program. It also defines a first-order jet program by replacing each scalar by a value–tangent pair. The number and direction of dependencies are unchanged. Locally near any nonzero real point, $\log(z^2)$ has a consistent analytic branch; global branch choices cannot be omitted for complex inputs.

Comparative cost and preferred method

The shared expression uses six elementary nodes. Recomputing x^2, $\log(x^2)$ or $\sin x$ for each requested derivative can increase work; storing the node values avoids it. A competent conventional expression graph already performs this sharing, so the proposed graph has no separate asymptotic advantage. Its benefit is an explicit, checkable dependency contract.

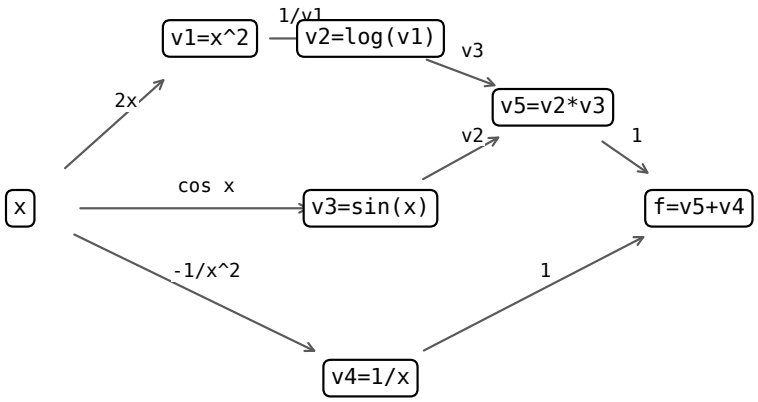


Figure 1.3: Dependency graph with local derivative labels. Forward tangent products and reverse adjoint products use the same eight edges.

1.13 AC-013 | Derivative labels on the expression graph

Reference: [1], exercise 6.

Source task and conventions

Differentiate the graph for $f(x) = \log(x^2)\sin x + x^{-1}$ and label the local derivatives.

Conventional solution and proof

The edge derivatives are

$$\frac{\partial v_1}{\partial x} = 2x, \quad \frac{\partial v_2}{\partial v_1} = v_1^{-1}, \quad \frac{\partial v_3}{\partial x} = \cos x,$$
$$\frac{\partial v_4}{\partial v_2} = v_3, \quad \frac{\partial v_4}{\partial v_3} = v_2, \quad \frac{\partial v_5}{\partial x} = -x^{-2}, \quad \frac{\partial v_6}{\partial v_4} = \frac{\partial v_6}{\partial v_5} = 1.$$

Summing products along the three directed paths from x to v_6 gives

$$\boxed{f'(x) = \frac{2\sin x}{x} + \log(x^2)\cos x - \frac{1}{x^2}.}$$

Each path term is the chain rule; the sum is the derivative of the addition node. This proves the result on both real half-lines.

Structural or analytic derivation

The same derivative is the coefficient of ε in the six-node jet evaluation. The product node produces two contributions, and the reciprocal produces the negative term. This gives an algebraic check that no path contribution was omitted. The graph figure prints each of the seven local edge labels.

Comparative cost and preferred method

Both scalar symbolic differentiation and graph accumulation use a constant number of elementary operations. Common-subexpression reuse avoids repeated logarithm and sine evaluations. No asymptotic speedup is claimed for this one-input, one-output graph; the measured comparison treats it as a correctness control.

Graphical reference. See Figure 1.3 for the shared construction.

1.14 AC-014 | Forward and reverse accumulation

Reference: [1], exercise 7.

Source task and conventions

Evaluate the derivative of $\log(x^2)\sin x + x^{-1}$ by both forward and reverse differentiation.

Conventional solution and proof

For forward mode set $\dot{x} = 1$. The tangents are

$$\dot{v}_1 = 2x, \quad \dot{v}_2 = 2/x, \quad \dot{v}_3 = \cos x,$$
$$\dot{v}_4 = (2/x)\sin x + \log(x^2)\cos x, \quad \dot{v}_5 = -x^{-2}, \quad \dot{v}_6 = \dot{v}_4 + \dot{v}_5.$$

For reverse mode start with $\bar{v}_6 = 1$. Then $\bar{v}_4 = \bar{v}_5 = 1$, $\bar{v}_2 = v_3$, $\bar{v}_3 = v_2$, and $\bar{v}_1 = \bar{v}_2/v_1$. Contributions to the single input accumulate as

$$\bar{x} = 2x\bar{v}_1 + \cos x\bar{v}_3 - x^{-2}\bar{v}_5 = \frac{2\sin x}{x} + \log(x^2)\cos x - x^{-2}.$$

Thus the two sweeps agree term by term. Every local rule is a proved scalar derivative, and applying the chain rule in topological or reversed topological order establishes correctness.

Structural or analytic derivation

Forward mode transports a tangent vector through the graph; reverse mode transports a covector through the transposed local differentials. Their equality here follows from the associativity of the Jacobian product: for a scalar chain, transposition only reverses the order of multiplication and accumulation. The common numerical test evaluates both on a grid bounded away from zero.

Comparative cost and preferred method

Both modes traverse the six-node graph once after the primal values are known. With one input and one output each is $\Theta(L)$ for L nodes; neither has a dimensional advantage. Reverse mode stores the primal tape, while forward mode can discard values after their last use. This source problem is a tie in asymptotic arithmetic and a small-memory preference for forward mode.

Graphical reference. See Figure 1.3 for the shared construction.

1.15 AC-015 | When each differentiation mode is preferable

Reference: [1], exercise 8.

Source task and conventions

Compare forward and reverse automatic differentiation in arithmetic, storage and parallelism.

Conventional solution and proof

Consider a straight-line program with L elementary operations computing $F : \mathbb{R}^n \to \mathbb{R}^m$. A forward sweep seeded by v computes $J_F v$ with a constant amount of derivative arithmetic per node. Therefore n basis-vector sweeps recover a full Jacobian in $O(nL)$ work. A reverse sweep seeded by w computes $J_F^{\mathsf{T}} w$, so m basis-vector sweeps cost $O(mL)$. These bounds follow by applying the local chain rule once per node per seed. A straightforward reverse implementation stores $O(L)$ primal intermediates. Forward mode stores tangents with the live primal values, requiring $O(W)$ storage per seed when W is the maximum live width. Checkpointing can trade reverse storage for recomputation. Independent seeds and independent graph branches can be parallelized, but dependency chains retain their depth.

Structural or analytic derivation

Use forward mode for a small input dimension, directional derivatives or Jacobian–vector products. Use reverse mode for a scalar loss with many parameters, or vector–Jacobian products. Hessian–vector products can combine a forward directional sweep with reverse gradients. Holomorphic functions admit complex derivatives, while general real losses on complex arrays require the real differential; the mode choice does not remove this distinction.

Comparative cost and preferred method

For a full Jacobian the dimension-dependent work comparison is nL versus mL. At $n = 1000, m = 1$, reverse mode needs one seed instead of 1000 forward seeds, a factor of 1000 in seed-sweep count under this model. This is not a measured runtime factor and does not include parallel scheduling or checkpoint overhead. For $n = m = 1$ the count is tied. Both methods must write mn entries when a dense full Jacobian is requested.

1.16 AC-016 | Transposition is an involution

Reference: [1], exercise 1.

Source task and conventions

Prove $(A^\mathsf{T})^\mathsf{T} = A$ for a rectangular matrix.

Conventional solution and proof

If $A \in \mathbb{R}^{m \times n}$, then for every valid i, j, $[(A^\mathsf{T})^\mathsf{T}]_{ij} = (A^\mathsf{T})_{ji} = A_{ij}$. Both matrices have shape $m \times n$ and identical entries, hence are equal. The argument applies over any coefficient ring.

Structural or analytic derivation

A transpose is the permutation $(i, j) \mapsto (j, i)$ of the two coordinate positions. Applying this permutation twice is the identity. The proof does not require moving any stored elements.

Comparative cost and preferred method

The mathematical identity has a uniform index proof. A transpose implemented as a stride view costs $O(1)$ metadata, so two views cost $O(1)$. Materializing two copies costs $\Theta(mn)$ memory traffic. The advantage is representation-dependent; a conventional array view already attains it.

1.17 AC-017 | Linearity of transposition

Reference: [1], exercise 2.

Source task and conventions

Prove $A^\mathsf{T} + B^\mathsf{T} = (A + B)^\mathsf{T}$ for equal-shaped matrices.

Conventional solution and proof

For every i, j, $[(A + B)^\mathsf{T}]_{ij} = A_{ji} + B_{ji} = [A^\mathsf{T} + B^\mathsf{T}]_{ij}$. Equality of all entries establishes the claim. Equal shapes are required for the sum to be defined.

Structural or analytic derivation

Coordinate permutation is linear: it commutes with coordinatewise addition. This proves the identity as a statement about linear maps on the space of matrices. It also justifies moving a transpose through an addition in a compiler's expression graph.

Comparative cost and preferred method

Either parenthesization performs mn scalar additions for a dense output. A view-based transpose adds only constant metadata; a copying transpose can add one or two mn-element copies. Thus the arithmetic count is tied, with potential memory-traffic savings determined by the implementation contract.

1.18 AC-018 | Symmetrization

Reference: [1], exercise 3.

Source task and conventions

For square A, prove that $A + A^{\mathsf{T}}$ is symmetric using the two preceding identities.

Conventional solution and proof

By linearity and involution,

$$(A + A^{\mathsf{T}})^{\mathsf{T}} = A^{\mathsf{T}} + (A^{\mathsf{T}})^{\mathsf{T}} = A^{\mathsf{T}} + A = A + A^{\mathsf{T}}.$$

This is exactly the definition of symmetry. No assumption that A was symmetric is used.

Structural or analytic derivation

The map $P(A) = (A + A^{\mathsf{T}})/2$ is a projection onto the symmetric subspace: $P(P(A)) = P(A)$ follows from the proved symmetry. Moreover $A - P(A) = (A - A^{\mathsf{T}})/2$ is skew-symmetric. Their Frobenius inner product is zero because each off-diagonal pair cancels and the skew diagonal vanishes. Hence P is the orthogonal symmetrization projection.

Comparative cost and preferred method

The source proof uses two reusable identities. Forming the whole output costs $\Theta(n^2)$, which is optimal for a dense $n \times n$ output. Symmetric storage needs $n(n+1)/2$ entries instead of n^2, but only when the required output accepts that representation. The basic proof is already optimal in explanatory economy without a meaningful timing metric.

1.19 AC-019 | Tensor norms and library conventions

Reference: [1], exercise 9.

Source task and conventions

Apply a norm routine to a tensor of at least three axes and identify the quantity computed.

Conventional solution and proof

The answer depends on the named routine and its parameters. The executed example uses NumPy with no explicit order or axis:

```
X = np.arange(24., dtype=np.float64).reshape(2, 3, 4)
v = np.linalg.norm(X)
```

It returns the Euclidean norm of the flattened array,

$$v = \left(\sum_{i,j,k} |X_{ijk}|^2 \right)^{1/2} = \sqrt{\sum_{r=0}^{23} r^2} = \sqrt{4324}.$$

NumPy's documented default with both `ord=None` and `axis=None` is this flattened norm. An explicitly specified matrix spectral or nuclear norm has a different input contract. The corresponding PyTorch `linalg.vector_norm` also flattens by default. The local version and the numerical output are recorded with the tests.

Structural or analytic derivation

Flattening is an isometry from the coefficient tensor equipped with the sum-of-squares inner product to $\mathbb{R}^{24}$. For complex entries use $|X_{ijk}|^2 = \overline{X_{ijk}}X_{ijk}$. This is a Hilbert norm of coefficients; it is not generally the operator norm of the tensor interpreted as a multilinear map.

Comparative cost and preferred method

For N arbitrary entries, the default norm uses $\Theta(N)$ work, and every entry can affect the result, giving an $\Omega(N)$ input-reading lower bound. Flattening by a view avoids copying when the strides permit; a noncontiguous layout may require a copy. No complex-analytic speedup applies.

Source qualification

The source says only `linalg.norm`; this solution explicitly fixes NumPy's default contract and records a PyTorch equivalent. Other choices of axis or order are not silently equated. Official documentation is listed as references N1 and N2.

1.20 AC-020 | Optimal association of a three-matrix product

Reference: [1], exercise 10.

Source task and conventions

Compare $(AB)C$ with $A(BC)$ for shapes $2^{10} \times 2^{16}$, $2^{16} \times 2^5$, and $2^5 \times 2^{14}$.

Conventional solution and proof

For dense classical multiplication of an $r \times s$ matrix by an $s \times t$ matrix, each of the rt outputs is a length-s dot product: rst multiplications and $rt(s-1)$ additions. Thus $(AB)C$ uses

$$2^{31} + 2^{29} = 2,684,354,560$$

multiplications. The alternative $A(BC)$ uses

$$2^{35} + 2^{40} = 1,133,871,366,144.$$

The exact multiplication-count ratio is 422.4. The left intermediate has 2^{15} entries; the right has 2^{30}. At eight bytes per entry these occupy 256 KiB and 8 GiB, respectively: a factor of $32,768$. Common input and final-output storage is excluded from both intermediate counts. Associativity over exact real arithmetic proves output equivalence; floating-point association can give different rounding errors.

Structural or analytic derivation

Treat matrix multiplication as contraction of a three-edge tensor network. Eliminating the large 2^{16} index first produces the small $2^{10} \times 2^5$ intermediate. Enumerating the two possible binary association trees proves that the left association minimizes classical multiplication count and intermediate size among these two schedules. The random Gaussian initialization does not change these dimension-based counts.

Comparative cost and preferred method

The left association is the proved optimum within the two classical dense schedules. The reduction is 422.4 in multiplication count and $32,768$ in intermediate entry count. These are exact arithmetic/storage ratios, not measured wall-clock speedups. Scaled, matched-output experiments test the trend without allocating the full 8 GiB intermediate; fast matrix multiplication and hardware-dependent blocking lie outside the cost model.

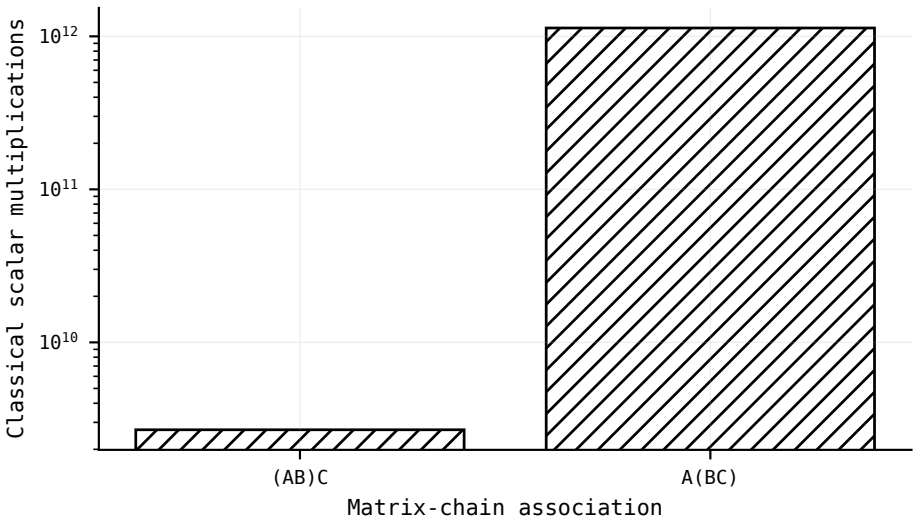


Figure 1.4: The two exact multiplication counts differ by factor 422.4; the intermediate arrays differ by factor 32,768 in entries.

1.21 AC-021 | Transpose views and equal arithmetic counts

Reference: [1], exercise 11.

Source task and conventions

Compare AB and AC^{T} for the printed shapes, then let $C = B^{\mathsf{T}}$ without copying.

Conventional solution and proof

Here A has shape 1024×65536, B has shape 65536×32, and C has shape 32×65536. Both products have shape 1024×32 and use 2^{31} classical scalar multiplications. If B and C are independently generated, the outputs differ, although the operation counts agree. Performance can differ because of memory order and packing. If $C = B^{\mathsf{T}}$ is a view, then $C^{\mathsf{T}} = B$ as an indexed array and both expressions use the same logical right factor. Double transposition restores the original strides. A view shares storage and needs only metadata; cloning creates another 2^{21}-entry array, or 16 MiB in double precision. A modification through the view also changes the shared base array.

Structural or analytic derivation

The index permutation viewpoint separates algebraic equality from physical layout. The contraction indices are identical in the two expressions; no spectral or complex transformation changes their mathematical operation count. A compiler or BLAS library may exploit transpose flags without materializing a copy.

Comparative cost and preferred method

The two methods are tied at 2^{31} multiplications. View construction is $O(1)$ metadata, compared with $\Theta(2^{21})$ elements copied when cloning. Wall-clock differences are implementation-dependent and are measured separately on a scaled instance. No universal faster product can be proved from the shapes alone.

1.22 AC-022 | A stack along the third axis

Reference: [1], exercise 12.

Source task and conventions

Stack three 100×200 matrices so that the second slice of the third axis is B, and verify the result.

Conventional solution and proof

Choose the axis explicitly:

```
T = np.stack([A, B, C], axis=2)
assert T.shape == (100, 200, 3)
assert np.array_equal(T[:, :, 1], B)
```

By definition, $T_{ij0} = A_{ij}$, $T_{ij1} = B_{ij}$ and $T_{ij2} = C_{ij}$, proving the assertion for every entry. The tensor has three axes and $60,000$ scalar entries. The second coordinate is index one in zero-based indexing.

Structural or analytic derivation

The operation adjoins a three-valued label index to the matrix coordinates. Projection onto label one recovers B. A default stack on axis zero would instead have shape $(3, 100, 200)$ and require $T[1, :, :]$; the source's requested third-axis slice therefore determines the explicit axis choice.

Comparative cost and preferred method

Materializing the tensor copies $60,000$ entries, an unavoidable $\Theta(60,000)$ output cost for a new contiguous stack. The slice can be a constant-metadata view with no element copy. The index-level proof is the preferable method; complex analysis supplies no additional reduction.

Source qualification

The axis is specified to make the source's requested third-axis recovery correct. The default first-axis stack is recorded separately, not silently interchanged.

1.23 AC-023 | Variance and concentration of a coin estimate

Reference: [1], exercise 3.

Source task and conventions

For independent Bernoulli(p) observations, compute the variance of the sample head fraction, a Chebyshev bound and the central-limit relation.

Conventional solution and proof

Let $X_i \in \{0,1\}$ with $\mathbb{P}(X_i = 1) = p$, independently, and $\widehat{p}_n = n^{-1}\sum_i X_i$. Since $X_i^2 = X_i$, $\mathbb{E}X_i = p$ and $\operatorname{Var} X_i = p - p^2$. Expanding the variance of the sum, every off-diagonal covariance vanishes by independence. Therefore

$$\mathbb{E}\widehat{p}_n = p, \qquad \boxed{\operatorname{Var}\widehat{p}_n = p(1-p)/n.}$$

For $\varepsilon > 0$, the pointwise inequality $\varepsilon^2 \mathbf{1}_{\{|\widehat{p}_n - p| \ge \varepsilon\}} \le (\widehat{p}_n - p)^2$ and expectation give

$$\mathbb{P}(|\widehat{p}_n - p| \ge \varepsilon) \le \frac{p(1-p)}{n\varepsilon^2} \le \frac{1}{4n\varepsilon^2}.$$

For $0 < p < 1$, set $Y = (X-p)/\sqrt{p(1-p)}$. Its characteristic function has expansion $\mathbb{E}e^{itY} = 1 - t^2/2 + O(|t|^3)$, because Y is bounded, has mean zero and variance one. Independence gives $[1 - t^2/(2n) + O(n^{-3/2})]^n \to e^{-t^2/2}$ for the normalized sum. By the characteristic-function continuity theorem, $\sqrt{n}(\widehat{p}_n - p)/\sqrt{p(1-p)}$ converges in distribution to the standard normal. At $p = 0$ or one the estimate is deterministic.

Structural or analytic derivation

The probability generating function is $(1 - p + pz)^n$. Its first two derivatives at $z = 1$ give $\mathbb{E}S_n = np$ and $\mathbb{E}[S_n(S_n - 1)] = n(n-1)p^2$, hence the same variance. The analytic characteristic-function expansion explains the $n^{-1/2}$ standard-error scale and the normal limiting shape. The continuity theorem is used only for the stated distributional interpretation; the exact variance and finite-sample bound above have complete elementary proofs.

Comparative cost and preferred method

Enumerating all 2^n binary sequences is unnecessary. Independence or two generating-function derivatives give the variance in constant symbolic work and $O(1)$ numerical arithmetic after n, p are supplied. Computing the observed fraction still reads n observations, so it costs $\Theta(n)$. Conventional variance additivity already attains the efficient formula; the transform supplies a unified derivation, not a new speedup.

1.24 AC-024 | Shared sample means are dependent

Reference: [1], exercise 4.

Source task and conventions

For $z_m = m^{-1}\sum_{i=1}^{m} X_i$ with mean-zero unit-variance samples, clarify whether Chebyshev can be applied to every z_m independently.

Conventional solution and proof

Under independent sampling, $\mathbb{E}z_m = 0$ and $\operatorname{Var} z_m = 1/m$. Thus for every fixed m,

$$\mathbb{P}(|z_m| \geq \varepsilon) \leq 1/(m\varepsilon^2).$$

No independence between different z_m is needed to apply this inequality separately. However, for $m \leq n$ the means share their first m observations, and

$$\operatorname{Cov}(z_m, z_n) = \frac{1}{mn} \sum_{i \leq m, j \leq n} \operatorname{Cov}(X_i, X_j) = \frac{1}{n}.$$

They are not independent, since their covariance is nonzero. Their tail-event probabilities cannot therefore be multiplied on an independence assumption. A union bound over a finite set of indices is valid, though often loose. If only the marginal mean and variance are assumed, even $\operatorname{Var} z_m = 1/m$ need not hold: taking all X_i equal gives variance one.

Structural or analytic derivation

The vector of partial means is a triangular linear transform of the samples. Its covariance matrix has entries $1/\max(m, n)$. This operator representation exposes every dependence simultaneously and supplies a direct check on any joint concentration calculation. Chebyshev is a marginal second-moment result and remains valid for each coordinate.

Comparative cost and preferred method

The marginal bound takes constant arithmetic per chosen m. Constructing the full $N \times N$ covariance matrix requires $\Theta(N^2)$ outputs; a lazy formula represents every entry in $O(1)$ storage. The main improvement is correction of a false independence inference, not a runtime claim.

Source qualification

The source's wording suggests a negative answer. The precise answer separates valid individual applications of Chebyshev from invalid probabilistic independence of overlapping sample means.

1.25 AC-025 | Three-node Markov factorization

Reference: [1], exercise 6.

Source task and conventions

Simplify the joint law of A, B, C when $A \to B \to C$ is a Markov chain.

Conventional solution and proof

The probability chain rule gives $P(a, b, c) = P(a)P(b \mid a)P(c \mid a, b)$ whenever the conditionals are defined. The Markov assumption is $C \perp A \mid B$, so $P(c \mid a, b) = P(c \mid b)$ on positive-probability conditioning events. Hence

$$\boxed{P(a, b, c) = P(a)P(b \mid a)P(c \mid b).}$$

On zero-probability conditioning events the joint mass is zero and conditional kernels may be defined arbitrarily without changing the joint law. The same factorization holds for densities when such densities exist.

Structural or analytic derivation

For finite state spaces, the first transition is a stochastic matrix K_{AB} and the second is K_{BC}. Marginalizing B gives the matrix composition $K_{AC} = K_{AB}K_{BC}$. The factorization is an exact finite-state representation. No unconditional independence of A and C is inferred.

Comparative cost and preferred method

With k states at each node, a dense joint table has $k^3 - 1$ free probabilities. The factorized model has $(k-1) + 2k(k-1) = 2k^2 - k - 1$ free parameters. A single joint entry needs two multiplications after factor lookup. The storage reduction follows from the stated conditional independence and is already the standard graphical-model method.

1.26 AC-026 | Dependent-test Bayes calculation

Reference: [1], exercise 7.

Source task and conventions

Solve the supplied probability exercise with prior 0.0015, sensitivity 0.99, false-positive probability 0.1, conditional independence when $H = 1$, and $P(D_1 = D_2 = 1 \mid H = 0) = 0.02$.

Conventional solution and proof

Under $H = 0$, both marginals have positive probability 0.1. Subtracting the common $(1, 1)$ cell gives

	$D_2 = 0$	$D_2 = 1$
$D_1 = 0$	0.82	0.08
$D_1 = 1$	0.08	0.02

.

Under $H = 1$, independence gives cells $0.0001, 0.0099, 0.0099, 0.9801$ in the same order. For one positive test,

$$P(H = 1 \mid D_1 = 1) = \frac{0.0015(0.99)}{0.0015(0.99) + 0.9985(0.1)} = \frac{297}{20267} \approx 0.014654.$$

For two positive tests,

$$\boxed{P(H = 1 \mid D_1 = D_2 = 1) = \frac{0.0015(0.9801)}{0.0015(0.9801) + 0.9985(0.02)} = \frac{29403}{428803} \approx 0.068570.}$$

Each denominator is the total probability of the conditioning event, obtained by partitioning on H. All four cells sum to one and have the required marginals.

Structural or analytic derivation

Posterior odds equal prior odds times the likelihood ratio. The one-test likelihood ratio is $0.99/0.1 = 9.9$; the two-test ratio is $0.9801/0.02 = 49.005$. Squaring 9.9 would incorrectly impose independence under $H = 0$. Thus a likelihood-ratio representation produces the same answers and makes the effect of the dependent false positives explicit.

Comparative cost and preferred method

Both Bayes-table and odds methods require a fixed number of products and one division. There is no asymptotic reduction for two tests. The odds formulation is convenient for repeated evidence, provided the joint likelihood is correctly modeled. This is a numerical probability exercise, not clinical guidance.

1.27 AC-027 | Mean–variance portfolio algebra

Reference: [1], exercise 8.

Source task and conventions

Compute the expected return and variance of weights α with $\mathbf{1}^\mathsf{T}\alpha = 1$, describe unconstrained return maximization, and formulate a variance-constrained problem.

Conventional solution and proof

Linearity of expectation gives $\mathbb{E}(\alpha^{\mathsf{T}} s) = \alpha^{\mathsf{T}} \mu$. Centering and expanding the square give

$$\operatorname{Var}(\alpha^{\mathsf{T}} s) = \alpha^{\mathsf{T}} \Sigma \alpha.$$

With nonnegative weights, expected return is a convex combination of the means and is maximized by allocating to an index of maximal mean; ties allow any mixture of maximizers. The printed budget constraint alone does not impose nonnegative weights. If shorting is unrestricted and two means differ, weights $te_i + (1-t)e_j$ make expected return unbounded as t tends to the appropriate infinity. A precise risk-constrained formulation is

$$\max_{\alpha} \ \mu^{\mathsf{T}} \alpha \quad \text{subject to} \quad \mathbf{1}^{\mathsf{T}} \alpha = 1, \quad \alpha^{\mathsf{T}} \Sigma \alpha \le V,$$

with $\alpha \ge 0$ appended when a long-only portfolio is intended. The covariance is positive semidefinite because $v^{\mathsf{T}} \Sigma v = \mathbb{E}[v^{\mathsf{T}}(s-\mu)]^2 \ge 0$, so the variance constraint is convex.

Structural or analytic derivation

For the budget-only model with $\Sigma \succ 0$, define $a = \mathbf{1}^{\mathsf{T}} \Sigma^{-1} \mathbf{1}$, $b = \mathbf{1}^{\mathsf{T}} \Sigma^{-1} \mu$, $d = \mu^{\mathsf{T}} \Sigma^{-1} \mu - b^2/a$, and $\alpha_0 = \Sigma^{-1}\mathbf{1}/a$. Let $v = \Sigma^{-1}(\mu - b\mathbf{1}/a)$. Every budget-feasible $\alpha = \alpha_0 + w$ has $\mathbf{1}^{\mathsf{T}} w = 0$ and variance $1/a + w^{\mathsf{T}} \Sigma w$. Cauchy–Schwarz in the Σ inner product gives $\mu^{\mathsf{T}} w = v^{\mathsf{T}} \Sigma w \le \sqrt{d}\sqrt{w^{\mathsf{T}} \Sigma w}$. For $V \ge 1/a$ and $d > 0$, equality is attained uniquely by

$$\alpha^* = \alpha_0 + \sqrt{(V - 1/a)/d}\, v.$$

If $V < 1/a$ the feasible set is empty. If $d = 0$, all budget-feasible portfolios have the same expected return. This proves the closed-form optimum under the stated additional positive-definite, unrestricted-sign assumptions; it does not solve an arbitrary long-only active-set problem.

Comparative cost and preferred method

Computing the two linear solves with one dense factorization costs $O(p^3)$ setup and $O(p^2)$ per right-hand side. A whole risk-budget curve then uses $O(p)$ vector arithmetic per point. A general quadratic-programming solve at every risk level repeats setup. The attained Cauchy–Schwarz bound certifies the optimum for this model; no investment-performance or universal solver-speed claim follows.

1.28 AC-028 | Length of a rank-three array

Reference: [1], exercise 4.

Source task and conventions

For X of shape $(2, 3, 4)$, determine and check $\operatorname{len}(X)$.

Conventional solution and proof

For a NumPy array, length is the size of axis zero. Thus $\operatorname{len}(X) = 2$. The executable check creates $X = \operatorname{arange}(24).\operatorname{reshape}(2, 3, 4)$ and asserts both the shape and the length. This differs from $X.\texttt{size} = 24$ and $X.\texttt{ndim} = 3$.

Structural or analytic derivation

Think of X as a sequence of two 3×4 arrays. Length counts the outer sequence, while total scalar size is the product $2 \cdot 3 \cdot 4$. The distinction follows directly from the indexing map.

Comparative cost and preferred method

Reading a stored shape entry takes $O(1)$ metadata operations. Counting all 24 scalar entries is unnecessary and answers a different question. The standard array operation is already the preferable method.

1.29 AC-029 | Length, leading axis and scalar exceptions

Reference: [1], exercise 5.

Source task and conventions

Identify the axis measured by length for an arbitrary-shaped tensor.

Conventional solution and proof

For a NumPy array with at least one axis and shape $(d_0, \ldots, d_{k-1})$, $\mathrm{len}(X) = d_0$. This includes $d_0 = 0$, which gives zero. A zero-dimensional scalar array has no leading axis and raises a type error when length is requested. Thus the claim requires positive rank.

Structural or analytic derivation

The indexing domain is a product of k finite sets. The array is a sequence indexed by the first set, whose cardinality is d_0. This gives a representation-level proof of the rule and explains the scalar exception.

Comparative cost and preferred method

The answer is an $O(1)$ metadata lookup, independent of the total entry count. No analytic transformation improves the output contract. The conventional representation is retained.

1.30 AC-030 | Shapes of axis reductions

Reference: [1], exercise 8.

Task and exact solution

For an array X of shape $(2, 3, 4)$, summing one axis removes that index when `keepdims=False`. The three outputs have shapes $(3, 4)$, $(2, 4)$ and $(2, 3)$:

$$(S_0)_{jk} = \sum_{i=0}^{1} X_{ijk}, \quad (S_1)_{ik} = \sum_{j=0}^{2} X_{ijk}, \quad (S_2)_{ij} = \sum_{k=0}^{3} X_{ijk}.$$

With retained axes their shapes are $(1, 3, 4)$, $(2, 1, 4)$ and $(2, 3, 1)$. These identities directly determine every output entry.

Structural interpretation and operation count

Each reduction contracts one tensor factor with its all-ones covector. Under separate binary-addition evaluation of each group, the respective counts are $12(2-1) = 12$, $8(3-1) = 16$ and $6(4-1) = 18$. A sum of r arbitrary inputs needs at least $r-1$ binary additions in this groupwise model: combining r disconnected input components into one requires at least $r-1$ merges. The displayed evaluation attains that count. Determining the output shape is constant-size metadata work. The structural description supplies the same calculation; no complex-analytic acceleration is asserted.

Chapter 2

Discrete Mathematics: An Open Introduction

Oscar Levin. Reference problems and derived analyses.

2.1 AC-031 | Four-colour squares and five-colour dominoes

Reference: [2], exercise 6.

Source task and conventions

Count length-n tilings with four square colours and five domino colours; give a recurrence, the first six positive-index terms and a closed form.

Conventional solution and proof

Set $a_0 = 1$ for the empty tiling and $a_1 = 4$. For $n \geq 2$, the final tile is either one of four squares, leaving a length-$n-1$ tiling, or one of five dominoes, leaving a length-$n-2$ tiling. These cases are disjoint and exhaustive, so

$$a_n = 4a_{n-1} + 5a_{n-2}.$$

The requested terms are $4, 21, 104, 521, 2604, 13021$. The characteristic polynomial factors as $(r-5)(r+1)$. Thus $a_n = c5^n + d(-1)^n$ is a solution. The initial conditions give $c+d = 1$, $5c-d = 4$, hence $c = 5/6, d = 1/6$:

$$\boxed{a_n = \frac{5^{n+1} + (-1)^n}{6}.}$$

Substitution verifies the recurrence; equality at indices zero and one implies equality at every index by induction, proving uniqueness.

Structural or analytic derivation

Summing the recurrence gives $(1 - 4z - 5z^2)A(z) = 1$. Since $1 - 4z - 5z^2 = (1-5z)(1+z)$,

$$A(z) = \frac{5/6}{1-5z} + \frac{1/6}{1+z}.$$

Geometric coefficient extraction yields the same closed form. The two poles also explain the dominant 5^n growth and alternating correction.

Comparative cost and preferred method

Enumerating all tilings takes at least a_n output operations. Counting the prefix by recurrence takes $n-1$ updates. A single remote coefficient uses one integer power by $O(\log n)$ multiplications, plus a parity decision and division by six. The characteristic-root baseline already obtains this formula; the generating function supplies an equivalent proof. For the first six terms, the recurrence is simpler than repeated powers.

2.2 AC-032 | Two-colour squares and three-colour dominoes

Reference: [2], exercise 7.

Source task and conventions

Give the recursive count and solve it by characteristic roots for two square colours and three domino colours.

Conventional solution and proof

There is one empty tiling and two tilings of length one. Partitioning by the last tile proves $a_0 = 1$, $a_1 = 2$ and $a_n = 2a_{n-1} + 3a_{n-2}$ for $n \geq 2$. The characteristic roots of $r^2 - 2r - 3$ are 3 and -1. The candidate $c3^n + d(-1)^n$ meets the initial conditions when $c = 3/4, d = 1/4$. Therefore

$$\boxed{a_n = (3^{n+1} + (-1)^n)/4.}$$

Both basis sequences satisfy the recurrence, and induction proves that this initial-value solution is unique.

Structural or analytic derivation

The generating function is $A(z) = (1 - 2z - 3z^2)^{-1} = \frac{3/4}{1-3z} + \frac{1/4}{1+z}$. Extracting the coefficient proves the same expression and records the two exponential modes.

Comparative cost and preferred method

The source explicitly requires characteristic roots, so that derivation is retained. It is as efficient symbolically as partial fractions. A prefix of length N uses $O(N)$ recurrence updates; a single large-index term uses $O(\log n)$ integer multiplications by exponentiation. No new advantage over the prescribed method is asserted.

2.3 AC-033 | A repeated unit root

Reference: [2], exercise 8.

Source task and conventions

Solve $a_n = 2a_{n-1} - a_{n-2}$, impose the stated initial values, and classify initial values giving $a_9 = 30$ or $a_9 = x$.

Conventional solution and proof

Rearrange to $a_n - a_{n-1} = a_{n-1} - a_{n-2}$. The first differences are constant, so telescoping gives

$$a_n = a_0 + n(a_1 - a_0).$$

This proves the complete two-parameter solution. With $(a_0, a_1) = (1, 2)$ it is $a_n = n + 1$. The condition $a_9 = 30$ is exactly $9a_1 - 8a_0 = 30$. Over the reals, choose any $a_0 = u$ and $a_1 = (30 + 8u)/9$. For integer initial values all solutions are $a_0 = 3 + 9t$, $a_1 = 6 + 8t$ with $t \in \mathbb{Z}$: the first equation modulo nine gives $a_0 \equiv 3$, and substitution gives the second. Every real x is attainable with real initial values by $a_0 = a_1 = x$; with integer initial values, precisely integer x are attainable. With nonnegative initial values, a nonnegative constant choice attains every nonnegative x in the same scalar domain.

Structural or analytic derivation

The generating function is

$$A(z) = \frac{a_0 + (a_1 - 2a_0)z}{2} = \frac{a_0}{} + (a_1 - a_0)\frac{z}{2}.$$

The double pole produces an affine polynomial in the index. Coefficient extraction gives the same formula, explaining the repeated-root case without a limiting argument between two equal roots.

Comparative cost and preferred method

The first-difference proof is the strongest conventional baseline and uses one invariant. Evaluating a_n needs one multiplication and one addition after the initial difference is formed, in unit-cost scalar arithmetic. The generating-function derivation adds setup and does not improve this evaluation. The preferred solution is telescoping.

2.4 AC-034 | A repeated root at two

Reference: [2], exercise 9.

Source task and conventions

Solve $a_n = 4a_{n-1} - 4a_{n-2}$ in general and for initial pairs $(1, 2)$ and $(1, 8)$.

Conventional solution and proof

Set $b_n = a_n/2^n$. Division of the recurrence by 2^n gives $b_n = 2b_{n-1} - b_{n-2}$. Its first differences are constant, so

$$\boxed{a_n = \left[a_0 + n\left(\frac{a_1}{2} - a_0\right)\right] 2^n.}$$

Substitution verifies the recurrence and the two initial conditions, and induction proves uniqueness. The first pair gives $a_n = 2^n$; the second gives $a_n = (1 + 3n)2^n$. The repeated characteristic root is two, and the $n2^n$ term is necessary to span all initial-value solutions.

Structural or analytic derivation

Summation gives $A(z) = [a_0 + (a_1 - 4a_0)z]/(1 - 2z)^2$. Using $[z^n](1 - 2z)^{-1} = 2^n$ and $[z^n]z(1 - 2z)^{-2} = n2^{n-1}$ yields the same expression. The double pole accounts for the linear factor in n.

Comparative cost and preferred method

Rescaling and one first-difference invariant solve the recurrence directly. A remote exact term uses a binary shift for 2^n and arithmetic on $\Theta(n)$-bit integers; outputting those bits already costs $\Omega(n)$ bit operations. A full prefix takes linear recurrence updates. The analytic and characteristic-root methods are equivalent in asymptotic coefficient arithmetic.

2.5 AC-035 | Seven elementary generating functions

Reference: [2], exercise 1.

Source task and conventions

Find the ordinary generating functions of all seven printed sequences, beginning each at index zero.

Conventional solution and proof

From $G(z) = 1/(1-z) = \sum_{n\geq 0} z^n$, formal differentiation gives $G'(z) = 1/(1-z)^2 = \sum_{n\geq 0}(n+1)z^n$. Scaling, shifting and replacing z give the complete table:

(a)	$4/(1-z)$
(b)	$2/(1-z)^2$
(c)	$2z^3/(1-z)^2$
(d)	$1/(1-5z)$
(e)	$1/(1+3z)$
(f)	$1/(1-5z^2)$
(g)	$z/(1-z^3)^2$.

For (c), multiplication by z^3 inserts exactly three zeros. For (f), the coefficient at $2k$ is 5^k and odd coefficients vanish. For (g), the coefficient at $3k+1$ is $k+1$ and the other two residue classes vanish. These descriptions reproduce every term, not just the displayed prefixes.

Structural or analytic derivation

All series are ordinary generating functions $A(z) = \sum_{n\geq 0} a_n z^n$. Identities hold formally: the coefficient of any fixed power uses finitely many operations. When complex-analytic evaluation is desired, the same identities hold inside the nearest pole. The geometric identity follows by multiplying $\sum_{n\geq 0} z^n$ by $1-z$; coefficient cancellation leaves one.

Comparative cost and preferred method

The source already requests generating functions; the standard method is therefore the same structural calculus used in the PHI synopsis. Rational simplification has fixed symbolic cost for these fixed-degree expressions. Extracting the first N coefficients by the resulting fixed-order recurrence takes $O(N)$ coefficient arithmetic and matches the $\Omega(N)$ output-count lower bound. Exact growing integers require their actual bit lengths; no constant-time claim is made for large-index integer output.

2.6 AC-036 | Sequences from five rational functions

Reference: [2], exercise 2.

Source task and conventions

Identify the sequences generated by the five rational functions in the question.

Conventional solution and proof

The geometric expansion and its derivative give:

$4z/(1-z)$	$a_0 = 0,\ a_n = 4\ (n \geq 1)$
$1/(1-4z)$	$a_n = 4^n$
$z/(1+z)$	$a_0 = 0,\ a_n = (-1)^{n-1}\ (n \geq 1)$
$3z/(1+z)^2$	$a_0 = 0,\ a_n = 3n(-1)^{n-1}\ (n \geq 1)$.

For the last function, multiplying $1+z+z^2$ by $\sum_{n\geq 0}(n+1)z^n$ gives $a_0 = 1$, $a_1 = 3$, and $a_n = (n+1)+n+(n-1) = 3n$ for $n \geq 2$. Its sequence is $1, 3, 6, 9, 12, \ldots$. The fourth sequence is $0, 3, -6, 9, -12, \ldots$; the third is $0, 1, -1, 1, -1, \ldots$.

Structural or analytic derivation

All series are ordinary generating functions $A(z) = \sum_{n\geq 0} a_n z^n$. Identities hold formally: the coefficient of any fixed power uses finitely many operations. When complex-analytic evaluation is desired, the same identities hold inside the nearest pole. The geometric identity follows by multiplying $\sum_{n\geq 0} z^n$ by $1-z$; coefficient cancellation leaves one.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.7 AC-037 | Triangular numbers by three constructions

Reference: [2], exercise 3.

Source task and conventions

Obtain the triangular-number generating function by two differentiations, differencing, and multiplication.

Conventional solution and proof

Use $T_n = n(n+1)/2$ for $n \geq 0$, so the indexed sequence is $0, 1, 3, 6, \ldots$. Let $G(z) = 1/(1-z)$. *Differentiation:* $zG'(z) + \frac{1}{2}z^2G''(z) = z/(1-z)^3$, and its coefficient is $n + n(n-1)/2 = T_n$. *Differencing:* $T_0 = 0$ and $T_n - T_{n-1} = n$, whence $(1-z)T(z) = \sum_{n\geq 1} nz^n = z/(1-z)^2$ and $T(z) = z/(1-z)^3$. *Multiplication:* the coefficient of z^n in $[z/(1-z)^2][1/(1-z)]$ is $\sum_{k=0}^{n} k = T_n$. Pairing k with $n-k$ proves this sum equals $n(n+1)/2$. If the first listed triangular number is one at index zero, the generating function is instead $T(z)/z = 1/(1-z)^3 = G''(z)/2$. Both indexing conventions are thereby explicit.

Structural or analytic derivation

The triple pole at one gives a degree-two coefficient polynomial. The identity $[z^n](1-z)^{-3} = \binom{n+2}{2}$ follows by multiplying three geometric series and counting nonnegative triples summing to n. Shifting by one gives $T_n = \binom{n+1}{2}$, agreeing with all three required derivations.

Comparative cost and preferred method

Each required derivation uses a constant number of rational operations. Directly summing the first n integers costs $n-1$ additions; the closed expression uses one product and division by two. The generating function and the conventional pairing proof both give the same efficient formula. The source explicitly asks for three derivations, so none is omitted to reduce page count.

2.8 AC-038 | A triangular perturbation of a constant sequence

Reference: [2], exercise 4.

Source task and conventions

Use differencing for $4, 5, 7, 10, 14, 19, 25, \ldots$.

Conventional solution and proof

With $a_0 = 4$, the first differences are $a_n - a_{n-1} = n$ for $n \geq 1$. Thus

$$(1-z)A(z) = 4 + \sum_{n\geq 1} nz^n = 4 + \frac{z}{(1-z)^2},$$

so

$$\boxed{A(z) = \frac{4}{1-z} + \frac{z}{(1-z)^3}.}$$

Coefficient extraction gives $a_n = 4 + n(n+1)/2$, whose first differences and initial value verify the intended sequence at every index.

Structural or analytic derivation

A rational pole decomposition separates a constant component from the triangular component. The formula $A(z) = (4 - 7z + 4z^2)/(1-z)^3$ is equivalent and can be checked by bringing the preceding terms to a common denominator.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.9 AC-039 | An inhomogeneous numerator from initial data

Reference: [2], exercise 5.

Source task and conventions

Find the generating function for $a_n = 3a_{n-1} - a_{n-2}$, $a_0 = 1, a_1 = 5$.

Conventional solution and proof

Sum over $n \geq 2$: $A - 1 - 5z = 3z(A-1) - z^2 A$. Rearrangement yields

$$\boxed{A(z) = \frac{1+2z}{1-3z+z^2}.}$$

The coefficient at zero is one, at one is five, and the coefficients from index two satisfy the source recurrence because multiplication by the denominator cancels them. This proves both correctness and uniqueness.

Structural or analytic derivation

The two roots of $r^2 - 3r + 1$ are $r_\pm = (3 \pm \sqrt{5})/2$. Partial fractions give

$$a_n = \frac{7+\sqrt{5}}{2\sqrt{5}} r_+^n + \frac{\sqrt{5}-7}{2\sqrt{5}} r_-^n.$$

The constants solve $c_+ + c_- = 1$ and $c_+ r_+ + c_- r_- = 5$. This optional coefficient formula shows how the poles encode the two recurrence modes.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.10 AC-040 | The Fibonacci generating function

Reference: [2], exercise 6.

Source task and conventions

Use the Fibonacci recurrence to derive its generating function.

Conventional solution and proof

Use the source's $F_0 = 0, F_1 = 1$ convention. For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$. Writing $F(z) = \sum_{n\geq 0} F_n z^n$ gives $F - z = zF + z^2F$, and therefore

$$\boxed{F(z) = z/(1 - z - z^2).}$$

Multiplication by the denominator recovers the initial terms and recurrence, proving the result. For the shifted sequence $1, 1, 2, 3, \ldots$ beginning at zero, the function is $F(z)/z = 1/(1 - z - z^2)$; this shift is used explicitly in Exercise 13.

Structural or analytic derivation

The factorization $1-z-z^2 = (1-\varphi z)(1-\psi z)$, with $\varphi = (1+\sqrt{5})/2$, $\psi = (1-\sqrt{5})/2$, yields $F_n = (\varphi^n - \psi^n)/\sqrt{5}$. Substitution of the two roots and the initial conditions is an independent proof of this coefficient formula.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

Source qualification

The convention $F_0 = 0, F_1 = 1$ is stated in the supplied book's recurrence discussion, physical p.378. The shifted sequence used in a later exercise is kept distinct.

2.11 AC-041 | Partial sums of Fibonacci numbers

Reference: [2], exercise 7.

Source task and conventions

Use multiplication to obtain the generating function of $S_n = \sum_{k=0}^{n} F_k$.

Conventional solution and proof

The Cauchy product with $1/(1 - z)$ has coefficient $\sum_{k=0}^{n} F_k$, so

$$\boxed{S(z) = \frac{z}{(1 - z)(1 - z - z^2)}.}$$

The recurrence gives $F_k = F_{k+2} - F_{k+1}$. Telescoping from zero to n therefore proves $S_n = F_{n+2} - 1$, consistent with $S_0 = 0$. This also verifies the product coefficient independently.

Structural or analytic derivation

Multiplication by $(1-z)^{-1}$ is the discrete integration operator on coefficient sequences. It adds one simple pole at one to the Fibonacci resolvent, exactly accounting for the constant correction in $F_{n+2}-1$.

Comparative cost and preferred method

Computing each prefix sum independently uses $\Theta(N^2)$ additions for N outputs. One cumulative pass uses $\Theta(N)$, and the formula gives a remote value with a fast Fibonacci algorithm in $O(\log n)$ coefficient multiplications. The strongest standard cumulative-sum algorithm matches the linear prefix complexity; the generating function provides a proof and remote-query formula.

2.12 AC-042 | A sum of two geometric modes

Reference: [2], exercise 8.

Source task and conventions

Find the generating function for $a_n = 2 \cdot 5^n + 7(-3)^n$.

Conventional solution and proof

Summing the two geometric series separately gives

$$\boxed{A(z) = \frac{2}{1-5z} + \frac{7}{1+3z} = \frac{9-29z}{1-2z-15z^2}.}$$

The coefficient of z^n in the first expression is the specified closed formula. The rational identity follows by multiplying out its numerator. It also gives $a_0 = 9, a_1 = -11$ and $a_n = 2a_{n-1} + 15a_{n-2}$.

Structural or analytic derivation

All series are ordinary generating functions $A(z) = \sum_{n\geq 0} a_n z^n$. Identities hold formally: the coefficient of any fixed power uses finitely many operations. When complex-analytic evaluation is desired, the same identities hold inside the nearest pole. The geometric identity follows by multiplying $\sum_{n\geq 0} z^n$ by $1-z$; coefficient cancellation leaves one.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.13 AC-043 | A shifted geometric coefficient

Reference: [2], exercise 9.

Source task and conventions

Find a_n from $A(z) = 3z/(1-4z) + 1/(1-z)$.

Conventional solution and proof

The second summand contributes one at every index. The first contributes zero at index zero and $3 \cdot 4^{n-1}$ at index $n \geq 1$. Therefore

$$\boxed{a_0 = 1, \qquad a_n = 1 + 3 \cdot 4^{n-1} \quad (n \geq 1).}$$

This follows from $3z \sum_{k \geq 0} 4^k z^k = \sum_{n \geq 1} 3 \cdot 4^{n-1} z^n$. Extending the positive-index formula to zero would give an incorrect value.

Structural or analytic derivation

All series are ordinary generating functions $A(z) = \sum_{n \geq 0} a_n z^n$. Identities hold formally: the coefficient of any fixed power uses finitely many operations. When complex-analytic evaluation is desired, the same identities hold inside the nearest pole. The geometric identity follows by multiplying $\sum_{n \geq 0} z^n$ by $1 - z$; coefficient cancellation leaves one.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.14 AC-044 | A seventh coefficient of a product

Reference: [2], exercise 10.

Source task and conventions

Compute a_7 for $A(z) = 2(1 - z)^{-2} z (1 - z - z^2)^{-1}$.

Conventional solution and proof

The factors have coefficients $2(k + 1)$ and F_j, respectively. Hence

$$a_7 = 2 \sum_{j=0}^{7} (8 - j) F_j = 2(0 + 7 + 6 + 10 + 12 + 15 + 16 + 13) = \boxed{158}.$$

The Cauchy product is finite at degree seven, so no convergence assumption is needed.

Structural or analytic derivation

Twice cumulative summation gives the general identity

$$a_n = 2 \sum_{k=0}^{n} \sum_{j=0}^{k} F_j = 2 \sum_{k=0}^{n} (F_{k+2} - 1) = 2(F_{n+4} - n - 3).$$

The telescoping Fibonacci sum proves the last equality. At $n = 7$, $F_{11} = 89$ gives $2(89 - 10) = 158$.

Comparative cost and preferred method

For the requested seventh term the eight-term Cauchy product is already small. For a remote n, directly forming the weighted sum uses $n + 1$ terms, while the proved formula reduces it to one Fibonacci value and scalar arithmetic. Fast doubling computes that Fibonacci value in $O(\log n)$ coefficient multiplications. The gain is for the generalized remote-index workload, not a claimed large runtime saving for $n = 7$.

2.15 AC-045 | Coefficients of a double geometric pole

Reference: [2], exercise 11.

Source task and conventions

Explain why $(1-z)^{-2}$ generates $1, 2, 3, 4, \ldots$.

Conventional solution and proof

Multiply two geometric series. The coefficient of z^n is the number of ordered pairs of nonnegative integers (i, j) with $i + j = n$. The pairs are $(0, n), (1, n-1), \ldots, (n, 0)$, exactly $n + 1$ of them. Therefore $(1-z)^{-2} = \sum_{n \geq 0}(n+1)z^n$.

Structural or analytic derivation

Alternatively, formally differentiating $1/(1-z) = \sum_{n \geq 0} z^n$ gives the same expansion after shifting the summation index. Formal differentiation is valid coefficientwise; on $|z| < 1$ the analytic power series can also be differentiated term by term.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.16 AC-046 | Four transformations of the counting sequence

Reference: [2], exercise 12.

Source task and conventions

Starting with $(1-z)^{-2}$, obtain the four generating functions in the source.

Conventional solution and proof

Let $G(z) = \sum_{n \geq 0}(n+1)z^n = (1-z)^{-2}$. Then

	generating function	coefficient rule
(a)	$G(z^2) = (1-z^2)^{-2}$	$a_{2k} = k+1,\ a_{2k+1} = 0$
(b)	$G(-z) = (1+z)^{-2}$	$a_n = (-1)^n(n+1)$
(c)	$3zG(z) = 3z/(1-z)^2$	$a_n = 3n$
(d)	$3zG(z)/(1-z) = 3z/(1-z)^3$	$a_n = 3n(n+1)/2$.

The fourth is the cumulative sum of the third: $0, 3, 9, 18, 30, \ldots$. The shift and substitution rules follow by replacing powers in the defining series, and the cumulative-sum rule follows by the finite coefficient Cauchy product.

Structural or analytic derivation

The formal-series convention and geometric-series justification are established in AC-036. The displayed transformation uses that same coefficientwise arithmetic on its stated convergence domain.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.17 AC-047 | Four transformations of shifted Fibonacci numbers

Reference: [2], exercise 13.

Source task and conventions

Use $(1 - z - z^2)^{-1} = 1 + z + 2z^2 + 3z^3 + \cdots$ to identify the four sequences.

Conventional solution and proof

Let $b_n = F_{n+1}$, with $F_0 = 0, F_1 = 1$. The requested functions give:

$$\begin{array}{l|l} (a) & 0, 0, 1, 1, 2, 3, 5, \ldots; \quad a_n = 0 \ (n < 2),\ a_n = F_{n-1} \ (n \geq 2) \\ (b) & 1, 0, 1, 0, 2, 0, 3, 0, 5, \ldots; \quad a_{2k} = F_{k+1},\ a_{2k+1} = 0 \\ (c) & 1, 3, 18, 81, 405, \ldots; \quad a_n = 3^n F_{n+1} \\ (d) & 1, 2, 4, 7, 12, 20, \ldots; \quad a_n = F_{n+3} - 1. \end{array}$$

Parts (a)–(c) follow from multiplication by z^2, substitution z^2, and substitution $3z$. In (d), multiplication by $(1 - z)^{-1}$ gives $\sum_{k=0}^{n} F_{k+1}$, which telescopes to $F_{n+3} - 1$. These arguments establish all coefficients.

Structural or analytic derivation

All series are ordinary generating functions $A(z) = \sum_{n \geq 0} a_n z^n$. Identities hold formally: the coefficient of any fixed power uses finitely many operations. When complex-analytic evaluation is desired, the same identities hold inside the nearest pole. The geometric identity follows by multiplying $\sum_{n \geq 0} z^n$ by $1 - z$; coefficient cancellation leaves one.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.18 AC-048 | Alternating powers of two

Reference: [2], exercise 14.

Source task and conventions

Find the generating function of $1, -2, 4, -8, 16, \ldots$.

Conventional solution and proof

The coefficient formula is $a_n = (-2)^n$. The geometric identity with ratio $-2z$ gives

$$\boxed{A(z) = 1/(1 + 2z).}$$

Multiplication by $1 + 2z$ cancels every positive-degree coefficient and leaves one, proving the formal identity.

Structural or analytic derivation

The pole at $z = -1/2$ records the alternating sign and exponential magnitude. Analytically the power series converges for $|z| < 1/2$; the formal coefficient identity does not require evaluating z.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.19 AC-049 | Two initial constants followed by a counting sequence

Reference: [2], exercise 15.

Source task and conventions

Find the generating function for $1, 1, 1, 2, 3, 4, 5, 6, \ldots$.

Conventional solution and proof

The first two coefficients are one. Beginning at index two, the coefficients are $n - 1$. Therefore

$$\boxed{A(z) = 1 + z + \frac{z^2}{(1-z)^2}.}$$

Indeed, $z^2 \sum_{k\geq 0}(k+1)z^k$ contributes $1, 2, 3, \ldots$ precisely at indices two onward. This verifies the offset and all tail coefficients.

Structural or analytic derivation

The formal-series convention and geometric-series justification are established in AC-036. The displayed transformation uses that same coefficientwise arithmetic on its stated convergence domain.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

2.20 AC-050 | Reconstructing a sequence from its differences

Reference: [2], exercise 16.

Source task and conventions

For $A(z)$ generating $3, 5, 9, 15, 23, 33, \ldots$, find the generating function of consecutive differences in terms of A, identify it directly, and solve for A.

Conventional solution and proof

Let $b_n = a_{n+1} - a_n$, so $b_0 = 2, b_1 = 4, b_2 = 6, \ldots$. Then

$$B(z) = \sum_{n \geq 0} (a_{n+1} - a_n) z^n = \frac{A(z) - 3}{z} - A(z).$$

Since $b_n = 2(n+1)$, $B(z) = 2/(1-z)^2$. Solving $zB = (1-z)A - 3$ gives

$$\boxed{A(z) = \frac{3}{1-z} + \frac{2z}{(1-z)^3} = \frac{3 - 4z + 3z^2}{(1-z)^3}.}$$

The coefficient formula is $a_n = 3 + n(n+1)$; subtracting consecutive values gives $2(n+1)$ and $a_0 = 3$, proving the reconstruction.

Structural or analytic derivation

Forward differencing removes a pole order from polynomial sequences. Here an affine first difference gives a quadratic original coefficient, represented by a triple pole at one. The initial value supplies the independent constant-sequence term.

Comparative cost and preferred method

The fixed-order recurrence and output-size comparison in AC-035 applies to this coefficient sequence. The calculation above supplies its particular initial data and rational function.

Chapter 3

Introduction to Algorithms

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Reference problems and derived analyses.

3.1 AC-051 | Fibonacci numbers: generating function, rounding and

Reference: [3], exercise 4-5.

Source task and conventions

Solve all parts of CLRS Problem 4-5 using $F_0 = 0, F_1 = 1$: generating-function identity, factorization, coefficient formula, nearest-integer statement and the lower bound $F_{i+2} \geq \varphi^i$.

Conventional solution and proof

(a) Multiply $F_n = F_{n-1} + F_{n-2}$ by z^n and sum for $n \geq 2$. The initial terms give $F(z) = z + zF(z) + z^2F(z)$. (b) Therefore $F(z) = z/(1 - z - z^2)$. Let $\varphi = (1 + \sqrt{5})/2$ and $\psi = (1 - \sqrt{5})/2$. Their sum is one, product is minus one, and difference is $\sqrt{5}$, so

$$F(z) = \frac{z}{(1 - \varphi z)(1 - \psi z)} = \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \varphi z} - \frac{1}{1 - \psi z} \right).$$

(c) Expanding the two formal geometric series yields $F_n = (\varphi^n - \psi^n)/\sqrt{5}$. Formal coefficient uniqueness justifies the identification without imposing a numerical convergence domain. (d) Since $|\psi| < 1$, $|F_i - \varphi^i/\sqrt{5}| = |\psi|^i/\sqrt{5} < 1/2$. The integer F_i is therefore the unique nearest integer for every $i > 0$. (e) For $i = 0$, $F_2 = 1 = \varphi^0$; for $i = 1$, $F_3 = 2 \geq \varphi$. If the inequality holds at $i - 1, i - 2$, then $F_{i+2} = F_{i+1} + F_i \geq \varphi^{i-1} + \varphi^{i-2} = \varphi^i$, since $\varphi^2 = \varphi + 1$. Induction completes the proof.

Structural or analytic derivation

The resolvent denominator has two simple poles and coefficient extraction resolves the recurrence into its two spectral modes. For exact computation, avoid rounding an insufficiently precise floating Binet value. Fast doubling uses

$$F_{2k} = F_k(2F_{k+1} - F_k), \qquad F_{2k+1} = F_k^2 + F_{k+1}^2.$$

These identities follow from $\left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right)^k = \left(\begin{smallmatrix}F_{k+1}&F_k\\F_k&F_{k-1}\end{smallmatrix}\right)$, proved by induction, and squaring. Recursing on $\lfloor n/2 \rfloor$ returns (F_n, F_{n+1}) exactly.

Comparative cost and preferred method

The source's own generating-function approach is retained as the standard proof. Linear recurrence computation uses $\Theta(n)$ integer additions; fast doubling uses $O(\log n)$ multiplication-level stages. Since F_n has $\Theta(n)$ bits, these are not bit-complexity bounds or constant-time closed forms. The nearest-integer theorem requires enough precision to separate the true value from the half-integer boundary.

3.2 AC-052 | Binary-tree enumeration and the Catalan asymptotic

Reference: [3], exercise 12-4.

Source task and conventions

Solve CLRS Problem 12-4: the tree recurrence, quadratic generating function, exact Catalan coefficient and asymptotic estimate.

Conventional solution and proof

(a) The empty tree is unique, so $b_0 = 1$. A nonempty rooted ordered binary tree has one root, k left-subtree nodes and $n-1-k$ right-subtree nodes. This unique decomposition gives $b_n = \sum_{k=0}^{n-1} b_k b_{n-1-k}$. (b) The coefficient Cauchy product yields $B = 1 + zB^2$. Solving the quadratic and choosing the branch whose constant coefficient is one gives

$$B(z) = \frac{1 - \sqrt{1-4z}}{2z};$$

the other sign has a negative-power term and is not the required power series. (c) The formal binomial expansion gives

$$b_n = -\frac{1}{2}\binom{1/2}{n+1}(-4)^{n+1} = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n+1}\binom{2n}{n}.$$

For the algebraic simplification, expand the numerator of the generalized binomial coefficient as $(1/2)(-1/2)(-3/2)\cdots(-(2n-1)/2)$ and use $(2n)! = 2^n n!(1\cdot 3\cdots(2n-1))$. The case $n = 0$ is one directly. The recurrence and initial value determine every coefficient uniquely, so the formula counts the original trees. (d) Stirling's formula with remainder, $n! = \sqrt{2\pi n}(n/e)^n e^{r_n}$ with $r_n = O(1/n)$, gives $\binom{2n}{n} = 4^n(\pi n)^{-1/2}(1 + O(1/n))$. Dividing by $n+1 = n(1+1/n)$ proves

$$\boxed{b_n = \frac{4^n}{\sqrt{\pi}\, n^{3/2}}(1 + O(1/n)).}$$

The displayed error follows because $r_{2n} - 2r_n = O(1/n)$ and $e^{O(1/n)} = 1 + O(1/n)$.

Structural or analytic derivation

The square-root singularity at $z = 1/4$ explains the exponential factor 4^n and the algebraic factor $n^{-3/2}$. Exact coefficients also satisfy

$$b_{n+1} = \frac{2(2n+1)}{n+2} b_n, \quad b_0 = 1,$$

by dividing adjacent factorial formulas. Each division is exact because the values equal the proved integer tree counts. This recurrence is the useful computational consequence of the algebraic generating function.

Comparative cost and preferred method

The convolution recurrence uses $N(N+1)/2$ products to produce $b_1, \ldots, b_N$. The multiplicative recurrence uses N updates, each with small-index multiplication and exact division, reducing coefficient-operation growth from quadratic to linear. The source's closed formula and a competent combinatorial implementation already support this reduction. The asymptotic estimate alone is not an exact-integer algorithm.

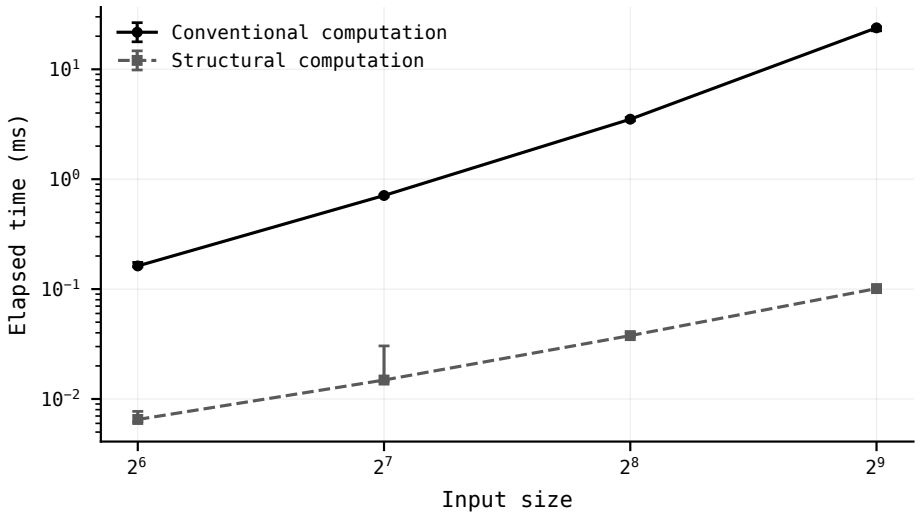


Figure 3.1: Catalan exact prefix: medians and min/max ranges of seven complete calls; both methods return the same full output. Setup in each implementation is included.

3.3 AC-053 | An explicit cubic polynomial product

Reference: [3], exercise 30.1-1.

Source task and conventions

Multiply $A(x) = 7x^3 - x^2 + x - 10$ and $B(x) = 8x^3 - 6x + 3$ by coefficient convolution and point-value multiplication.

Conventional solution and proof

Distributing each term and collecting equal powers gives

$$\boxed{A(x)B(x) = 56x^6 - 8x^5 - 34x^4 - 53x^3 - 9x^2 + 63x - 30.}$$

In ascending coefficient order, $a = (-10, 1, -1, 7)$ and $b = (3, -6, 0, 8)$. The coefficient formula $c_k = \sum_{i+j=k} a_i b_j$ yields $(-30, 63, -9, -53, -34, -8, 56)$. This proves the product coefficient by coefficient. Evaluating at any seven distinct points, multiplying the paired values and interpolating gives the same degree-at-most-six polynomial by uniqueness of polynomial interpolation.

Structural or analytic derivation

Pad both vectors to length eight. For $\omega = e^{2\pi i/8}$, evaluate with an eight-point DFT, multiply componentwise, and apply its inverse. Since the linear product has degree six, no coefficient wraps modulo $x^8 - 1$. Root-of-unity orthogonality proves that the inverse recovers exactly the displayed convolution in exact arithmetic. The supplied numerical test rounds only after comparing with the exact coefficient calculation.

Comparative cost and preferred method

The generic coefficient algorithm uses 16 scalar products for the 4×4 convolution, of which four vanish because one input coefficient is zero. An eight-point transform introduces complex operations and setup, so direct multiplication is preferable here. For degree-bound n inputs, the same transform construction reduces generic dense convolution from $\Theta(n^2)$ to $O(n \log n)$ arithmetic; this asymptotic statement does not predict a gain for cubics.

3.4 AC-054 | Synthetic division and Horner evaluation

Reference: [3], exercise 30.1-2.

Source task and conventions

Compute the quotient and remainder of $A(x)$ divided by $x - x_0$ in $\Theta(n)$ time, where $\deg A < n$.

Conventional solution and proof

For $n = 1$ the quotient is zero and the remainder is a_0. For $n \geq 2$, define

$$q_{n-2} = a_{n-1}, \qquad q_{j-1} = a_j + x_0 q_j \quad (j = n-2, \ldots, 1), \qquad r = a_0 + x_0 q_0.$$

Let $Q(x) = \sum_{j=0}^{n-2} q_j x^j$. The leading coefficient of $(x - x_0)Q + r$ is $q_{n-2} = a_{n-1}$; each middle coefficient is $q_{j-1} - x_0 q_j = a_j$; and the constant is $r - x_0 q_0 = a_0$. Thus $A = (x - x_0)Q + r$ identically, and substitution gives $r = A(x_0)$.

Structural or analytic derivation

The descending recurrence is Horner's rule with its intermediate values retained. It is also repeated extraction of the constant term in powers of $x - x_0$. The quotient coefficients are obtained during the same pass that evaluates the remainder; no separate interpolation is needed.

Comparative cost and preferred method

There are $n - 1$ multiplications and $n - 1$ additions after copying the leading coefficient, with $n - 1$ quotient outputs and one remainder. The $\Omega(n)$ input/output lower bound proves $\Theta(n)$ optimality for this representation. The standard synthetic-division algorithm is retained; a transform adds unnecessary setup.

3.5 AC-055 | Reversing a polynomial in point-value form

Reference: [3], exercise 30.1-3.

Source task and conventions

Convert a nonzero-node point-value representation of A into one for its degree-bound-n coefficient reversal.

Conventional solution and proof

From the definition,

$$A_{\mathrm{rev}}(x) = \sum_{j=0}^{n-1} a_{n-1-j} x^j = x^{n-1} A(1/x), \quad x \neq 0.$$

For every supplied pair (x_i, y_i) with $x_i \neq 0$, form

$$\boxed{\left(x_i^{-1},\ x_i^{-(n-1)} y_i\right).}$$

Indeed $A_{\mathrm{rev}}(1/x_i) = x_i^{-(n-1)} A(x_i)$. Distinct original nodes remain distinct after inversion. The transformed nodes need not be the original nodes, which is permitted in a point-value representation.

Structural or analytic derivation

This is pullback under the inversion map on the Riemann sphere together with the degree weight x^{n-1}. The weight makes the reversed polynomial regular at zero even though the formula uses $1/x$ away from zero. Its constant term is a_{n-1}, so the map's apparent singularity is removable for the polynomial.

Comparative cost and preferred method

With m supplied pairs, m reciprocals and m integer powers give $O(m \log n)$ field operations using binary exponentiation, plus m products. Reconstructing coefficients and reevaluating each transformed node can require $\Theta(n^2)$ work under conventional interpolation. If coefficients are already available, reversing them costs only $\Theta(n)$ and is the preferable representation. The nonzero-node hypothesis is essential.

3.6 AC-056 | The number of values needed for interpolation

Reference: [3], exercise 30.1-4.

Source task and conventions

Prove that fewer than n distinct values cannot uniquely specify a polynomial of degree below n.

Conventional solution and proof

Suppose $m < n$ distinct pairs (x_i, y_i) are consistent with one polynomial A of degree below n. Let $R(x) = \prod_{i=1}^{m}(x - x_i)$, with $R = 1$ for $m = 0$. For every scalar c, the polynomial $A_c = A + cR$ still has degree below n and satisfies all the pairs. Since R is nonzero, different c give different polynomials over an infinite field such as $\mathbb{R}$ or $\mathbb{C}$. Thus uniqueness fails. Conversely, if two degree-below-n polynomials agree at n distinct points, their difference has n roots and degree below n, hence is zero: repeated factorization by each distinct linear factor proves this elementary root bound.

Structural or analytic derivation

Evaluation at m nodes is a linear map from an n-dimensional coefficient space to an m-dimensional value space. Its kernel contains the nonzero polynomial R and in fact has dimension $n - m$, since every degree-below-n vanishing polynomial is divisible by R. This gives the exact ambiguity dimension.

Comparative cost and preferred method

The result is an information lower bound: at least n independent scalar values are required for arbitrary degree-below-n coefficients. Neither complex evaluation nor faster arithmetic removes it. The constructive vanishing-polynomial proof is the most direct conventional certificate.

3.7 AC-057 | Quadratic-time coefficient interpolation

Reference: [3], exercise 30.1-5.

Source task and conventions

Use the Lagrange formula to interpolate n distinct point-value pairs in $\Theta(n^2)$ arithmetic.

Conventional solution and proof

Construct $P(x) = \prod_{j=0}^{n-1}(x - x_j)$ by multiplying successive linear factors, using $O(n^2)$ coefficient arithmetic. For each k, synthetic division yields $P_k = P/(x - x_k)$ in $O(n)$ time. Compute $d_k = \prod_{j \neq k}(x_k - x_j)$ in $O(n)$ time; it is nonzero because the nodes are distinct. Accumulate

$$\boxed{A(x) = \sum_{k=0}^{n-1} \frac{y_k}{d_k} P_k(x).}$$

At $x = x_i$ every summand vanishes except $k = i$, whose value is y_i. The result has degree below n, so the preceding interpolation-uniqueness proof establishes correctness. Building P, all synthetic divisions, all denominators and all coefficient accumulations each use at most $O(n^2)$ operations.

Structural or analytic derivation

The quotient $P/(x - x_k)$ is the polynomial numerator of a residue-like interpolation basis, and $d_k = P'(x_k)$. A balanced product/remainder tree can eventually reduce large problems to quasi-linear polynomial arithmetic, but the source specifically asks for the elementary quadratic construction. The displayed algorithm supplies it completely.

Comparative cost and preferred method

The algorithm performs n length-n divisions and accumulations, hence $\Theta(n^2)$ in the stated implementation. Rebuilding every P_k independently by multiplying $n - 1$ linear factors would cost $\Theta(n^3)$. For repeated value queries, barycentric evaluation avoids materializing coefficients, but that is a different output contract.

3.8 AC-058 | Why pointwise ratios need not perform polynomial

Reference: [3], exercise 30.1-6.

Source task and conventions

Analyze point-value division both when polynomial division is exact and when it leaves a nonzero remainder.

Conventional solution and proof

Let $A = BQ + R$ with $\deg R < \deg B$. At any node where $B(x_i) \neq 0$,

$$A(x_i)/B(x_i) = Q(x_i) + R(x_i)/B(x_i).$$

If $R = 0$, the ratios are genuine quotient values. They determine Q when enough distinct nonzero-denominator nodes are retained and a suitable quotient degree bound is known. However, a root of B is also a root of A, so the ratio at such a node is $0/0$ and is undefined; for example $A = x^2, B = x$ at zero. The degree bound of Q can also be lower than the original representation's bound. If $R \neq 0$, the ratios are values of a rational function, not generally values of the polynomial quotient. For $A = x^2$, $B = x + 1$, one has $Q = x - 1, R = 1$. Ratios at zero and one are zero and $1/2$, whose linear interpolant is $x/2$, not $x - 1$. Thus the naive method fails even with no zero denominators.

Structural or analytic derivation

Polynomial division is a decomposition into a polynomial part at infinity and a proper rational remainder. Pointwise evaluation of the rational function does not separate these two parts. An analytic Laurent expansion at infinity would identify Q, but it must be accompanied by the degree and pole information that the naive division discards.

Comparative cost and preferred method

Exact division from valid values can be useful with an established quotient bound. General Euclidean division or fast polynomial division supplies the correct output directly. The counterexample uses two evaluations and completely refutes universal correctness; no benchmark can turn the invalid ratio rule into a valid faster method.

3.9 AC-059 | Cartesian sums through coefficient convolution

Reference: [3], exercise 30.1-7.

Source task and conventions

For two n-element sets of integers in $[0, 10n]$, compute all sum multiplicities in $O(n \log n)$ arithmetic.

Conventional solution and proof

Create indicator polynomials $A(z) = \sum_{a \in \mathcal{A}} z^a$ and $B(z) = \sum_{b \in \mathcal{B}} z^b$. Their product has coefficient

$$c_k = [z^k]A(z)B(z) = \#\{(a, b) \in \mathcal{A} \times \mathcal{B} : a + b = k\}.$$

This follows by distributing the finite sums; each ordered pair contributes one monomial. Form the product by a length-M convolution with $M \geq 20n + 1$ chosen as a suitable FFT length. Read the coefficients with $c_k > 0$ to obtain the Cartesian-sum set, and retain c_k as its multiplicity. Building the indicators and scanning the output cost $O(n)$; the transform product costs $O(n \log n)$ in the source's arithmetic model.

Structural or analytic derivation

For exact integer output, a number-theoretic transform with a supported primitive Mth root can replace complex floating arithmetic. Since the inputs are sets, every coefficient is at most n: for each a, at most one $b = k - a$ contributes. A prime modulus $p > n$ therefore determines the nonnegative coefficient uniquely, provided it supports the chosen transform. More generally CRT reconstruction requires a combined modulus exceeding a proved coefficient bound. A floating FFT needs an error enclosure below $1/2$ before rounding can be certified.

Comparative cost and preferred method

Direct pair enumeration uses exactly n^2 pair sums. The transform uses $O(n \log n)$ field operations and $O(n)$ storage for the bounded range. This proves the asymptotic arithmetic improvement for the specified range. Prime/root construction and bit arithmetic are additional costs outside the unit-field-operation bound. For sparse, small sets the direct algorithm may win, and it remains the exact benchmark control.

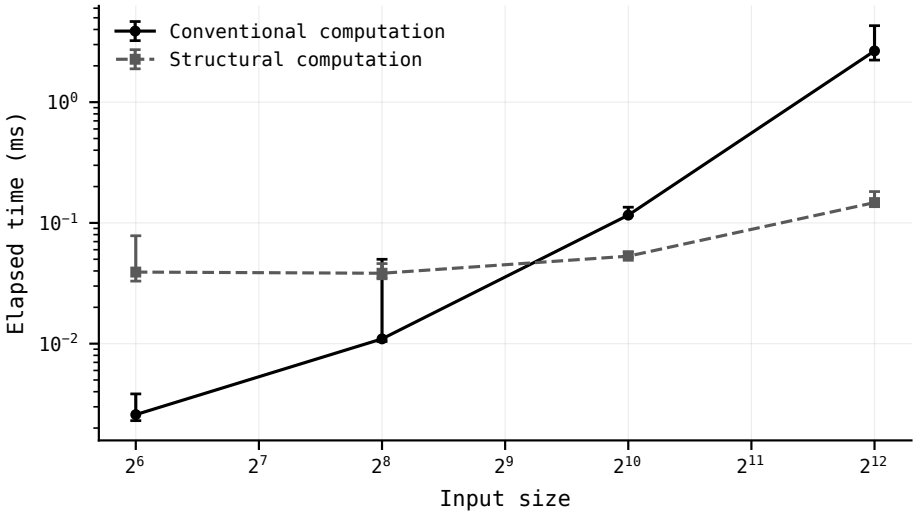


Figure 3.2: Linear convolution: medians and min/max ranges of seven complete calls; both methods return the same full output. Setup in each implementation is included.

3.10 AC-060 | A four-point DFT in the textbook sign convention

Reference: [3], exercise 30.2-2.

Source task and conventions

Compute the DFT of $(0, 1, 2, 3)$ with the CLRS convention $\omega_4 = e^{2\pi i/4} = i$.

Conventional solution and proof

By $y_k = \sum_{j=0}^{3} a_j i^{jk}$,

$$y_0 = 6, \quad y_1 = i - 2 - 3i = -2 - 2i, \quad y_2 = -1 + 2 - 3 = -2, \quad y_3 = -i - 2 + 3i = -2 + 2i.$$

Thus $\boxed{y = (6, -2 - 2i, -2, -2 + 2i)}$. The inverse formula $a_j = \frac{1}{4} \sum_k y_k i^{-jk}$ recovers $(0, 1, 2, 3)$; cancellation of the nonconstant geometric sums proves that inverse formula.

Structural or analytic derivation

Split into even input $(0, 2)$ and odd input $(1, 3)$. Their two-point transforms are $(2, -2)$ and $(4, -2)$. Combining with twiddles $1, i$ yields the same four outputs. This is the complete radix-two butterfly calculation.

Comparative cost and preferred method

For four inputs, direct arithmetic is already modest, and the butterfly mainly exposes reusable symmetry. The general radix-two recurrence is $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$ instead of $\Theta(n^2)$ direct sums. Many software libraries use the opposite exponential sign; comparing outputs requires this explicit convention.

3.11 AC-061 | Inverse FFT pseudocode and correctness

Reference: [3], exercise 30.2-4.

Source task and conventions

Give a $\Theta(n \log n)$ inverse DFT algorithm for power-of-two n.

Conventional solution and proof

Let `FFTplus` denote the source's positive-exponent radix-two transform. Then:

```
def inverse_dft(y):
    n = len(y)
    if n == 0: raise ValueError("nonempty transform")
    z = FFTplus([conjugate(v) for v in y])
    return [conjugate(v) / n for v in z]
```

Conjugating the transform changes ω_n^{jk} to ω_n^{-jk}, so the return value is $n^{-1} \sum_k y_k \omega_n^{-jk}$. Composing with the forward transform gives $n^{-1} \sum_l a_l \sum_k \omega_n^{(l-j)k}$. The inner sum is n when $l = j$ and zero otherwise: if $l \neq j$, multiply the finite geometric sum by $1 - \omega_n^{l-j} \neq 0$. Hence the composition is the identity.

Structural or analytic derivation

Equivalently, use the same recursive butterfly with conjugate roots and divide by n once at the top level. The normalized DFT is unitary because its rows are orthonormal by the same sum identity. Unitarity gives a norm-preserving exact transform; it does not eliminate floating-point rounding in its implementation.

Comparative cost and preferred method

One forward FFT plus two conjugation passes and one scaling pass costs $\Theta(n \log n)$. Normalization at every recursive level is unnecessary. This is the standard efficient inverse transform, so the PHI interpretation supplies a proof and conditioning information with no additional asymptotic speedup.

3.12 AC-062 | Radix-three Fourier recursion

Reference: [3], exercise 30.2-5.

Source task and conventions

Generalize the FFT to $n = 3^s$, give its recurrence and solve the recurrence.

Conventional solution and proof

Let $m = n/3$ and split a_j by $j = 3r + q$, $q = 0, 1, 2$. Compute the three length-m transforms $B_q(k) = \sum_{r=0}^{m-1} a_{3r+q}\omega_m^{rk}$. For $k = 0, \ldots, m-1$ and $t = 0, 1, 2$, combine

$$\boxed{y_{k+tm} = B_0(k) + \omega_n^{k+tm} B_1(k) + \omega_n^{2(k+tm)} B_2(k).}$$

Substitution of $j = 3r + q$ in the original DFT gives the formula because $\omega_n^{3r(k+tm)} = \omega_m^{rk}$. The base transform of one entry returns that entry. This proves the recursive algorithm by induction on s.

Structural or analytic derivation

The output consists of three cosets of the smaller frequency lattice. The local three-point Fourier matrix combines them, with ω_3^t twiddle factors. Precomputing successive powers of ω_n by multiplication keeps the combine stage linear.

Comparative cost and preferred method

There are three subproblems of size $n/3$ and a cn combine cost. Thus $T(n) = 3T(n/3) + cn$, $T(1) = c_0$. Every one of the $\log_3 n$ nontrivial levels costs cn, so $T(n) = c_0 n + cn \log_3 n = \Theta(n \log n)$. The source's generalized FFT is already the preferable method for this transform family.

3.13 AC-063 | The chirp transform including the zero parameter

Reference: [3], exercise 30.2-8.

Source task and conventions

Compute $y_k = \sum_{j=0}^{n-1} a_j z^{kj}$ for arbitrary complex z in $O(n \log n)$ arithmetic.

Conventional solution and proof

For $z \neq 0$, choose either square root q with $q^2 = z$. The identity $k^2 + j^2 - (k-j)^2 = 2kj$ gives

$$y_k = q^{k^2} \sum_{j=0}^{n-1} (a_j q^{j^2}) q^{-(k-j)^2}.$$

Set $b_j = a_j q^{j^2}$ and $c_l = q^{-l^2}$ for $-(n-1) \leq l \leq n-1$. Choose a power-of-two $M \geq 2n-1$, zero-pad b to M, and store c_l at index $l \bmod M$. These indices are distinct on the required difference interval. An M-point circular convolution then supplies the indicated sum for $k = 0, \ldots, n-1$, since every difference $k - j$ lies in that interval. Multiply by q^{k^2} to finish. Changing the sign of q does not change the result because the combined exponent is even. For $z = 0$, interpret $z^0 = 1$ as in the polynomial definition. Then $y_0 = \sum_j a_j$ and $y_k = a_0$ for $k \geq 1$. This case costs $O(n)$ and avoids the invalid negative powers in the chirp formula.

Structural or analytic derivation

The phase is a quadratic form whose polarization converts the bilinear term kj into a difference convolution. All chirp powers can be prepared in $O(n)$ multiplications: starting from $q^0 = 1$, update $q^{(j+1)^2} = q^{j^2} q^{2j+1}$ and update the ratio by q^2. The inverse-power sequence is generated analogously. This is a complete Bluestein-type construction for arbitrary nonzero parameter, not only roots of unity.

Comparative cost and preferred method

Direct evaluation uses n^2 weighted terms. Two forward FFTs, one inverse and linear preparation yield $O(n \log n)$ arithmetic. For $|z| \neq 1$, chirp magnitudes can grow or decay exponentially in n^2, so precision, overflow and bit complexity require separate control. On the DFT specialization, optimized SciPy FFT is a stronger conventional baseline because it already uses Bluestein for poorly factorizable sizes (reference S1). The zero-parameter branch is essential to the word "any" in the question.

3.14 AC-064 | All derivatives at one point by one convolution

Reference: [3], exercise 30-3.

Source task and conventions

Solve every part of CLRS Problem 30-3 for a degree-below-n polynomial and a point x_0.

Conventional solution and proof

(a) If $A(x) = \sum_{j=0}^{n-1} b_j (x - x_0)^j$, differentiation gives $A^{(t)}(x_0) = t! b_t$. Successively update $t!$ and multiply each b_t, using $O(n)$ arithmetic. (b) The values $A(x_0 + \omega_n^k)$ are the DFT of b. One inverse FFT therefore recovers all b_t in $O(n \log n)$. (c) Expand each monomial by the binomial theorem:

$$A(x_0 + w) = \sum_{j=0}^{n-1} a_j \sum_{r=0}^{j} \binom{j}{r} x_0^{j-r} w^r = \sum_{r=0}^{n-1} \frac{w^r}{r!} \sum_{j=r}^{n-1} a_j j! \frac{x_0^{j-r}}{(j-r)!}.$$

Set $f(j) = a_j j!$ and $g(l) = x_0^{-l}/(-l)!$ for $-(n-1) \leq l \leq 0$, with $g(l) = 0$ for $l > 0$. At $w = \omega_n^k$ this is exactly the source's asserted convolution formula, with $g(r - j)$ zero for $j < r$. (d) Form $u_l = a_{n-1-l}(n - 1 - l)!$ and $v_l = x_0^l / l!$ for $0 \leq l < n$. Their linear convolution satisfies

$$(u * v)_{n-1-r} = \sum_{j=r}^{n-1} a_j j! x_0^{j-r}/(j-r)! = A^{(r)}(x_0).$$

One zero-padded FFT convolution thus gives all derivatives directly. Dividing them by $r!$ and taking one FFT supplies all shifted evaluations, also within $O(n \log n)$. The convention $x_0^0 = 1$ handles $x_0 = 0$ without dividing by x_0.

Structural or analytic derivation

The calculation is exact Taylor translation expressed as a weighted convolution. It unifies the first-order jet method with coefficient extraction of every order. Since A is a polynomial of degree below n, the circle sampling in part (b) has no aliasing; an arbitrary analytic function would additionally require a tail bound.

Comparative cost and preferred method

Repeated symbolic differentiation followed by evaluation or a triangular Taylor-shift table uses $\Theta(n^2)$ coefficient operations. The convolution method uses $O(n \log n)$ and $O(n)$ coefficient storage. It is the efficient method already developed by the source problem. Factorials and high derivatives can be ill-conditioned or overflow in floating arithmetic; the operation bound is over exact or suitably controlled coefficient arithmetic, not a blanket stability theorem.

3.15 AC-065 | Constructing a polynomial from its roots

Reference: [3], exercise 30.2-7.

Source task and conventions

Given n roots with multiplicities, compute coefficients of a degree-n polynomial with exactly those roots in $O(n \log^2 n)$ arithmetic.

Conventional solution and proof

Choose the monic polynomial $P(x) = \prod_{j=0}^{n-1}(x - z_j)$. Every listed value is a zero with its specified multiplicity, and over $\mathbb{C}$ a nonzero product vanishes only when one of its factors does, so there are no other zeros. Build a balanced product tree: leaves are the n linear factors; each internal node multiplies its two child polynomials by an FFT polynomial multiplication. Pad missing leaves with the constant polynomial one to reach a power of two. Induction up the tree proves that every node equals the product of its leaf factors, so the root is the desired P.

Structural or analytic derivation

At level l, each product has degree $O(2^l)$ and there are $O(n/2^l)$ products. With multiplication cost $M(d) = O(d \log d)$, the work at that level is $O(nl)$. Summing $l = 1, \ldots, \lceil \log_2 n \rceil$ gives $O(n \log^2 n)$. This explicitly proves the requested bound, including repeated roots.

Comparative cost and preferred method

Sequential multiplication by linear factors takes $\Theta(n^2)$ coefficient operations. The balanced tree has the proved $O(n \log^2 n)$ arithmetic cost; breadth-first construction can retain only adjacent levels with $O(n)$ coefficients. Repeated or nearby roots can make coefficient/root conversion numerically sensitive, so algebraic correctness and floating stability are separate.

3.16 AC-066 | Multipoint evaluation by a remainder tree

Reference: [3], exercise 30-4.

Source task and conventions

Solve all four parts of CLRS Problem 30-4, assuming degree-d polynomial remainders can be computed in $O(d \log d)$ arithmetic.

Conventional solution and proof

(a) Euclidean division by $x - z$ has the form $A = (x - z)Q + r$ with constant r. Substitution gives $r = A(z)$. (b) Hence $Q_{kk} = A(x_k)$. The polynomial $P_{0,n-1}$ has degree n, greater than $\deg A$, so its remainder is A itself. (c) Since P_{ik} divides P_{ij}, the difference $A - Q_{ij}$ is divisible by P_{ik}. Thus A and Q_{ij} have the same unique remainder modulo P_{ik}, proving $Q_{ik} = Q_{ij} \bmod P_{ik}$. The same argument applies to P_{kj}, including the shared endpoint in the source notation. (d) Build a balanced product tree for the factors $x - x_k$. At the root store A; at each node reduce its stored remainder modulo each child product. At a leaf the result is the constant $A(x_k)$ by (a). Induction down the tree, using (c), proves that each stored value equals the remainder of the original A modulo that node's product.

Structural or analytic derivation

The algorithm passes local residue classes down a divisibility hierarchy. In its implementation, child index ranges are disjoint, $[i, k]$ and $[k + 1, j]$; this is compatible with the source's more general divisibility identity. Repeated evaluation points do not invalidate the remainder argument, though repeated leaves return repeated values.

Comparative cost and preferred method

At level l, the total degree is $O(n)$ and the largest division size is $O(n/2^l)$. Under the source's remainder-cost assumption, that level costs $O(n \log(n/2^l))$. Summing the logarithmic levels yields $O(n \log^2 n)$; the product tree has the same bound. Separate Horner evaluations cost $\Theta(n^2)$. At roots of unity, the more specialized FFT costs only $O(n \log n)$ and remains preferable.

3.17 AC-067 | An invertible Fourier transform over a composite ring

Reference: [3], exercise 30.2-6.

Source task and conventions

For $n = 2^k \geq 2$, $m = 2^{tn/2} + 1$ and $\omega = 2^t \pmod m$, prove that the DFT and inverse are well defined even when m is composite.

Conventional solution and proof

The modulus is odd, so both two and n are units modulo m. Also $\omega^{n/2} = 2^{tn/2} = -1 \pmod m$, giving $\omega^n = 1$. Its order is exactly n: every proper divisor of the power-of-two n divides $n/2$, where the value is not one. The essential point is orthogonality without division by a possible zero divisor. For $0 < r < n$, write $r = 2^s u$ with odd u and $0 \leq s < k$. Put $L = n/2^s$. In each block of L terms of $\sum_{j=0}^{n-1} \omega^{rj}$, pair index j with $j + L/2$. The ratio is $\omega^{rL/2} = \omega^{un/2} = (-1)^u = -1$. Every pair cancels, so the sum is zero modulo m. For $r = 0$ the sum is n. Define $V_{kj} = \omega^{kj}$ and $W_{jk} = n^{-1}\omega^{-jk}$. The preceding sums give $(WV)_{jl} = \delta_{jl}$ and likewise $VW = I$. Thus both transforms are well-defined inverse linear maps over $\mathbb{Z}/m\mathbb{Z}$.

Structural or analytic derivation

This is a principal-root construction in a ring, not an argument that the ring is a field. The pair-cancellation proof supplies the exact property needed by the butterfly recursion. The usual geometric-series step dividing by $1 - \omega^r$ would be unjustified when that element is a zero divisor.

Comparative cost and preferred method

The radix-two algorithm uses $O(n \log n)$ ring operations and avoids floating roundoff. However m has $tn/2 + 1$ bits, so a ring operation is not a constant-cost machine-word operation as n grows. The next source problem supplies an alternative prime-modulus word model. The one-point DFT is trivially the identity; the printed modulus formula needs separate interpretation at $n = 1$ when $tn/2$ is nonintegral.

3.18 AC-068 | Prime-modulus Fourier transforms

Reference: [3], exercise 30-5.

Source task and conventions

Solve the heuristic prime-search, invertibility, FFT implementation and explicit modulus-17 calculation in CLRS Problem 30-5.

Conventional solution and proof

(a) Near $p = kn + 1$, prime-density heuristics assign probability on the order of $1/\log p$ to a candidate (with a constant adjustment for the fixed odd progression when $n \geq 2$ is a power of two). This suggests $\Theta(\log n)$ trials and a typical p of order $n \log n$. Its bit length is $\log_2 n + O(\log \log n) = \Theta(\log n)$. This is the requested heuristic, not a proved expectation for the first prime in this progression. (b) A generator g of the multiplicative group modulo p has order kn. Thus $w = g^k$ has order n. For $0 < r < n$, $1 - w^r \neq 0$ is invertible in the field, and the geometric sum $\sum_{j=0}^{n-1} w^{rj} = 0$. Also n is nonzero modulo p. These facts prove that $n^{-1}w^{-jk}$ is the inverse DFT matrix. (c) Apply the radix-two recursion with all additions and multiplications reduced modulo p. The combine identity is polynomial algebra and remains valid in this field. Use w^{-1} and one final multiplication by n^{-1} for the inverse. The recurrence $T(n) = 2T(n/2) + \Theta(n)$ gives $\Theta(n \log n)$ operations on the stipulated $O(\log n)$-bit words. (d) Modulo 17, $3^4 = 13$, $3^8 = 16 = -1$ and $3^{16} = 1$, so three has order 16. For $n = 8$, take $w = 3^2 = 9$ and $8^{-1} = 15$. The full transform of $(0, 5, 3, 7, 7, 2, 1, 6)$ is printed in the executable-results table; each entry is calculated by $\sum_{j=0}^{7} a_j 9^{jk} \pmod{17}$, and the inverse is checked exactly.

Structural or analytic derivation

Finite-field orthogonality gives an exact analogue of the complex Fourier basis. Integer convolution is recovered from modular coefficients only after a coefficient bound is supplied: one modulus does not distinguish arbitrary integers congruent modulo p. CRT or a sufficiently large prime provides the additional information.

Comparative cost and preferred method

Direct evaluation needs n^2 modular weighted terms; the FFT needs $\Theta(n \log n)$ word operations once p, w are given. Part (a)'s prime search and finding a generator are not included in that conditional cost. The modulus-17 example is an exact arithmetic check, not a benchmark of asymptotic speed.

k	DFT mod 17
0	14
1	10
2	10
3	4
4	8
5	11
6	13
7	15

3.19 AC-069 | The Vandermonde determinant

Reference: [3], exercise D-1.

Source task and conventions

Prove $\det V(x_0, \ldots, x_{n-1}) = \prod_{j<k}(x_k - x_j)$, where row i is $(1, x_i, \ldots, x_i^{n-1})$.

Conventional solution and proof

For $n = 1$ both sides equal one. For $n > 1$, working right to left for $j = n-1, \ldots, 1$, replace column j by column $j - x_0$ times column $j-1$; in conventional notation the operation is $C_j \leftarrow C_j - x_0 C_{j-1}$. Because the order is right to left, the subtracted column has not yet changed. These determinant-preserving operations make the first row $(1, 0, \ldots, 0)$. In row $i \geq 1$, column $j \geq 1$ is $x_i^j - x_0 x_i^{j-1} = (x_i - x_0)x_i^{j-1}$. Expand along the first row and factor $x_i - x_0$ from each remaining row. The residual determinant is $V(x_1, \ldots, x_{n-1})$. Induction gives

$$\det V = \prod_{i=1}^{n-1}(x_i - x_0) \prod_{1 \leq j < k < n}(x_k - x_j) = \prod_{0 \leq j < k < n}(x_k - x_j).$$

The proof remains valid when nodes coincide, where both sides vanish.

Structural or analytic derivation

The determinant is an alternating polynomial in the nodes and vanishes whenever two nodes coincide. The product of differences displays the complete zero divisor; the column proof above fixes the multiplicative constant and sign. Distinctness of the nodes is therefore exactly the invertibility condition for polynomial interpolation.

Comparative cost and preferred method

Evaluating the product uses $n(n-1)/2$ differences and one fewer product after initialization, with no $n \times n$ matrix formation. Generic elimination costs $\Theta(n^3)$ arithmetic and $\Theta(n^2)$ storage. This structured determinant formula proves an arithmetic reduction to $\Theta(n^2)$ for the requested scalar output. Near-colliding nodes can still cause numerical sensitivity.

3.20 AC-070 | Newton iteration: corrected local error theorem

Reference: [3], exercise 33-1.

Source task and conventions

Derive the Newton update, establish the correct local error relation and iteration bound, and compare Newton with gradient descent for $(x-3)^2$ from $x_0 = 3.5$.

Conventional solution and proof

(a) Setting the tangent-line equation to zero gives $x_{t+1} = x_t - f(x_t)/f'(x_t)$ whenever $f'(x_t) \neq 0$. (b) The printed Taylor and error formulas contain defects. Taylor's theorem with Lagrange remainder is

$$0 = f(x_*) = f(x_t) + f'(x_t)(x_* - x_t) + \tfrac{1}{2} f''(\xi_t)(x_* - x_t)^2.$$

Subtracting the Newton update gives the exact relation

$$\boxed{e_{t+1} = \frac{|f''(\xi_t)|}{2|f'(x_t)|}\, e_t^2, \qquad e_t = |x_t - x_*|.}$$

The derivative in the denominator is at x_t, and the error is squared. The displayed source formula with a first power of e_t is not the quadratic-convergence identity. (c) Choose a closed interval about x_* lying in the differentiability domain, with $|f'| \geq m > 0$ and $|f''| \leq M$, and set $C = M/(2m)$. If $Ce_0 < 1$ and the initial error ball lies in this interval, then $e_{t+1} \leq Ce_t^2 < e_t$, so every iterate remains inside the interval. Induction gives

$$e_t \leq C^{-1}(Ce_0)^{2^t}.$$

For $C > 0$ and a target smaller than the initial error, it suffices that

$$t \geq \left\lceil \log_2 \left(\frac{\log(1/(C\delta))}{\log(1/(Ce_0))} \right) \right\rceil,$$

with zero iterations when the initial error already meets the target. This is $O(\log\log(1/\delta))$ for fixed admissible C, e_0. If $M = 0$, the affine function is solved in one step. The condition $e_0 < 1$ alone does not imply $Ce_0 < 1$. (d) For $f(x) = (x-3)^2$, Newton gives $x_{t+1} - 3 = (x_t - 3)/2$ and $e_t = 2^{-(t+1)}$. Thus the exact tolerance count is $\max\{0, \lceil \log_2(1/(2\delta)) \rceil\}$. The root is multiple and violates the nonzero-derivative hypothesis at x_*. Gradient descent with step η gives $e_t = \frac{1}{2}|1-2\eta|^t$. It converges for $0 < \eta < 1$, has the same rate at $\eta = 1/4$, and reaches the minimizer in one step at $\eta = 1/2$. Without a specified step size there is no unique numerical comparison.

Structural or analytic derivation

The Newton map is a dynamical system. At a root of known multiplicity two, the modified map $x \mapsto x - 2f/f'$ reaches three in one step for this exact quadratic. The conventional gradient method with the exact quadratic step also reaches it in one step. A root certificate or multiplicity calculation can justify the modification, while a generic claim of quadratic convergence at a multiple root is false. The graph shows both linear Newton convergence and the one-step quadratic-model methods.

Comparative cost and preferred method

For the simple-root theorem, the proved iteration count is double-logarithmic in inverse tolerance within the invariant neighbourhood. For the specified double root, ordinary Newton is only logarithmic. Modified Newton and correctly chosen gradient descent tie at one step on this quadratic. These are exact iteration counts; each method's derivative, multiplicity and step-selection costs remain explicit.

Source qualification

The source page was inspected visually (physical pp.1060–1061). Equation (33.34) omits the factor 1/2; part (b) prints $f'(\gamma^{(t)})$ and a first power of the error. The corrected theorem and the invalid assumptions are stated explicitly, not attributed as an unchanged source proof.

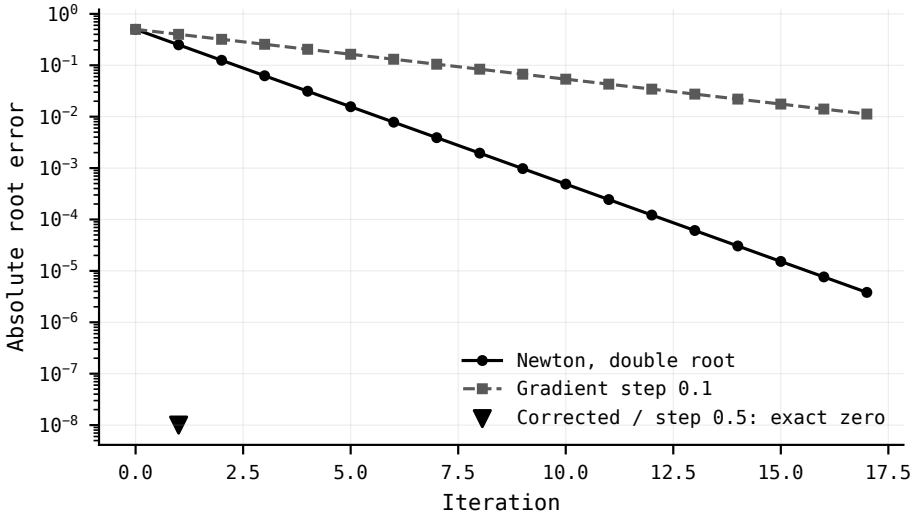


Figure 3.3: The square (x-3)^2 has a multiple root, so ordinary Newton is linear. The marker represents exact termination, not a nonzero error of 10^-8.

3.21 AC-071 | Horner evaluation with a loop invariant

Reference: [3], exercise 2-3.

Source task and conventions

Solve CLRS Problem 2-3: runtime, a naive evaluator and the full loop-invariant proof of Horner's rule.

Conventional solution and proof

(a) The printed loop performs $n+1$ iterations with constant scalar work, so its runtime is $\Theta(n)$. (b) A direct-from-scratch evaluator is:

```
p = 0
for k in range(n+1):
    power = 1
    for j in range(k):
        power *= x
    p += A[k] * power
return p
```

It performs $\sum_{k=0}^{n} k = n(n+1)/2$ power multiplications plus $n+1$ coefficient multiplications, hence $\Theta(n^2)$ arithmetic. (c) At the start of the descending iteration i, assert $p = \sum_{k=0}^{n-i-1} A[k+i+1]x^k$, with an empty sum interpreted as zero. Initially $i = n$ and $p = 0$, so the invariant holds. The update makes

$$A[i] + xp = A[i] + \sum_{k=1}^{n-i} A[k+i]x^k = \sum_{k=0}^{n-i} A[k+i]x^k,$$

which is exactly the invariant for the next iteration $i-1$. On termination the next index would be -1, giving $p = \sum_{k=0}^{n} A[k]x^k = P(x)$. Thus initialization, preservation and termination are all established.

Structural or analytic derivation

Initializing $p = A[n]$ and looping from $n-1$ down to zero removes the first multiplication by zero and uses exactly n multiplications and n additions. This is the nested affine-composition representation of the polynomial. A first-order jet carried through the same loop yields the value and derivative together with only constant overhead per coefficient.

Comparative cost and preferred method

The printed Horner procedure is $\Theta(n)$, compared with the requested naive-from-scratch $\Theta(n^2)$ evaluator. Reading arbitrary coefficients gives an $\Omega(n)$ input lower bound. A conventional power accumulator also achieves $\Theta(n)$, so Horner's asymptotic advantage is specific to the prescribed naive baseline; its smaller operation count and natural derivative extension remain useful.

3.22 AC-072 | Optimal bitonic Euclidean tours

Reference: [3], exercise 14-3.

Source task and conventions

Give a proved $O(n^2)$ algorithm for an optimal bitonic tour when all point x coordinates are distinct.

Conventional solution and proof

Sort the points as $p_1, \ldots, p_n$ by increasing x. Let $d(i, j)$ be Euclidean distance. For $i < j$, let $B[i, j]$ be the minimum total length of two increasing paths starting at p_1, covering exactly $p_1, \ldots, p_j$, sharing only p_1, and ending at p_i, p_j. The base is $B[1, 2] = d(1, 2)$. For $j \geq 3$,

$$B[i, j] = B[i, j-1] + d(j-1, j) \quad (i < j-1),$$

$$B[j-1, j] = \min_{k<j-1} \{B[k, j-1] + d(k, j)\}.$$

For the first case, the new rightmost point must extend the path whose endpoint was $j-1$; removing its last edge gives exactly the preceding state. In the second case it extends the other path, whose previous endpoint is some $k < j-1$; removing and then restoring that edge gives all and only the alternatives in the minimum. Induction on j therefore proves optimality of every state. The optimal closed length is $B[n-1, n] + d(n-1, n)$. Equivalently, minimize $B[i, n] + d(i, n)$ over $i < n$ and use the recurrence to obtain the same value. Store each minimizing k. Backtracking the recurrence adds the recorded edge and recursively visits the previous state; append the final closing edge to recover the tour. For two points the closed tour traverses the segment twice; one point has length zero.

Structural or analytic derivation

Identify planar points with complex numbers z_j. Distances become $|z_i - z_j|$, but the dynamic-programming state and proof are unchanged. The structural simplification is the bitonic ordering constraint, which reduces the search to two frontier endpoints. It does not produce an optimum for unrestricted Euclidean travelling-salesperson instances.

Comparative cost and preferred method

There are $\Theta(n^2)$ states. Nonadjacent-endpoint states take constant work; the adjacent-endpoint minima scan at most $j-2$ choices, whose total is $O(n^2)$. Sorting costs $O(n \log n)$ and is dominated. Storage is $O(n^2)$ with reconstruction pointers, or $O(n)$ for the length alone using suitable frontiers. Enumeration of all assignments to the two chains is exponential. The standard dynamic program is the preferable method; complex coordinates add no speedup.

3.23 AC-073 | Integer linear programming and duality gaps

Reference: [3], exercise 29-3.

Source task and conventions

Prove weak duality for integer programs, give a counterexample to strong duality, and prove $IP \leq P = D \leq ID$.

Conventional solution and proof

Use the source's standard primal $\max c^{\mathsf{T}}x$ subject to $Ax \leq b$, $x \geq 0$, and dual $\min b^{\mathsf{T}}y$ subject to $A^{\mathsf{T}}y \geq c$, $y \geq 0$. (a) For any feasible pair, including integer pairs,

$$c^{\mathsf{T}}x \leq (A^{\mathsf{T}}y)^{\mathsf{T}}x = y^{\mathsf{T}}Ax \leq y^{\mathsf{T}}b.$$

The first inequality uses $x \geq 0$, the second $y \geq 0$, proving weak duality without any continuity assumption. (b) Consider maximizing x subject to $2x \leq 1$ and $x \in \mathbb{Z}_{\geq 0}$. Its optimum is zero. The integer version of the corresponding dual minimizes y subject to $2y \geq 1$ and $y \in \mathbb{Z}_{\geq 0}$, whose optimum is one. Both optima are finite and feasible, so strong equality fails. (c) The integer primal feasible set is contained in the real primal set, so $IP \leq P$. The integer dual feasible set is contained in the real dual set and the objective is minimized, so $D \leq ID$. By the real linear-programming strong-duality theorem under the stated feasible, finite optimum assumptions, $P = D$. Combining these gives the claimed chain. In the example it reads $0 \leq 1/2 = 1/2 \leq 1$.

Structural or analytic derivation

Relaxation provides a certified upper bound on an integer maximization, and an integer dual feasible point gives a further upper certificate. Neither algebraic manipulation nor holomorphic continuation restores the missing integrality. The explicit one-variable example isolates the obstruction with no computational experiment.

Comparative cost and preferred method

Weak-duality verification is matrix–vector arithmetic proportional to the nonzero data and gives a certificate without solving the integer program. It does not imply a polynomial-time exact integer optimizer. The native linear and integer optimization methods are retained; there is no justified complex-analytic improvement for the general task.

Chapter 4

Database System Concepts

Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Reference problems and derived analyses.

4.1 AC-074 | Counting all oriented labelled join orders

Reference: [4], exercise 16.12.

Source task and conventions

Prove that n distinct relations have $(2n-2)!/(n-1)!$ binary join orders under the source's ordered-tree convention.

Conventional solution and proof

A binary join expression is a rooted ordered full binary tree with n leaves, together with a bijection between its leaves and the n relations. Such a tree has $n-1$ internal nodes: counting edges by parents gives $2I$, while counting a tree's vertices gives $I+n-1$, hence $I=n-1$. The supplied Catalan formula gives $C_{n-1}=\frac{1}{n}\binom{2n-2}{n-1}$ shapes. Every shape admits $n!$ distinct leaf labellings. Their product is

$$n!C_{n-1}=\frac{n!}{n}\frac{(2n-2)!}{((n-1)!)^2}=\boxed{\frac{(2n-2)!}{(n-1)!}}.$$

The tree shape and leaf labelling are recovered uniquely from the expression, proving that neither orders nor labels are overcounted within this convention.

Structural or analytic derivation

Writing J_n for the result, factorial cancellation gives $J_1=1$ and $J_n=2(2n-3)J_{n-1}$. The generating-function Catalan derivation is proved in the binary-tree question and supplies the same count. Commutativity of a relational join does not identify physical left/right plan choices under the source's ordered-plan convention. If one deliberately identifies swapped children, the counting problem changes.

Comparative cost and preferred method

Enumerating plans has an unavoidable $\Omega(J_n)$ output count. Computing only their number needs $n-1$ multiplicative updates and $O(1)$ stored big integers. This is an exponential reduction in enumerated objects, not an algorithm for finding the cheapest physical plan or proving SQL semantic equivalence. Against direct use of the source's Catalan hint, the analytic derivation supplies no further asymptotic gain.

Chapter 5

Natural Language Processing with Python

Steven Bird, Ewan Klein, Edward Loper. Reference problems and derived analyses.

5.1 AC-075 | Recursive and dynamic Catalan computation

Reference: [5], exercise 4.26.

Source task and conventions

Implement recursive and dynamic-programming functions for $C_0 = 1$, $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$, and compare them with `timeit`.

Conventional solution and proof

The two required implementations are:

```
def catalan_recursive(n):
    if n < 0: raise ValueError("n must be nonnegative")
    if n == 0: return 1
    return sum(catalan_recursive(i) *
               catalan_recursive(n-1-i) for i in range(n))

def catalan_dp(n):
    if n < 0: raise ValueError("n must be nonnegative")
    c = [1] + [0] * n
    for k in range(1, n+1):
        c[k] = sum(c[i] * c[k-1-i] for i in range(k))
    return c[n]
```

Strong induction proves the recursive function: every recursive argument is smaller, and the returned sum is the defining recurrence. For the dynamic function, before iteration k, all stored entries below k equal the corresponding Catalan numbers; the update establishes entry k, preserving the invariant through termination. The code therefore returns the same exact integer.

Structural or analytic derivation

The factorial identity proved in the binary-tree problem gives the additional exact update

```
def catalan_multiplicative(n):
    if n < 0: raise ValueError("n must be nonnegative")
    c = 1
    for k in range(n):
        c = c * (4*k + 2) // (k + 2)

    return c
```

The integer division is exact at every step by that identity. The executable study compares all three, including the two functions specifically requested by the source. It uses separate `timeit.Timer.repeat` measurements, records repetitions and verifies equality before timing.

Comparative cost and preferred method

Counting each recursive invocation gives $T_0 = 1$ and $T_n = 1 + 2\sum_{i<n} T_i$, hence $T_n = 3^n$. The dynamic algorithm performs $n(n+1)/2$ products and stores $n+1$ integers. The multiplicative algorithm performs n updates and stores one Catalan value. These are exact counts in the indicated models. Timing tables report actual measurements separately and include the recursive setup; integer bit growth is retained as a limitation of unit-cost counting.

Chapter 6

Basics of Compiler Design

Torben Aegidius Mogensen. Reference problems and derived analyses.

6.1 AC-076 | Decimal languages with leading zeros

Reference: [6], exercise 2.1.

Source task and conventions

Construct regular expressions for nonempty decimal strings representing exactly 42, not 42, and a value greater than 42.

Conventional solution and proof

Let $D = [0\text{-}9]$ and $N = [1\text{-}9]$ be alphabet shorthands. The three expressions are

$$(a)\quad 0^*42,$$
$$(b)\quad 0^*\big(0 \mid N \mid [1\text{-}3]D \mid 4[013\text{-}9] \mid [5\text{-}9]D \mid NDDD^*\big),$$
$$(c)\quad 0^*\big(4[3\text{-}9] \mid [5\text{-}9]D \mid NDDD^*\big).$$

In $NDDD^*$, the two explicit D symbols after N impose at least three significant digits. To prove the classification, remove all leading zeros from a nonzero string. One remaining digit has value 1–9. Two remaining digits represent 10–99 and the displayed alternatives partition these values, excluding 42 in (b) and keeping 43–99 in (c). At least three significant digits represent a value at least 100. An all-zero nonempty string is supplied by 0^*0 in (b). Every case is disjoint by significant length or first digit, and no expression accepts the empty string.

Structural or analytic derivation

A finite-state comparison automaton can store whether the significant prefix has length zero, one, two or at least three and, at length two, whether its value lies below, at or above 42. This gives a finite-state proof of regularity. The expressions above are preferable for the source's requested output because they directly enumerate the decisive cases.

Comparative cost and preferred method

Scanning a length-m decimal string takes $\Theta(m)$ character operations and constant state storage; each input symbol must in general be inspected. Parsing the entire integer creates a potentially $\Theta(m)$-digit arithmetic object with unnecessary bit cost. The structural finite-state comparison avoids that object, but it is also a standard lexer method. No complex-analytic method improves the regular-expression construction.

6.2 AC-077 | NFA and subset DFA for a starred prefix

Reference: [6], exercise 2.2.

Source task and conventions

Construct the NFA for $a^*(a \mid b)aa$ and convert it by the source's subset algorithm.

Conventional solution and proof

State	Outgoing labelled transitions
0	none
1	$a\to 0$
2	$a\to 1$
3	$\varepsilon\to 5$, $\varepsilon\to 6$
4	$\varepsilon\to 2$
5	$a\to 4$
6	$b\to 4$
7	$\varepsilon\to 3$, $\varepsilon\to 8$
8	$a\to 7$

The NFA starts at state 7 and accepts only at state 0.

DFA	NFA subset	a	b	Accepts
D_0	$\{3,5,6,7,8\}$	D_1	D_2	no
D_1	$\{2,3,4,5,6,7,8\}$	D_3	D_2	no
D_2	$\{2,4\}$	D_4	D_5	no
D_3	$\{1,2,3,4,5,6,7,8\}$	D_6	D_2	no
D_4	$\{1\}$	D_7	D_5	no
D_5	$\varnothing$	D_5	D_5	no
D_6	$\{0,1,2,3,4,5,6,7,8\}$	D_6	D_2	yes
D_7	$\{0\}$	D_5	D_5	yes

The NFA uses the source's literal, union, concatenation and star fragments, with shared continuation endpoints. A literal adds a labelled edge; a union branches by epsilon and rejoins; concatenation shares its continuation; a star has epsilon exit and epsilon entry into a body that returns to the star state. Induction on these four constructors proves that each fragment accepts exactly its expression's language. For the DFA, D_0 is the epsilon closure of the NFA start, and each table transition is

$$\delta_D(S,c) = \varepsilon\text{-closure}\{t : \exists s \in S,\ s \xrightarrow{c} t\}.$$

Induction on input length proves that the reached subset is exactly the set of reachable NFA states after that input. It accepts precisely when it contains the NFA final state. Thus both requested conversions are fully specified and proved equivalent. The empty subset is included to give a total transition function.

Structural or analytic derivation

The language consists of a^n with $n \geq 3$ and $a^k baa$ with $k \geq 0$. This gives an independent semantic description. Four states count a pure-a prefix up to three; three states record that a b has occurred and that zero, one or two required final a symbols have followed; one dead state handles any forbidden continuation. These eight states correspond to the eight DFA rows. Distinct pure-prefix states are separated by the number of a symbols needed to reach length three, except that appending a b and then aa distinguishes a pure prefix from a post-b state. Among post-b states, suffixes aa, a, ε distinguish the three remaining lengths. The dead state accepts no suffix. This supplies a distinguishing-suffix proof of minimality.

Comparative cost and preferred method

The fragment NFA has nine states; the total subset DFA has eight. Simulation of the DFA requires one transition per input symbol, hence $\Theta(m)$ time and constant state memory for a length-m string. A generic subset construction can take exponential preprocessing, so its advantage is not free. Counting words through a matrix resolvent is an optional different output; it does not replace the required automata.

6.3 AC-078 | A block-language NFA and DFA

Reference: [6], exercise 2.3.

Source task and conventions

Construct the NFA and subset DFA for $((a \mid b)(a \mid bb))^*$.

Conventional solution and proof

State	Outgoing labelled transitions
0	none
1	$\varepsilon\to 0$, $\varepsilon\to 7$
2	$\varepsilon\to 4$, $\varepsilon\to 6$
3	$\varepsilon\to 1$
4	$a\to 3$
5	$b\to 3$
6	$b\to 5$
7	$\varepsilon\to 9$, $\varepsilon\to 10$
8	$\varepsilon\to 2$
9	$a\to 8$
10	$b\to 8$

The NFA starts at state 1 and accepts only at state 0.

DFA	NFA subset	a	b	Accepts

D_0	$\{0,1,7,9,10\}$	D_1	D_1	yes
D_1	$\{2,4,6,8\}$	D_2	D_3	no
D_2	$\{0,1,3,7,9,10\}$	D_1	D_1	yes
D_3	$\{5\}$	D_4	D_2	no
D_4	$\varnothing$	D_4	D_4	no

The expression-fragment construction and its structural-induction proof are established in AC-077. Applied to the present block expression, they give the displayed NFA. Its subset transition is

$$\delta_D(S,c) = \varepsilon\text{-closure}\{t : \exists s \in S,\ s \xrightarrow{c} t\}.$$

The reachable-subset induction in AC-077 proves the language equivalence of this table, including its empty dead state. The table here specifies the states of the present expression.

Structural or analytic derivation

The accepted blocks are aa, ba, abb, bbb. In the subset table, D_0 and D_2 have the same acceptance and outgoing transitions, so they merge. A four-state minimal DFA has boundary state R (initial and accepting), state S after the first symbol of a block, state T after the first b in the second component, and dead state X:

	a	b
R	S	S
S	R	T
T	X	R
X	X	X

.

Parsing from a boundary consumes one arbitrary symbol, then either a or exactly bb, and returns to the boundary, proving equivalence. The empty suffix separates R from all others; suffix a separates S from T, X; suffix b separates T from X. Hence all four states are pairwise distinguishable and minimal.

Comparative cost and preferred method

The source-fragment NFA has 11 states, the unminimized total subset DFA has five, and minimization gives four: a 20% reduction in DFA state and two-symbol transition-table count. Recognition remains $\Theta(m)$ under both deterministic representations. No wall-clock gain is inferred from a one-state difference.

6.4 AC-079 | Counting and modular-counting automata

Reference: [6], exercise 2.7.

Source task and conventions

Construct DFAs for exactly three b symbols, a multiple-of-three b count, and a difference of a and b counts divisible by three.

Conventional solution and proof

For (a), use states $0, 1, 2, 3, \bot$, start zero and accept only three. An a leaves the state unchanged. A b sends $0 \to 1 \to 2 \to 3 \to \bot$, and $\bot$ loops. Induction on prefix length shows that states zero through three equal the b count and $\bot$ means at least four. For (b), use states $0, 1, 2$, start/accept zero, with $\delta(r, a) = r$ and $\delta(r, b) = r + 1 \pmod 3$. The invariant is the b count modulo three. For (c), use the same start and accepting state with $\delta(r, a) = r + 1 \pmod 3$ and $\delta(r, b) = r - 1 \pmod 3$. Its invariant is $\#a - \#b \pmod 3$. Each invariant begins correctly at the empty word and is preserved by the specified transition, proving all three languages.

Structural or analytic derivation

The modular DFAs are actions of the cyclic group $\mathbb{Z}/3\mathbb{Z}$. Their transition matrices are permutation matrices; a roots-of-unity diagonalization can count accepted words of a fixed length. The group invariant itself is already the simplest recognition proof. For exact-three counting, a truncated nonnegative counter with overflow is the appropriate finite-state object.

Comparative cost and preferred method

The state counts are five, three and three. They are minimal: differing exact counts are separated by a suffix supplying the missing b symbols, and differing modular residues are separated by a suffix completing one residue to zero. Recognition uses exactly one transition per character and $O(1)$ state memory. Computing both integer counts and subtracting is unnecessary storage for the modular tasks; standard remainder automata already attain the bound.

6.5 AC-080 | Minimal binary divisibility automata

Reference: [6], exercise 2.9.

Source task and conventions

Construct nonempty binary divisibility DFAs for 4 and 5 and prove the minimal state count for a general positive modulus $n = a2^b$ with odd a.

Conventional solution and proof

After a prefix of numeric value v, appending bit c gives $2v + c$. Thus the standard remainder transition is $r \mapsto 2r + c \pmod n$. For four, merge odd remainders and use states Z (zero, accepting), O (odd) and T

(remainder two). Start at T to reject the empty word:

	0	1
Z	Z	O
O	T	O
T	Z	O

.

The start has the same nonempty continuation behaviour as remainder two, so leading zeros and the one-symbol word zero are handled correctly. For five, use remainders $0, \ldots, 4$, accept zero, and a separate nonaccepting start S:

	0	1
S	0	1
0	0	1
1	2	3
2	4	0
3	1	2
4	3	4

.

The prefix invariant proves correctness of these tables.

Structural or analytic derivation

Here is a full minimality proof for the general case. From remainder r, a suffix of length k and value t, $0 \le t < 2^k$, is accepted exactly when $n \mid 2^k r + t$. For $k \ge b$, this condition depends on r only through $r \bmod a$. For $k < b$, divisibility by 2^k forces $t = 0$, after which the condition is $a2^{b-k} \mid r$. Consequently all remainders in the same nonzero residue class modulo a are equivalent. Remainders $r = aj$ split into $b + 1$ classes according to $d = b - \min(b, v_2(j))$, with $d = 0$ for $j = 0$: precisely d trailing zero bits are needed for an immediate accepting zero suffix. There are $(a - 1) + (b + 1) = a + b$ classes. Different residues modulo a are distinguishable: choose $k \ge b$ so large that $2^k \ge n$ and choose the unique $t \in [0, n)$ satisfying $2^k r + t \equiv 0 \pmod{n}$. The same suffix fails for a different residue because a is odd. Among the zero-residue classes, the suffix 0^{d_1} distinguishes classes with $d_1 < d_2$. Every class is reachable by a nonempty binary representation of a representative remainder, allowing the word zero for remainder zero. If $b \ge 1$, the nonaccepting initial residual is equivalent to remainder $n/2$: after the first bit both have the same remainders as a zero-valued prefix, and both reject the empty suffix. No extra state is needed. If $b = 0$, the initial residual is distinct from accepting remainder zero and from every nonzero remainder: a sufficiently long all-zero suffix is accepted from the start but not from any nonzero remainder, since n is odd. Thus

$$\boxed{\text{minimum states} = \begin{cases} a + 1, & b = 0, \\ a + b, & b \ge 1. \end{cases}}$$

This includes $n = 1$: two states recognize all nonempty binary words.

Comparative cost and preferred method

A literal remainder DFA with a separate start has $n + 1$ states. The proved quotient uses $a + b$ states for even n, a ratio $(n + 1)/(a + b)$ in a complete two-symbol transition table, and $n + 1$ for odd n. Recognition remains linear in input length, while storage is reduced. This is an algebraic finite-state optimum certified by distinguishable residual languages; complex analysis is unnecessary.

6.6 AC-081 | The empty regular language and dead states

Reference: [6], exercise 2.10.

Source task and conventions

Prove the four laws for the empty-language expression ϕ, extend the NFA construction, and explain the minimization consequence.

Conventional solution and proof

Since $L(\phi) = \varnothing$, set union gives $L(\phi \mid s) = \varnothing \cup L(s) = L(s)$. Concatenating with the empty set produces no pairs of words, so $L(\phi s) = L(s\phi) = \varnothing$. The star is the union of all finite concatenation powers, including the zeroth power $\{\varepsilon\}$; every positive power of the empty language is empty. Hence $L(\phi^*) = \{\varepsilon\}$. For a fragment representing ϕ, create an entry and an exit with no path between them. It accepts no string, and can be inserted into the same union, concatenation and star constructions. The complete NFA for the empty language can be a single nonaccepting start state with no transitions.

Structural or analytic derivation

The extension admits dead states: states from which no accepting state is reachable. In minimization of a partial DFA, a missing transition and a transition to a dead state represent the same rejecting continuation, so treating them as different can prevent legitimate mergers. Either remove dead states before partial-DFA minimization, or complete every missing transition to one rejecting sink and apply the usual total-DFA equivalence refinement. All dead states then have identical empty right language and merge.

Comparative cost and preferred method

Coaccessibility can be found by reverse graph search from accepting states in $O(|Q|+|E|)$ time. Removing dead states or adding one sink has linear or constant-state overhead. This preprocessing restores the assumptions of the conventional minimizer. No analytic substitute is more efficient for the source's language-semantic requirement.

6.7 AC-082 | Four closure constructions for regular languages

Reference: [6], exercise 2.11.

Source task and conventions

Construct NFAs for prefixes, suffixes, scattered subsequences and reversals of a regular language.

Conventional solution and proof

Let $N = (Q, \Sigma, E, s, F)$ be an NFA, possibly with epsilon transitions. *Prefixes:* keep the start and transitions and make every state from which some final state is reachable accepting. A path labelled u reaches such a state exactly when some continuation v gives $uv \in L$. *Suffixes:* add a fresh start with epsilon edges to every state reachable from s, keep the original accepting states. A new accepting path labelled v begins at a state reached by some prefix u, hence $uv \in L$; every such factorization yields a path. *Scattered subsequences:* for every labelled edge $q \xrightarrow{c} r$, add an epsilon edge $q \to r$. An accepting new path can skip any selected original symbols. Reinsert each skipped edge label to obtain a word of L, proving one direction; delete unwanted symbols along any old accepting path to prove the other. *Reversal:* reverse all edges (preserving their labels), make the old start the sole final state, and add a fresh start with epsilon edges to the old final states. Reversing an accepting path reverses its label, and reversing it back proves the converse. All constructions are finite NFAs, establishing regularity for all four operations.

Structural or analytic derivation

These are reachability and path transformations on a finite labelled graph. The prefix operation changes a terminal observation; the suffix operation changes the admissible initial states; subsequence closure adds silent edges; reversal transposes labelled adjacency. Counting accepted words through a resolvent can be added later, but epsilon ambiguity must first be resolved to avoid counting paths as distinct words.

Comparative cost and preferred method

Reachability and coaccessibility take $O(|Q|+|E|)$ graph work. Reversal and epsilon-edge insertion have $O(|E|)$ output size and construction work; the number of states grows by at most one for the stated constructions. These bounds match the need to inspect or output the graph. Determinizing the results can be exponential and is not required by the exercise.

Source qualification

The source defines subsequence by deleting symbols from anywhere, not only by taking a contiguous substring (physical pp.57–58).

6.8 AC-083 | Three false closure claims

Reference: [6], exercise 2.12.

Source task and conventions

Decide whether arbitrary subsets, arbitrary supersets, and anagram closures of regular languages are regular.

Conventional solution and proof

All three assertions are false over an alphabet containing a, b. (a) The nonregular language $K = \{a^n b^n : n \geq 0\}$ is a subset of the regular language a^*b^*. (b) The same K is a superset of the regular language $\{\varepsilon\}$. To prove nonregularity, suppose K had pumping length p. Every decomposition $a^p b^p = xyz$ with $|xy| \leq p$ and $|y| > 0$ has $y = a^k$ with $k > 0$. Pumping y zero times gives $a^{p-k}b^p \notin K$, a contradiction. (c) Let $L = (ab)^*$. Its anagrams are exactly the words with equally many a and b symbols. If this anagram language were regular, its intersection with a^*b^* would be K and would be regular. Intersection closure follows from a product DFA that tracks both states and accepts precisely when both components accept. This contradicts the proof above.

Structural or analytic derivation

The anagram operation preserves the commutative count vector (the Parikh vector) but discards order. A finite automaton recognizing $(ab)^*$ does not thereby yield a finite automaton testing equality of two unbounded counts. The invariant identifies the obstruction and the regular intersection isolates it rigorously.

Comparative cost and preferred method

The counterexamples settle universal closure assertions with finite proofs; exhaustive testing of words through any finite length cannot establish them. The conventional pumping and product-automaton proof is the appropriate efficient method. There is no valid complex-analytic acceleration of this semantic classification.

Chapter 7

Introduction to Information Retrieval

Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze. Reference problems and derived analyses.

7.1 AC-084 | Transition matrix of the three-state figure

Reference: [7], exercise 21.5.

Source task and conventions

Read Figure 21.2 in the declared state order (A, B, C).

Conventional solution and proof

The figure's centre state is A. Its outgoing probabilities to B and C are both $1/2$; each of B, C returns to A with probability one. With rows as current states and columns as next states,

$$P = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Every row sums to one, and each positive entry corresponds to its labelled directed arrow. The stationary row vector is $(1/2, 1/4, 1/4)$: the B, C balance equations each give one half of the A mass, and normalization fixes it. The chain has period two, so a stationary distribution does not imply ordinary convergence from every starting state.

Structural or analytic derivation

The row-stochastic convention gives column-distribution evolution $p_{t+1} = P^{\top} p_t$. Stating this convention prevents transposition errors when the figure is converted into a matrix.

Comparative cost and preferred method

Four nonzero transitions suffice; a graph-based update takes four weighted contributions. A dense 3×3 update is also trivial. This question's optimal solution is direct transcription plus row-sum checks; no complex-analytic acceleration is warranted.

7.2 AC-085 | Teleportation matrices at three probabilities

Reference: [7], exercise 21.6.

Source task and conventions

Compute the exact matrices for $\alpha = 0, 1/2, 1$.

Conventional solution and proof

In node order $(1, 2, 3)$, the link walk is

$$W = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}.$$

Uniform teleportation gives $P_\alpha = (1-\alpha)W + (\alpha/3)\mathbf{1}\mathbf{1}^\top$. Thus

$$P_0 = W, \qquad P_{1/2} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}, \qquad P_1 = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

These matrices are nonnegative and every row sums to one. The zero-teleport matrix is periodic; the other two have positive diagonal and positive communication between all nodes.

Structural or analytic derivation

The teleport correction is rank one. Its action on a probability vector is simply addition of $\alpha/3$ to each output coordinate, so it need not be stored as a dense matrix in a large graph.

Comparative cost and preferred method

For a graph with E links, one transition-vector product costs $O(E + N)$ using the link representation and rank-one teleportation, instead of $O(N^2)$ dense work. For this three-node question the exact fractions are the principal deliverable and direct computation is preferable.

7.3 AC-086 | A Markov state for browser backtracking

Reference: [7], exercise 21.7.

Source task and conventions

Determine a valid state model for one and repeated back-button operations.

Conventional solution and proof

The current page alone is generally not a Markov state. Two histories ending at the same page x, one arriving from a and one from $b \neq a$, have different next-page distributions when the back action has positive probability. A state (previous page, current page) records enough information for one backward step but not arbitrary repeated backtracking. For repeated back actions, take the state to be the entire retained browser stack $(v_0, \ldots, v_k)$, with current page v_k. A hyperlink action appends its destination; a back action removes v_k when $k > 0$, and a declared rule handles an empty history, for example a self-transition. Assign action probabilities and hyperlink-choice probabilities as functions of this state. The next-stack distribution then depends only on the present stack, proving the Markov property. With an unbounded history the chain has a countable state space even on a finite web. With a history cap L, the state space is finite, bounded by $\sum_{j=1}^{L+1} N^j$ before restricting to valid link paths. A model including a forward button needs the stored forward history and its current index as well. Human action probabilities depending on unmodelled information are not made Markov merely by naming the process a chain.

Structural or analytic derivation

State augmentation converts a process with memory into a Markov process by retaining the information needed to determine the transition law. It is an exact representation of the specified stack policy, not an assertion that a real user follows stationary transition probabilities.

Comparative cost and preferred method

A stack implementation uses $O(k)$ storage and $O(1)$ push/pop cost per action. Enumerating every possible augmented state can be exponential in the cap. The gain is correct modelling; a page-only matrix cannot represent repeated back actions faithfully.

7.4 AC-087 | Ergodicity of a parameterized three-state chain

Reference: [7], exercise 21.8.

Source task and conventions

Determine every admissible value of p_A.

Conventional solution and proof

For $0 < p_A < 1$, all three states communicate: $A \to B$, $B \to C$, $C \to A$ have positive probability. Returns to A of length two (A, B, A) and three (A, B, C, A) both have positive probability. Their greatest common divisor is one, so the irreducible finite chain is aperiodic and hence ergodic. At $p_A = 0$, the chain is the deterministic three-cycle $A \to B \to C \to A$ and has period three. At $p_A = 1$, C is transient and the closed A, B class alternates with period two; the full chain is also reducible. Thus, under the source's irreducible-and-aperiodic meaning,

$$\boxed{0 < p_A < 1.}$$

The stationary distribution for an interior parameter is $\pi_A = \pi_B = 1/(3 - p_A)$ and $\pi_C = (1 - p_A)/(3 - p_A)$, obtained directly from the balance equations.

Structural or analytic derivation

Cycle lengths identify aperiodicity without computing eigenvalues. The stationary formula supplies a separate normalization and balance check but cannot replace the period argument at the endpoints.

Comparative cost and preferred method

A finite graph-reachability and cycle-gcd test proves the classification exactly. Numerical simulation near $p_A = 0$ or 1 can mix slowly and would not establish the endpoint distinction. The standard graph method is already the appropriate efficient method.

7.5 AC-088 | Full-support teleportation and ergodicity

Reference: [7], exercise 21.9.

Source task and conventions

Prove the claim for the source's uniform teleport operation and state the essential assumptions.

Conventional solution and proof

Let W be a row-stochastic link matrix, with a probability row assigned to every dangling vertex. For $0 < \alpha \leq 1$, uniform teleportation gives $P = (1 - \alpha)W + \alpha \mathbf{1}\mathbf{1}^\top / N$, so $P_{ij} \geq \alpha/N > 0$. Every state reaches every other in one step and every state has a self-loop. Therefore the finite chain is irreducible and aperiodic. This proves ergodicity for every directed link graph, including a graph with no edges after dangling rows are completed. A separate quantitative proof uses distributions u, v:

$$\|P^\top u - P^\top v\|_1 = (1 - \alpha)\|W^\top (u - v)\|_1 \leq (1 - \alpha)\|u - v\|_1.$$

The rank-one term cancels because both masses are one. Iteration on the compact probability simplex converges to a unique fixed point, with distance at most $2(1-\alpha)^t$ from an arbitrary initial distribution. At $\alpha = 0$ the theorem fails, for example on a directed cycle. Teleportation with restricted support also need not make the entire graph irreducible.

Structural or analytic derivation

The same operator has a convergent Neumann resolvent $[I-(1-\alpha)W^\top]^{-1} = \sum_{k\geq 0}(1-\alpha)^k(W^\top)^k$, whose norm is at most $1/\alpha$. This turns the qualitative graph theorem into an error bound for computation.

Comparative cost and preferred method

To reach ℓ_1 error ε from a probability initialization, $t \geq \lceil \log(2/\varepsilon)/[-\log(1-\alpha)] \rceil$ iterations suffice when $0 < \alpha < 1$; $\alpha = 1$ needs one step. A sparse step costs $O(E+N)$. This is a proved upper bound, not an exact mixing time for every graph.

7.6 AC-089 | Uniform lower bounds and the collapse of PageRank

Reference: [7], exercise 21.10.

Source task and conventions

Prove the pagewise bound and quantify the $\alpha \to 1$ limit.

Conventional solution and proof

Stationarity gives $\pi = (1-\alpha)W^\top\pi + \alpha u$, $u = \mathbf{1}/N$. Each component of the first term is nonnegative, so $\pi_i \geq \alpha/N$. Since the other $N-1$ components have the same lower bound,

$$\pi_i \leq 1 - (N-1)\alpha/N = \alpha/N + 1 - \alpha.$$

Hence $|\pi_i - \pi_j| \leq 1-\alpha$ for every pair. More precisely, $\pi = \alpha u + (1-\alpha)w$ for the probability vector $w = W^\top\pi$, yielding

$$\|\pi - u\|_\infty \leq (1-\alpha)(1-1/N), \qquad \|\pi - u\|_1 \leq 2(1-\alpha)(1-1/N).$$

Thus all scores tend to $1/N$ and pairwise contrasts vanish at least linearly. The elementary probability-simplex bounds prove all three inequalities.

Structural or analytic derivation

Teleportation is an explicit convex mixture. This representation yields parameter-uniform bounds without computing the particular graph's stationary vector.

Comparative cost and preferred method

The score bounds require $O(1)$ scalar work once N, α are given, whereas solving all N scores requires graph access. The simpler output is a bound, not the complete ranking. Near-uniform scores do not make their exact relative ordering automatically irrelevant or numerically certified.

7.7 AC-090 | Reproducing the seven-page PageRank example

Reference: [7], exercise 21.11.

Source task and conventions

Compute the source example using its actual graph and teleport rate .14.

Conventional solution and proof

The ordered adjacency lists read from Figure 21.4 are

$$0:[2],\quad 1:[1,2],\quad 2:[0,2,3],\quad 3:[3,4],\quad 4:[6],\quad 5:[5,6],\quad 6:[3,4,6].$$

Let $W_{ij}=1/d_i$ on these outgoing links and zero elsewhere; then $P=(43/50)W+(1/50)\mathbf{1}\mathbf{1}^\top$. In particular its three-outlink entries are $23/75$, not exactly .31. Solve $[I-(43/50)W^\top]\pi=(1/50)\mathbf{1}$. Exact rational elimination gives

$$\pi=\left(\frac{10399}{199557},\frac{2}{57},\frac{7451}{66519},\frac{120049}{488775},\frac{730688299}{3422402550},\frac{2}{57},\frac{349755251}{1140800850}\right)^\top.$$

Numerically this is

$$(.05211042,.03508772,.11201311,.24561199,.21350156,.03508772,.30658747)^\top.$$

It sums exactly to one and satisfies the exact stationary equations. Rounding each component to two decimal places reproduces $(.05,.04,.11,.25,.21,.04,.31)$. The rounded printed vector sums to 1.01, and two rounded printed transition rows also sum to 1.01; these are display-rounding effects, not a stochastic model to use unchanged.

```
import numpy as np
adj = [[2], [1,2], [0,2,3], [3,4], [6], [5,6], [3,4,6]]
W = np.zeros((7,7))
for i, js in enumerate(adj):
    W[i, js] = 1 / len(js)
pi = np.linalg.solve(np.eye(7) - .86*W.T, np.ones(7)*.02)
assert np.allclose(pi.sum(), 1)
assert np.allclose(pi, .86*W.T@pi + .02)
```

Structural or analytic derivation

The resolvent system gives the exact target without taking a long stochastic trajectory. Sparse fixed-point iteration is an alternative; an approximate vector with residual $r=\widehat{\pi}-.86W^\top\widehat{\pi}-.02\mathbf{1}$ satisfies $\|\widehat{\pi}-\pi\|_1\le\|r\|_1/.14$.

Comparative cost and preferred method

For seven nodes, direct elimination is inexpensive and the strongest simple baseline. On a large sparse graph, factorization fill-in can make iterative sparse products preferable. The exact rational calculation and residual bound certify the reproduced digits; no improvement is inferred from using the word resolvent.

7.8 AC-091 | Sequential-disk PageRank

Reference: [7], exercise 21.12.

Source task and conventions

Give a correct streaming algorithm, its stopping certificate and its memory assumptions.

Conventional solution and proof

First consider the usual external-graph model in which two N-entry rank vectors fit in memory but the E links do not. Keep the previous vector p fixed during a pass. Initialize $s = 0$ and dangling mass $d = 0$. Read each adjacency list once in its stored order. For a nonempty list of node i, add $p_i/\deg^+(i)$ to s_j for each outgoing neighbour j; for an empty list add p_i to d. Under uniform dangling completion set

$$p_{\text{new}} = (1-\alpha)(s + d\mathbf{1}/N) + \alpha\mathbf{1}/N.$$

Do not update p in place while reading: that would change the iteration. By the definition of the adjacency matrix, the accumulated $s + d\mathbf{1}/N$ is exactly $W^\top p$, so each pass realizes the PageRank fixed-point map. Stop when $\|p_{\text{new}} - p\|_1/\alpha \le \varepsilon$; this bounds the error of p, and also bounds the new error by contraction. The a priori bound $2(1-\alpha)^t$ supplies an alternative pass limit.

If even rank vectors do not fit, store rank records ordered by node identifier and merge them with the sequential adjacency stream. Emit $(j, p_i/\deg^+(i))$ contribution records, externally sort by destination, sum each destination group, add teleport/dangling mass and write the next rank file. Sorting and merging, or an equivalent external aggregation, are necessary to combine contributions to nonconsecutive destinations. This variant respects the inability to load the graph or the rank vector into memory.

Structural or analytic derivation

The graph traversal is a sparse matrix-vector product and the teleport term has rank one. The contraction proof separates mathematical accuracy from the storage representation and applies to either streaming variant in exact arithmetic.

Comparative cost and preferred method

With rank vectors in RAM, each pass reads $O(N+E)$ graph records, performs $O(N+E)$ arithmetic, and uses $O(N)$ RAM plus the current adjacency list. With external vectors, each pass additionally incurs external-sort cost for E destination records. This is a genuine input/output complexity accounting; a dense transition matrix would require N^2 storage and violates the intended setting.

7.9 AC-092 | The support of topic-specific PageRank

Reference: [7], exercise 21.13.

Source task and conventions

Characterize the source set Y relative to the teleport seed set S.

Conventional solution and proof

Let S be the actual positive-probability support of the teleport vector v, and assume $0 < \alpha < 1$ and a fixed row-stochastic link matrix W. The stationary vector is

$$\pi = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k (W^\top)^k v.$$

Every summand is nonnegative. Thus $\pi_i > 0$ exactly when there is a positive-probability link-walk from a seed in S to i, including a zero-length walk. Consequently Y is the forward-reachable closure of S and $S \subseteq Y$. If the nominal sports list contains zero teleport weights, first replace it by $\operatorname{supp} v$. At $\alpha = 1$, link moves are absent and $Y = S$ directly. Dangling completion is part of W and can enlarge the reachable set.

Structural or analytic derivation

The nonnegative Neumann series turns stationary support into graph reachability. It also proves that an unreachable page has exactly zero topic-specific PageRank, not merely a small numerical value.

Comparative cost and preferred method

Breadth-first or depth-first reachability computes Y in $O(N + E)$ time, without solving the stationary probabilities. Restricting a subsequent solve to Y is exact. Reachability alone does not determine the relative scores within Y.

7.10 AC-093 | A topic need not reach the entire web

Reference: [7], exercise 21.14.

Source task and conventions

Answer whether Y must contain every page.

Conventional solution and proof

No. Consider two pages with self-links only, and let the teleport support be $S = \{1\}$. Page 1 cannot reach page 2 by a link walk, so the stationary vector is $(1, 0)^\top$ and $Y = \{1\}$. Starting at page 2, a positive teleport probability eventually moves the surfer to page 1; there is no route back. This is compatible with a unique stationary distribution despite reducibility of the full chain. For $0 < \alpha < 1$, Y equals all pages precisely when every page is reachable from some positive-weight seed in the completed link graph.

Structural or analytic derivation

A two-vertex graph gives an exact counterexample and shows why nonempty topic support is weaker than full-support teleportation.

Comparative cost and preferred method

The question is settled by two states and their reachability relation. Numerical iteration or spectral decomposition adds no necessary information. The conventional graph counterexample is the preferred method.

7.11 AC-094 | A counterexample to pagewise topical monotonicity

Reference: [7], exercise 21.15.

Source task and conventions

Answer the source's three-star question for a sports page in S.

Conventional solution and proof

Use deterministic links $0 \to 0$, $1 \to 0$, $2 \to 1$, seed set $S = \{0, 1\}$ and $\alpha = 1/5$. Under uniform teleportation on all three pages, the balance equations give $\pi_2 = 1/15$, $\pi_1 = 1/15 + (4/5)(1/15) = 3/25$, and $\pi_0 = 61/75$. Under uniform teleportation on S, there is no stationary mass at page 2, so $\pi_{s,2} = 0$, $\pi_{s,1} = 1/10$, and $\pi_{s,0} = 9/10$. Thus

$$\boxed{\pi_s(1) = 1/10 < 3/25 = \pi(1), \qquad 1 \in S.}$$

All values are nonnegative, sum to one and satisfy the respective equations. Removing teleportation to page 2 removes inflow that had supported sports page 1. The assertion is therefore false even when the sports teleport distribution is uniform.

Structural or analytic derivation

For a fixed link operator, $\pi(v_s) - \pi(v_u) = \alpha[I - (1-\alpha)W^\top]^{-1}(v_s - v_u)$. Although the inverse is nonnegative, the difference vector has both signs. Positivity of the operator therefore does not imply positivity of every sports coordinate.

Comparative cost and preferred method

The exact three-node counterexample resolves a universal claim with a few rational equations. Its source difficulty rating is preserved; no measured human solution-time reduction is claimed. The structural sign argument explains why a superficially plausible monotonicity proof cannot succeed.

7.12 AC-095 | The personalized teleportation chain

Reference: [7], exercise 21.16.

Source task and conventions

Construct and identify the stationary chain for a mixture of two topic interests.

Conventional solution and proof

Let W be the common row-stochastic link matrix, v_s, v_p the respective teleport probability vectors, $0 < \alpha \leq 1$, and $0 \leq q \leq 1$. The described experiment first chooses whether to teleport with probability α; conditional on teleporting it chooses sports with probability q and politics with probability $1 - q$, then a page according to the corresponding topic vector. Therefore its transition matrix is

$$P_q = (1-\alpha)W + \alpha \mathbf{1}[qv_s + (1-q)v_p]^\top.$$

Let $v_q = qv_s + (1-q)v_p$. The affine map $T_q x = (1-\alpha)W^\top x + \alpha v_q$ preserves probability vectors and is a contraction with factor $1 - \alpha$, so it has exactly one stationary probability vector. The proposed personalized vector $q\pi_s + (1-q)\pi_p$ satisfies

$$T_q(q\pi_s + (1-q)\pi_p) = qT_s\pi_s + (1-q)T_p\pi_p = q\pi_s + (1-q)\pi_p.$$

It is thus the steady-state distribution of precisely the random walk in the question. The endpoints $q = 0, 1$ reduce to the corresponding single topic.

Structural or analytic derivation

The same result follows from linearity of $\alpha[I - (1-\alpha)W^\top]^{-1}v$. A common link and dangling-node policy is essential. If the link operator itself changes with personalization, this displayed affine identity needs to be rederived for that model.

Comparative cost and preferred method

After two topic solves, each full personalized vector costs $O(N)$ mixing operations. For M users, independent iterative computation costs $O(Mt(E+N))$, while basis reuse costs $O(2t(E+N)+MN)$ and $2N$ stored scores. If only a few coordinates are requested, output work can be reduced accordingly. Conventional topic-sensitive PageRank already exploits this identity; the proof supplies its exact applicability conditions.

7.13 AC-096 | Ergodicity on the reachable topic support

Reference: [7], exercise 21.17.

Source task and conventions

Prove the correct ergodicity statement and resolve the full-web qualification.

Conventional solution and proof

Let $S_q = \operatorname{supp}(qv_s + (1-q)v_p)$ and Y_q its forward-reachable closure in the completed link graph for $0 < \alpha < 1$. Every state in Y_q can teleport to any seed in S_q, and from a suitable seed a link walk reaches any other state of Y_q. Thus the restricted chain on Y_q is irreducible. A seed has a positive teleport self-loop, so this irreducible class is aperiodic. It is closed under links and teleportation. Every state outside it has probability at least α per step of entering S_q; the probability of avoiding it for t steps is at most $(1-\alpha)^t$. Hence there is one recurrent ergodic class and one globally attracting stationary distribution, zero outside Y_q.

The literal assertion that the entire web chain is irreducible is false without reachability of every page. For example, take three self-linked pages, sports seed 1 and politics seed 2, with $0 < q < 1$. Page 3 cannot be reached from either seed, so the full chain is reducible. At $\alpha = 1$, the stationary vector is the teleport vector and its positive-support class is again ergodic. Thus the personalized stationary distribution exists and is unique under the source model, but full-state irreducibility requires Y_q to equal the entire web.

Structural or analytic derivation

Contraction proves convergence and uniqueness on the full probability simplex even when states outside the recurrent class are transient. Separating this fact from irreducibility avoids an incorrect use of the word ergodic.

Comparative cost and preferred method

Reachability plus a teleport self-loop proves the recurrent-class claim in $O(N+E)$ graph work. The fixed-point iteration has the quantitative geometric bound already derived. A generic eigenvalue test is unnecessary to establish these structural facts.

Source qualification

The source statement is completed with an explicit reachable-support qualification and a counterexample to the unrestricted reading.

7.14 AC-097 | Coordinatewise mixture identity

Reference: [7], exercise 21.18.

Source task and conventions

Prove the equality of every stationary coordinate in Exercise 21.17.

Conventional solution and proof

For each page i, the two stationary equations are

$$\pi_s(i) = (1-\alpha)\sum_j W_{ji}\pi_s(j) + \alpha v_s(i), \quad \pi_p(i) = (1-\alpha)\sum_j W_{ji}\pi_p(j) + \alpha v_p(i).$$

Multiply the first by q, the second by $1-q$, and add. The vector $r_i = q\pi_s(i) + (1-q)\pi_p(i)$ then satisfies the personalized stationary equation. It is a probability vector. Uniqueness follows because if r, r' are two fixed points, $\|r - r'\|_1 \le (1-\alpha)\|r - r'\|_1$, forcing equality to zero for $\alpha > 0$. Therefore the stationary probability is exactly the claimed value at every page, including pages outside one or both topic supports.

Structural or analytic derivation

This coordinate proof is independent of diagonalizing the transition matrix. The same linearity extends to any finite convex mixture of fixed-operator topic profiles.

Comparative cost and preferred method

For k precomputed topics, one requested coordinate costs $O(k)$ arithmetic and the full N-coordinate vector costs $O(kN)$. Computing a separate stationary system for every mixture is redundant. Basis construction and storage kN remain part of the total cost.

7.15 AC-098 | The first HITS update depends on update order

Reference: [7], exercise 21.19.

Source task and conventions

Compute one iteration using the sequential convention stated beneath Equation (21.8).

Conventional solution and proof

Let A be the directed adjacency matrix and initialize $h^{(0)} = a^{(0)} = \mathbf{1}$. The source states that hub scores are recomputed first and authority scores then use the recomputed hubs. Thus

$$h^{(1)} = A\mathbf{1} = d_{\text{out}}, \qquad a^{(1)} = A^\top h^{(1)} = A^\top A\mathbf{1}.$$

For a node v, the hub score is its out-degree and the authority score is $\sum_{u\to v} \deg^+(u)$. With weighted links, replace degrees by weighted row sums and include the incoming weight in the latter sum. If an implementation uses simultaneous updates from the two old vectors instead, the first authority score is the in-degree $A^\top\mathbf{1}$; that is a different convention. Any stated normalization then divides the entire corresponding vector by its chosen norm, without changing within-vector rankings.

Structural or analytic derivation

The two sparse products implement $A^\top A$ without forming a potentially dense co-citation matrix. This preserves the source's sequential semantics and explains the second-order authority count.

Comparative cost and preferred method

The first sequential update takes $O(E + N)$ work and vector storage; explicitly multiplying matrices to form $A^\top A$ can create $O(N^2)$ fill-in. Sparse composition is the appropriate standard implementation and the structural method confirms it.

7.16 AC-099 | Rank from a linear dependence

Reference: [7], exercise 18.1.

Source task and conventions

Determine the rank of the displayed 3×3 matrix.

Conventional solution and proof

The displayed matrix is $C = \left(\begin{smallmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{smallmatrix}\right)$. Its third row is the sum of its first two rows, so rank is at most two. Its upper-left 2×2 minor has determinant one, so rank is at least two. Therefore $\operatorname{rank} C = 2$. The source calls this matrix diagonal, but its displayed entries are not diagonal; the displayed matrix is the object solved here.

Structural or analytic derivation

A dependence witness proves the upper bound and one nonzero minor proves the lower bound. These are complementary certificates and do not require approximate singular values.

Comparative cost and preferred method

A few exact row operations establish the answer. For a large numerical matrix a rank threshold requires a noise/tolerance model, but that issue is absent in this integer example. The direct algebraic method is preferable.

7.17 AC-100 | An eigenvalue and its full eigenspace

Reference: [7], exercise 18.2.

Source task and conventions

Verify $\lambda = 2$ for $C = \left(\begin{smallmatrix} 6 & -2 \\ 4 & 0 \end{smallmatrix}\right)$ and find its eigenvectors.

Conventional solution and proof

The characteristic determinant is $(6-\lambda)(-\lambda)+8 = \lambda^2 - 6\lambda + 8 = (\lambda-2)(\lambda-4)$, so 2 is an eigenvalue. Solving $(C-2I)v = 0$ gives $4v_1 - 2v_2 = 0$ in both rows. Thus every corresponding eigenvector is $v = t(1,2)^\top$ with $t \neq 0$. Direct multiplication gives $C(1,2)^\top = (2,4)^\top = 2(1,2)^\top$. A unit eigenvector is $(1,2)^\top/\sqrt{5}$.

Structural or analytic derivation

The residual $Cv - 2v = 0$ is an exact certificate for a proposed nonzero eigenvector; the nullspace calculation identifies the entire eigenspace.

Comparative cost and preferred method

For this 2×2 problem direct algebra is minimal. A general eigensolver provides no relevant asymptotic advantage and introduces unnecessary rounding.

7.18 AC-101 | A normalized symmetric eigendecomposition

Reference: [7], exercise 18.3.

Source task and conventions

Compute the decomposition of Equation (18.4) and clarify its uniqueness convention.

Conventional solution and proof

For $S = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$, the normalized eigenvectors are $u_1 = (1,1)^\top/\sqrt{2}$ with eigenvalue 3, and $u_2 = (1,-1)^\top/\sqrt{2}$ with eigenvalue 1. Hence

$$Q = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \Lambda = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad S = Q\Lambda Q^\top.$$

One checks $Q^\top Q = I$ and the product explicitly. A decomposition is not literally unique: columns may be sign-changed or permuted together with the diagonal. Ordering the eigenvalues decreasingly and choosing positive first coordinates gives the displayed convention. The preceding source prose lists the two eigenvectors in the reverse correspondence to its listed eigenvalues; direct multiplication fixes the pairing.

Structural or analytic derivation

The two orthogonal directions are the symmetric and antisymmetric combinations of coordinates. Their invariant one-dimensional subspaces diagonalize the operator without determinant expansion.

Comparative cost and preferred method

Two additions/subtractions provide the invariant directions. The symmetry method is simpler for this special matrix; it is not a general complexity result for arbitrary symmetric matrices.

7.19 AC-102 | Term co-occurrence from an incidence matrix

Reference: [7], exercise 18.4.

Source task and conventions

Compute $CC^\top$ and interpret its diagonal.

Conventional solution and proof

For $C = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, row inner products give

$$CC^\top = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

For a binary incidence matrix, $(CC^\top)_{ii} = \sum_j C_{ij}^2 = \sum_j C_{ij}$ is the number of documents containing term i, its document frequency. Off-diagonal entry (i,l) counts documents containing both terms. The diagonal interpretation uses binarity; for term-frequency counts it becomes a sum of squared frequencies.

Structural or analytic derivation

The co-occurrence matrix is a Gram matrix of term-incidence vectors, so it is PSD and its diagonal counts arise directly as squared feature norms.

Comparative cost and preferred method

If only matrix-vector actions are needed, use $C(C^\top v)$ without storing the full co-occurrence matrix. With sparse incidence this costs $O(\operatorname{nnz} C)$ per action. The question asks all nine entries, for which the direct product is the appropriate small computation.

7.20 AC-103 | Exact verification of the printed thin SVD

Reference: [7], exercise 18.5.

Source task and conventions

Verify every SVD property using exact radicals in place of rounded decimals.

Conventional solution and proof

The exact matrices corresponding to the printed values are

$$U = \begin{pmatrix} -2/\sqrt{6} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} \\ -1/\sqrt{6} & 1/\sqrt{2} \end{pmatrix}, \quad \Sigma = \operatorname{diag}(\sqrt{3}, 1), \quad V^\top = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}.$$

The squared norms of the columns of U are $(4+1+1)/6 = 1$ and $(0+1+1)/2 = 1$, and their inner product is zero. The rows of $V^\top$ likewise have unit norm and zero mutual inner product. The singular values are nonnegative and decreasing. Multiplying the two rank-one terms gives

$$\sqrt{3} u_1 v_1^\top = \begin{pmatrix} 1 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}, \quad u_2 v_2^\top = \begin{pmatrix} 0 & 0 \\ -1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix},$$

whose sum is C from Equation (18.12). Thus $C = U\Sigma V^\top$ and $U^\top U = V^\top V = I_2$. This is a thin SVD; $UU^\top$ is a rank-two projector, not I_3. For a full SVD append $u_3 = (-1, 1, 1)^\top/\sqrt{3}$ and a zero row to Σ. The rounded decimals in the source satisfy these equalities only approximately.

Structural or analytic derivation

The exact verification supplies all orthogonality and reconstruction certificates and resolves the distinction between full and reduced SVD conventions.

Comparative cost and preferred method

Checking a supplied rank-r factorization costs the relevant matrix products, without solving an eigenproblem again. For this example the radical arithmetic is exact and preferable to accepting rounded entries as exact orthogonal matrices.

7.21 AC-104 | Document co-occurrence from a binary incidence matrix

Reference: [7], exercise 18.6.

Source task and conventions

Interpret every entry of $C^\top C$.

Conventional solution and proof

Entry (j, l) is $\sum_i C_{ij} C_{il}$, the number of vocabulary terms present in both documents j and l. On the diagonal it counts distinct represented terms in that document. For the two-document example in Equation (18.12), $C^\top C = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$. If repeated occurrences are encoded as counts, the same product becomes an unnormalized term-frequency inner product and loses the simple set-cardinality interpretation.

Structural or analytic derivation

The document Gram matrix and term Gram matrix have the same nonzero eigenvalues, because they arise from the same SVD. This is the basis of choosing the smaller dimension in latent semantic computations.

Comparative cost and preferred method

Compute a requested pair by intersecting its sparse term sets; materialize the full Gram matrix only when its complete entries are needed. For a linear operator action, $C^\top(Cv)$ avoids dense storage. No general time improvement is claimed for writing an output containing all document pairs.

7.22 AC-105 | Term-frequency inner products

Reference: [7], exercise 18.7.

Source task and conventions

Compute the specified $CC^\top$ and interpret its large entries.

Conventional solution and proof

With $C = \left(\begin{smallmatrix} 0 & 2 & 1 \\ 0 & 3 & 0 \\ 2 & 1 & 0 \end{smallmatrix}\right)$,

$$CC^\top = \begin{pmatrix} 5 & 6 & 2 \\ 6 & 9 & 3 \\ 2 & 3 & 5 \end{pmatrix}.$$

For instance, terms 1 and 2 contribute $2 \cdot 3 = 6$ from document 2, whereas terms 1 and 3 contribute $2 \cdot 1 = 2$. Diagonal entries are sums of squared counts. The product weights co-occurrence by magnitude; a large diagonal can therefore reflect large raw counts and is not a normalized similarity score.

Structural or analytic derivation

Writing the product as a sum of document outer products $CC^\top = \sum_j c_j c_j^\top$ localizes each document's contribution and proves PSD simultaneously.

Comparative cost and preferred method

The three-column calculation is direct. In a sparse collection, outer-product accumulation can exploit short documents, but the total work depends on the sum of squared document nonzero counts. The Gram interpretation supplies correctness, not an unconditional sparse speedup.

7.23 AC-106 | The optimal rank-one approximation and its exact error

Reference: [7], exercise 18.8.

Source task and conventions

Compute the rank-one approximation referred to by Equations (18.12)–(18.13).

Conventional solution and proof

The source's "Example 18.12" and "Exercise 18.13" are references to the numbered equations. Using their exact SVD gives

$$C_1 = \sqrt{3}u_1 v_1^\top = \begin{pmatrix} 1 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}.$$

The residual is $u_2 v_2^\top$, so its Frobenius norm is $\|u_2\|\|v_2\| = 1$. Directly, its four nonzero entries have magnitude $1/2$, and their squared sum is one. To prove optimality here without merely citing it, let B have rank at most one. Its restriction to the two-dimensional span of v_1, v_2 has a nonzero kernel vector; choose a unit vector w in that kernel. Then $\|(C-B)w\| = \|Cw\| \geq \sigma_2 = 1$, because $\|Cw\|^2 = 3|\langle w, v_1\rangle|^2 + |\langle w, v_2\rangle|^2 \geq 1$. Since $\|C-B\|_F \geq \|C-B\|_2 \geq 1$, the displayed approximation attains the least possible error.

Structural or analytic derivation

Orthogonal singular components isolate the discarded energy exactly. The lower-bound witness proves optimality over every rank-one competitor, not just over approximations obtained by deleting the second term.

Comparative cost and preferred method

A rank-one factorization stores $M + N + 1$ scalars and applies to a vector in $O(M+N)$ arithmetic. For this 3×2 example the factor representation is not smaller than the six original entries; it illustrates the theorem, not a practical compression win at this size.

7.24 AC-107 | A rank-one SVD term is an outer product

Reference: [7], exercise 18.9.

Source task and conventions

Prove the equality in Exercise 18.9 with dimensions made explicit.

Conventional solution and proof

Here $U_1 = u_1 \in \mathbb{R}^{M\times 1}$, $V_1 = v_1 \in \mathbb{R}^{N\times 1}$ and σ_1 is a scalar. For every entry (i,j),

$$(U_1 \sigma_1 V_1^\top)_{ij} = u_{i1}\sigma_1 v_{j1} = \sigma_1 (u_1 v_1^\top)_{ij}.$$

Thus $U_1\sigma_1 V_1^\top = \sigma_1 u_1 v_1^\top$. Its image is contained in $\operatorname{span} u_1$, so it has rank at most one and exactly one when $\sigma_1 > 0$ and the vectors are nonzero. For any vector x, its action is $\sigma_1 u_1 (v_1^\top x)$.

Structural or analytic derivation

The outer-product form is an exact separable representation of the two-index matrix. It is the finite-dimensional prototype of a rank-one operator on a Hilbert space.

Comparative cost and preferred method

Applying the factors costs N dot-product terms and M scalar multiplications, versus MN products for a materialized dense matrix. Producing all MN entries still costs $\Omega(MN)$ output work. The benefit applies to operator actions and storage, not to avoiding a requested full output.

7.25 AC-108 | Full and reduced singular-value arrays

Reference: [7], exercise 18.10.

Source task and conventions

Write both singular-value matrices for the 4×2 example.

Conventional solution and proof

Example 18.3 has $C = \left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \\ -1 & 1 \end{smallmatrix}\right)$. Its column Gram matrix is $C^\top C = \left(\begin{smallmatrix} 3 & -2 \\ -2 & 3 \end{smallmatrix}\right)$, with eigenvalues $5, 1$, so its singular values are $\sqrt{5}, 1$. In a full SVD with $U \in \mathbb{R}^{4\times 4}$ and $V \in \mathbb{R}^{2\times 2}$,

$$\Sigma_2 = \begin{pmatrix} \sqrt{5} & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \Sigma_2' = \begin{pmatrix} \sqrt{5} & 0 \\ 0 & 1 \end{pmatrix}.$$

The reduced representation uses the first two columns of U, so it yields exactly the same product. The example already displays a reduced SVD; its printed 2×2 array is therefore Σ_2'. Distinguishing that convention explains why no additional zero rows appear in the example itself.

Structural or analytic derivation

For a general rank-k truncation, $C_k = \sum_{j=1}^{k} \sigma_j u_j v_j^\top = U_k' \Sigma_k' V_k'^\top$, because every omitted diagonal entry in the full rectangular array is zero. Entrywise expansion proves the equality.

Comparative cost and preferred method

Storing factors requires $k(M + N) + k$ scalars and applying them costs $O(k(M + N))$, compared with a full matrix's MN entries and action cost. This is useful when $k \ll \min(M, N)$; at full rank of a small matrix it need not reduce storage.

Chapter 8

Foundations of Machine Learning

Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. Reference problems and derived analyses.

8.1 AC-109 | PCA and the Rayleigh quotient

Reference: [8], exercise 15.1.

Source task and conventions

Prove both variance and maximal-variance assertions for the uncentred data matrix.

Conventional solution and proof

Let the columns be $x_1, \ldots, x_m \in \mathbb{R}^N$, $\bar{x} = m^{-1} \sum_i x_i$, and $Z = X - \bar{x}\mathbf{1}^\top$. The projected observations are $s_i = u^\top x_i$ and their mean is $\bar{s} = u^\top \bar{x}$. Thus

$$\frac{1}{m} \sum_i (s_i - \bar{s})^2 = \frac{1}{m} \sum_i u^\top (x_i - \bar{x})(x_i - \bar{x})^\top u = u^\top C u, \qquad C = ZZ^\top / m.$$

This proves (a), with the book's divisor m. For (b), C is symmetric positive semidefinite because $v^\top C v = \|Z^\top v\|^2 / m \geq 0$. By the real spectral theorem, $C = Q \operatorname{diag}(\lambda_1, \ldots, \lambda_N) Q^\top$, $\lambda_1 \geq \cdots \geq 0$. For $\|u\| = 1$, writing $a = Q^\top u$ gives $u^\top C u = \sum_j \lambda_j a_j^2 \leq \lambda_1 \sum_j a_j^2 = \lambda_1$. Equality holds precisely when u lies in the top eigenspace. The leading left singular vector of Z is such a vector. If the top eigenvalue is repeated, the direction is not unique; if $Z = 0$, every unit direction is optimal.

Structural or analytic derivation

The orthogonal decomposition $Z = uu^\top Z + (I - uu^\top)Z$ gives

$$\|Z - uu^\top Z\|_F^2 = \|Z\|_F^2 - m u^\top C u.$$

Hence maximal variance and minimal rank-one projection residual are the same optimization problem. The equality follows column by column from Pythagoras, so it also proves the variational interpretation without differentiating a constrained objective.

Comparative cost and preferred method

Both correct spectral methods require centring. A covariance construction costs $O(mN^2)$ arithmetic and $O(N^2)$ storage; a thin SVD can work directly with Z, at $O(mN \min(m, N))$ classical dense cost. When $m \ll N$, the $m \times m$ Gram formulation reduces the eigensystem dimension. The proof certifies the global optimum; it does not establish a runtime advantage over an already appropriate thin-SVD implementation.

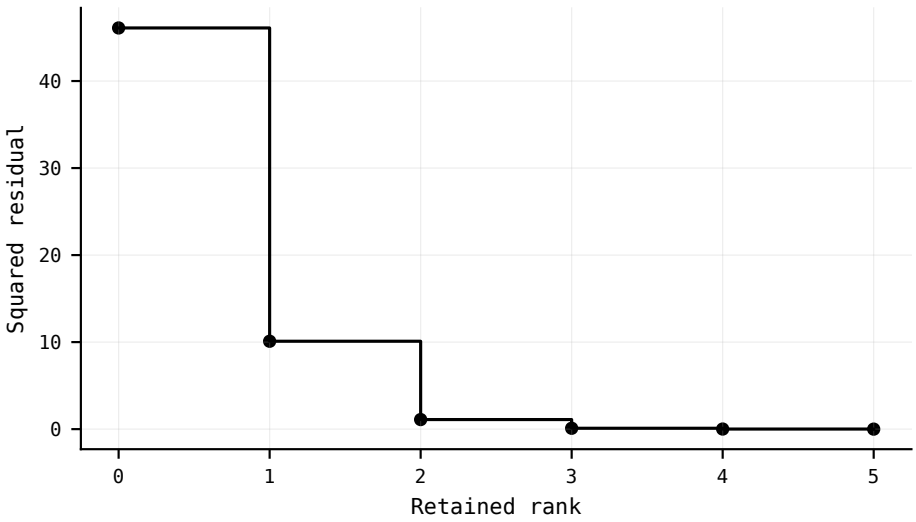


Figure 8.1: Declared illustrative singular spectrum (6,3,1,0.3,0.1). Each retained singular direction removes exactly its squared singular value from the Frobenius residual.

8.2 AC-110 | Nyström approximation as orthogonal projection

Reference: [8], exercise 15.4.

Source task and conventions

Complete parts (a)–(e), including singular landmark blocks and the specified storage calculation.

Conventional solution and proof

Let $K = X^\top X \succeq 0$, write $X = [X_0\ X_1]$, and let the compact SVD be $X_0 = U_r \Sigma_r V_r^\top$, where the diagonal entries of Σ_r are positive. Then

$$W = V_r \Sigma_r^2 V_r^\top, \quad W^\dagger = V_r \Sigma_r^{-2} V_r^\top, \quad X_0 W^\dagger X_0^\top = U_r U_r^\top =: P.$$

(a) Every principal submatrix of K is positive semidefinite: $a^\top W a = (a,0)^\top K (a,0) \geq 0$. Also $WW^\dagger W = W$ and $K_{21} W^\dagger W = X_1^\top X_0 V_r V_r^\top = K_{21}$. Expanding $C W^\dagger C^\top$ therefore reproduces the first row and column blocks of K, and

$$K - \widetilde{K} = \begin{pmatrix} 0 & 0 \\ 0 & K_{22} - K_{21} W^\dagger K_{21}^\top \end{pmatrix}.$$

The squared Frobenius norm is the sum of squared entries, proving the claimed norm equality.

(b) Since $C = X^\top X_0$, the SVD calculation gives $\widetilde{K} = X^\top P X$. The matrix P is the orthogonal projector onto $\operatorname{ran} X_0$: it fixes that span and annihilates its orthogonal complement. (c) For every a, $a^\top \widetilde{K} a = \|PXa\|^2 \geq 0$, so $\widetilde{K}$ is SPSD. The residual is also SPSD, since $K - \widetilde{K} = X^\top (I - P) X$. (d) If $\operatorname{rank} K = \operatorname{rank} W$, then $\dim \operatorname{ran} X = \dim \operatorname{ran} X_0$ and inclusion of the latter in the former implies equality. Consequently $PX = X$ and $\widetilde{K} = K$.

(e) With decimal $M = 10^6$, $K = 10^3$, and eight bytes per double, $m = 20{,}000{,}000$ gives $8m^2 = 3.2 \times 10^{15}$ bytes, or 3.2 PB. Storing C and W independently requires $8(ml + l^2) = 1.6008 \times 10^{12}$ bytes, or 1.6008 TB, for $l = 10{,}000$. If the duplicated W rows inside C are not separately stored, storage is $8ml = 1.6$ TB. These are raw array counts, excluding workspaces and metadata.

Structural or analytic derivation

The factorization $\widetilde{K} = (PX)^\top (PX)$ proves symmetry, positivity, the residual sign and exactness in one representation. A thin factor $B = CV_r\Sigma_r^{-1}$ gives $\widetilde{K} = BB^\top$ and can replace $W^\dagger$ in subsequent operations. The identity remains valid for singular W; an ordinary inverse would fail in that case.

Comparative cost and preferred method

The full matrix has m^2 entries; the stated approximation representation has $ml + l^2$. Their ratio is $m^2/(ml + l^2) \approx 1999.0005$. Matrix-vector products can use $B(B^\top v)$ in $O(mr)$ arithmetic after factor construction, compared with $O(m^2)$ for a stored dense matrix. This is an exact representation of the approximation, not a claim that arbitrary landmarks have small error. When the rank condition in (d) holds, the representation is exact. Landmark choice, kernel evaluation, factorization and external-memory costs must be included in an application.

Graphical reference. See Figure 8.1 for the shared construction.

8.3 AC-111 | Positive rescaling of a kernel

Reference: [8], exercise 6.1.

Source task and conventions

Prove positive semidefiniteness of $K(x, y)/(\alpha(x)\alpha(y))$ for strictly positive α.

Conventional solution and proof

For arbitrary points $x_1, \ldots, x_m$ and real coefficients c_i, put $d_i = c_i/\alpha(x_i)$. Symmetry is inherited from K, and

$$\sum c_i c_j K'(x_i, x_j) = \sum d_i d_j K(x_i, x_j) \geq 0.$$

The denominators exist because $\alpha(x) > 0$. This verifies the defining quadratic-form condition for every finite sample, including repeated points.

Structural or analytic derivation

If $K(x, y) = \langle \phi(x), \phi(y) \rangle$, use $\phi'(x) = \phi(x)/\alpha(x)$. Equivalently the Gram matrix is the congruence $D^{-1}GD^{-1}$, where $D = \mathrm{diag}(\alpha(x_i))$. A congruence preserves positive semidefiniteness.

Comparative cost and preferred method

The symbolic certificate is independent of sample size. Scaling an already stored $m \times m$ Gram matrix costs $O(m^2)$ arithmetic; testing it by a generic eigensolver costs $O(m^3)$ and only checks that one matrix. Both proofs are elementary and equally short; no general computational speedup in evaluating the kernel is implied.

8.4 AC-112 | Eleven positive kernels: explicit feature and integral

Reference: [8], exercise 6.2.

Source task and conventions

Prove parts (a)–(k). In part (e), identify the domain issue at zero.

Conventional solution and proof

We use three elementary closure facts. A feature inner product is PDS because its quadratic form is $\|\sum_i c_i\phi(x_i)\|^2$. Sums and nonnegative integrals of PDS kernels are PDS by linearity of finite quadratic forms. Products are PDS: for finite Gram matrices write $G_{ij} = \sum_a v_{ia}v_{ja}$ and $H_{ij} = \sum_b w_{ib}w_{jb}$; then $(G \circ H)_{ij} = \sum_{a,b}(v_{ia}w_{ib})(v_{ja}w_{jb})$. Nonnegative power series follow by finite partial sums and pointwise limits. In particular,

$$e^{-t\|x-y\|^2} = e^{-t\|x\|^2}e^{-t\|y\|^2}\sum_{n=0}^{\infty}\frac{(2t)^n}{n!}\langle x,y\rangle^n$$

is PDS for $t \geq 0$, since tensor powers represent the inner-product powers.

(a) $\cos(x-y) = \cos x\cos y + \sin x\sin y$, with feature $(\cos x, \sin x)$. (b) Compose this feature with $x \mapsto x^2$. (c) Each summand is the inner product of the nth tensor powers of $(\cos x_i^2, \sin x_i^2)$; their direct sum is a feature for the displayed sum.

(d) For $x, y > 0$, $(x+y)^{-1} = \int_0^\infty e^{-tx}e^{-ty}\,dt$, the inner product of the functions e^{-tx} in $L^2(0,\infty)$. They have squared norm $1/(2x) < \infty$.

(e) For nonzero vectors, $K(x,y) = \langle x/\|x\|, y/\|y\|\rangle$. The angle with a zero vector is undefined, so the source expression does not define a kernel on all of $\mathbb{R}^n$. The stated proof applies on $\mathbb{R}^n \setminus \{0\}$. Defining the feature at zero to be zero provides a PDS extension, with $K(0,y) = 0$.

(f) Set $v(x) = (\cos 2x, \sin 2x)$. Since $\|v(x)-v(y)\|^2 = 4\sin^2(x-y)$, the kernel is the Gaussian $e^{-(\lambda/4)\|v(x)-v(y)\|^2}$.

(g) For $a > 0$ and $s \geq 0$,

$$e^{-a\sqrt{s}} = \frac{a}{2\sqrt{\pi}}\int_0^\infty t^{-3/2}e^{-a^2/(4t)}e^{-ts}\,dt.$$

Here is a proof of the integral identity. Substituting $u = a/(2\sqrt{t})$ reduces the right side to $2I(b)/\sqrt{\pi}$, where $b = a\sqrt{s}/2$ and $I(b) = \int_0^\infty e^{-u^2-b^2/u^2}\,du$. For $b > 0$, differentiation under the integral and the substitution $v = b/u$ give $I'(b) = -2I(b)$. Moreover $I(0) = \sqrt{\pi}/2$ and dominated convergence gives continuity at zero. Thus $I(b) = (\sqrt{\pi}/2)e^{-2b}$. With $a = 1/\sigma$, $s = \|x-y\|^2$, the positive Gaussian mixture proves (g).

(h) Use $\phi_x(t) = \mathbf{1}_{[0,x]}(t) - x$ in $L^2[0,1]$. Expanding its inner product gives $\min(x,y) - xy$; endpoint choices on null sets are immaterial.

(i) The binomial expansion is

$$(1-s)^{-1/2} = \sum_{n\geq 0} b_n s^n, \qquad b_n = \binom{2n}{n}/4^n > 0.$$

For $\|x\|, \|y\| < 1$, use the Hilbert direct-sum feature $\phi(x) = (\sqrt{b_n}\, x^{\otimes n})_{n\geq 0}$. Its norm is finite because $\sum b_n \|x\|^{2n} = (1-\|x\|^2)^{-1/2}$. Its inner product is the required kernel.

(j) Since $(1+s)^{-1} = \int_0^\infty e^{-u} e^{-us}\, du$, insert $s = \|x-y\|^2/\sigma^2$ and use Gaussian positivity.

(k) The base kernel $J(x,y) = \sum_i \min(|x_i|, |y_i|)$ is PDS through the direct-sum features $\mathbf{1}_{[0,|x_i|]}(t)$. Therefore $e^{J/\sigma^2} = \sum_{n\geq 0} J^n/(\sigma^{2n} n!)$ is PDS. All series and integrals used above converge at every pair in the declared domain, so finite quadratic forms pass to their limits.

Structural or analytic derivation

The certificates specify finite features in (a), (b), and the tensor construction in (c), square-integrable functions in (d), (h), and (k), and positive mixtures or analytic series in (f), (g), (i), and (j). These representations also expose rank and singularities: (a)–(b) have sample Gram rank at most two; (i) requires strict unit-ball membership; (e) requires a nonzero norm or an explicit extension.

Comparative cost and preferred method

A numerical eigenvalue test on one m-point sample cannot prove the all-samples definition. The feature/integral proofs do. For (a)–(b), a Gram action Gc can use a two-column feature matrix in $O(m)$ operations and $O(m)$ storage, compared with $O(m^2)$ for a materialized dense Gram matrix. For the remaining kernels, positivity alone supplies no finite-rank reduction or approximation accuracy. The preferred method for the proof is structural; the preferred evaluation algorithm depends on the kernel and data.

Source qualification

Part (e) is completed with an explicit corrected domain and extension. No value of an undefined angle is assumed.

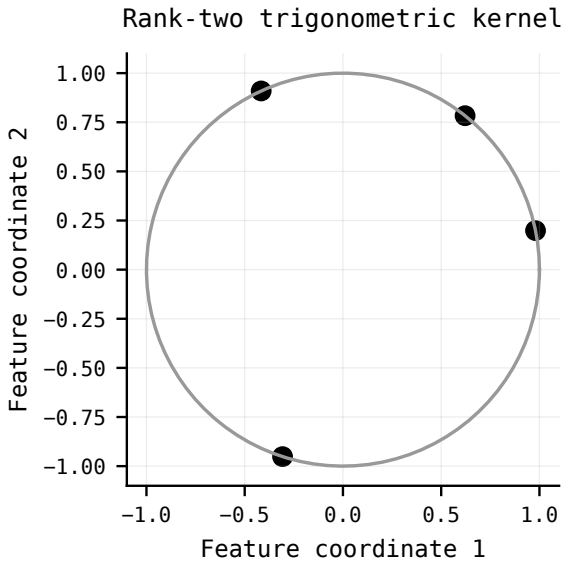


Figure 8.2: The feature map (cos x, sin x) makes all Gram matrices PSD and of rank at most two.

8.5 AC-113 | Two-step graph walks and Gram factorization

Reference: [8], exercise 6.3.

Source task and conventions

Interpret the source hint W and prove the resulting graph kernel is PDS.

Conventional solution and proof

Take the symmetric weighted adjacency matrix W, with zero for absent edges. The sum over length-two walks from p to q is

$$K(p,q) = \sum_{r \in V} W_{pr} W_{rq} = (W^2)_{pq}.$$

For any real vector c, $c^\top K c = c^\top W^\top W c = \|Wc\|^2 \geq 0$. No nonnegativity assumption on the edge weights is necessary. Symmetry is immediate.

Structural or analytic derivation

The feature of vertex p is its weighted adjacency row. The inner product of two such features sums their weighted common-neighbour contributions. If "path" is required to be simple and hence to exclude the walk p, r, p, the claimed theorem fails: a three-vertex path has zero diagonal and a positive $(1,3)$ entry, giving a negative quadratic form on $(1, 0, -1)$. The source's matrix hint determines the walk interpretation used in this solution.

Comparative cost and preferred method

With e nonzero adjacency entries, applying K as $W(Wc)$ costs $O(e)$ and uses the sparse adjacency representation, whereas storing a dense K uses $O(|V|^2)$ entries. The saving is output-sensitive: constructing all entries of K still requires writing them. The proof is a universal factorization certificate, not a sample eigenvalue check.

Source qualification

The word path is retained from the source, with the necessary walk convention explicitly resolved.

8.6 AC-114 | Symmetric-difference kernel

Reference: [8], exercise 6.4.

Source task and conventions

Prove positivity of $\exp(-|A\triangle B|/2)$.

Conventional solution and proof

Enumerate the finite universe and associate each set A with its indicator vector v_A. Each coordinate contributes one to $\|v_A - v_B\|^2$ precisely when membership differs, so $|A\triangle B| = \|v_A - v_B\|^2$. The Gaussian expansion proved in the preceding kernel dossier shows the kernel is PDS. Explicitly,

$$K(A,B) = e^{-|A|/2} e^{-|B|/2} e^{|A\cap B|},$$

and $|A\cap B| = v_A^\top v_B$ is a feature inner product; its exponential is a convergent nonnegative sum of tensor-power kernels.

Structural or analytic derivation

A second proof factors over the universe: $K(A,B) = \prod_j k(v_{A,j}, v_{B,j})$, where the two-state Gram matrix of k is $\left(\begin{smallmatrix} 1 & e^{-1/2} \\ e^{-1/2} & 1 \end{smallmatrix}\right)$. Its eigenvalues $1 \pm e^{-1/2}$ are positive. Tensor products preserve positivity, including all $2^{|X|}$ sets simultaneously.

Comparative cost and preferred method

The product proof reduces positivity on an exponentially large set domain to a 2×2 calculation and a proved closure rule. Kernel evaluation can use a bitwise symmetric difference and population count; this conventional implementation already realizes the set identity efficiently. No additional complex-analytic speedup is asserted.

8.7 AC-115 | Set-sum kernel

Reference: [8], exercise 6.5.

Source task and conventions

Prove positivity of the sum of a base kernel over two subsets.

Conventional solution and proof

For each subset A, let v_A be its indicator vector indexed by the finite universe and let G be the base Gram matrix. Then $K'(A,B) = v_A^\top G v_B$. For sets A_i and coefficients c_i,

$$\sum_{i,j} c_i c_j K'(A_i, A_j) = \Big(\sum_i c_i v_{A_i}\Big)^\top G \Big(\sum_j c_j v_{A_j}\Big) \geq 0.$$

The kernel is symmetric because G is symmetric. Empty sets have zero feature and cause no exception.

Structural or analytic derivation

A base feature map gives the aggregate feature $\Phi(A) = \sum_{x\in A} \phi(x)$. The set kernel is $\langle \Phi(A), \Phi(B)\rangle$. This proves that the construction sums features without requiring pairwise expansion on each repeated query.

Comparative cost and preferred method

Direct evaluation uses $|A||B|$ base-kernel calls. With an available d-dimensional base feature, aggregate construction costs $O(d(|A|+|B|))$ and each subsequent comparison costs $O(d)$. The advantage requires feature access and enough reuse; an arbitrary black-box kernel need not provide a small finite feature map.

8.8 AC-116 | Two conditionally negative kernels

Reference: [8], exercise 6.6.

Source task and conventions

Prove parts (a) and (b) under the NDS zero-sum convention.

Conventional solution and proof

For a symmetric kernel, NDS means $\sum_{i,j} c_i c_j K(x_i, x_j) \le 0$ whenever $\sum_i c_i = 0$. (a) $\sin^2(x-y) = (1 - \cos(2x-2y))/2$. The constant term has zero quadratic form, while the cosine term has quadratic form $(\sum_i c_i \cos 2x_i)^2 + (\sum_i c_i \sin 2x_i)^2$. Its negative half is nonpositive. (b) For $a > 0$,

$$\log a = \int_0^\infty \frac{e^{-t} - e^{-at}}{t}\, dt.$$

The integral converges at zero and infinity; its derivative in a is $\int_0^\infty e^{-at} dt = 1/a$ and it vanishes at $a = 1$, proving the identity. Inserting $a = x_i + x_j$ and using $\sum c_i = 0$ gives

$$\sum_{i,j} c_i c_j \log(x_i + x_j) = -\int_0^\infty \frac{(\sum_i c_i e^{-t x_i})^2}{t}\, dt \le 0.$$

For positive x_i, decay at infinity is exponential; at zero the numerator is $O(t^2)$ because the coefficients sum to zero. Thus the displayed integral and the finite interchange are justified.

Structural or analytic derivation

The negative quadratic forms are explicitly squared norms. Part (a) uses a two-dimensional trigonometric feature; part (b) uses a Laplace feature with the constant component eliminated by the zero-sum constraint.

Comparative cost and preferred method

Each proof treats all finite samples. Part (a)'s quadratic form can be evaluated in $O(m)$ operations instead of $O(m^2)$ pairwise evaluations. Part (b)'s integral certificate proves sign but does not supply a finite exact evaluation algorithm for arbitrary points. NDS must not be confused with ordinary negative semidefiniteness on unrestricted coefficient vectors.

8.9 AC-117 | A two-point obstruction to positive definiteness

Reference: [8], exercise 6.7.

Source task and conventions

Determine whether $|x - y|$ is PDS.

Conventional solution and proof

Choose points $0, 1$. The Gram matrix is $G = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. For $c = (1, -1)^\top$, $c^\top G c = -2 < 0$, so the kernel is not PDS. Equivalently, its eigenvalues are $1, -1$. A general PSD matrix with a zero diagonal entry must have the corresponding row zero, because its 2×2 principal minors satisfy $|G_{ij}|^2 \le G_{ii} G_{jj}$; this kernel violates that necessary condition.

Structural or analytic derivation

The counterexample is already minimal: one nonnegative diagonal value cannot disprove PDS, while two distinct points do. This obstruction is more economical than constructing a large sampled Gram matrix.

Comparative cost and preferred method

Two points and one quadratic form certify failure for the entire proposed kernel. The conventional counterexample is the preferred method; introducing contour integrals would add no information needed by the question.

8.10 AC-118 | Fractional distance: non-PDS and NDS

Reference: [8], exercise 6.8.

Source task and conventions

Determine both signs for $K(x, y) = \|x - y\|^{3/2}$.

Conventional solution and proof

The two points $0, e_1$ give the same indefinite matrix as the preceding question, so K is not PDS. For NDS, put $\alpha = 3/4$. Integration by parts, with vanishing endpoint terms, yields for $s \ge 0$

$$s^\alpha = \frac{\alpha}{\Gamma(1-\alpha)} \int_0^\infty (1 - e^{-ts}) t^{-1-\alpha}\, dt.$$

Indeed the substitution $u = ts$ reduces the integral to $s^\alpha \Gamma(1-\alpha)/\alpha$; at zero the integrand is $O(t^{-\alpha})$ and at infinity $O(t^{-1-\alpha})$. For coefficients summing to zero,

$$\sum_{i,j} c_i c_j \|x_i - x_j\|^{3/2} = -\frac{\alpha}{\Gamma(1-\alpha)} \int_0^\infty c^\top G_t c \; t^{-1-\alpha}\, dt \le 0,$$

where $(G_t)_{ij} = e^{-t\|x_i - x_j\|^2}$ is PDS by the Gaussian feature expansion. The finite sum of convergent difference integrals justifies the interchange.

Structural or analytic derivation

The exponent $3/2$ is twice a number in $(0, 1)$, so the distance power is a positive mixture of Gaussian-kernel deficits. This gives a sign certificate on every Euclidean dimension and every finite point set.

Comparative cost and preferred method

The PDS decision takes a two-point test; the NDS conclusion requires a universal representation. A finite eigensolver can supply a counterexample but cannot establish the all-samples NDS theorem. The mixture proof improves verification scope, not the cost of evaluating one distance.

8.11 AC-119 | Negative definiteness on a Hilbert space

Reference: [8], exercise 6.9.

Source task and conventions

Prove the source claim for $K(x, y) = 1 - \langle x, y\rangle$.

Conventional solution and proof

For real coefficients with $\sum_i c_i = 0$,

$$\sum_{i,j} c_i c_j [1 - \langle x_i, x_j\rangle] = (\sum_i c_i)^2 - \left\|\sum_i c_i x_i\right\|^2 \leq 0.$$

Symmetry holds in a real Hilbert space. In a complex Hilbert space, use Hermitian symmetry and coefficients $c_i \in \mathbb{C}$ with $\sum_i c_i = 0$, replacing the quadratic form by $\sum_{i,j} \overline{c}_i c_j K(x_i, x_j)$ in a consistent inner-product convention. The same norm identity applies.

Structural or analytic derivation

The constant kernel is annihilated on the zero-sum subspace, leaving the negative Gram form. No finite-dimensional approximation to the Hilbert space is required.

Comparative cost and preferred method

The certificate uses one vector sum and one squared norm. For m vectors of finite dimension d, this is $O(md)$ instead of explicitly forming m^2 pairwise inner products at $O(m^2 d)$. Ordinary unrestricted negative semidefiniteness would be a different and generally false assertion.

8.12 AC-120 | All least-squares solutions and the nullspace

Reference: [8], exercise 11.3.

Source task and conventions

Answer both rank and nonuniqueness questions with the source convention $X \in \mathbb{R}^{N\times m}$.

Conventional solution and proof

(a) $XX^\top$ is invertible exactly when X has row rank N, since $v^\top XX^\top v = \|X^\top v\|^2$ vanishes exactly on $\ker X^\top$. This requires $m \geq N$. (b) Let $X = U_r \Sigma_r V_r^\top$ be a compact SVD. Then

$$w_* = (XX^\top)^\dagger Xy = U_r \Sigma_r^{-1} V_r^\top y, \quad X^\top w_* = V_r V_r^\top y = X^\dagger Xy.$$

Consequently $X^\top w = X^\dagger Xy$ holds exactly when $X^\top(w - w_*) = 0$. All solutions are

$$\boxed{w = w_* + v, \quad v \in \ker X^\top} \quad \text{or} \quad w = w_* + (I - XX^\dagger)z,\ z \in \mathbb{R}^N.$$

The projector formula follows from $XX^\dagger = U_r U_r^\top$. Since $w_* \perp \ker X^\top$, $\|w\|^2 = \|w_*\|^2 + \|v\|^2$, proving that w_* is the unique minimum-norm solution even when the prediction-equivalent coefficient vector is not unique.

Structural or analytic derivation

The fitted response is the unique orthogonal projection of y onto $\operatorname{ran} X^\top$. The coefficient ambiguity is a whole affine nullspace. This distinction prevents a singular inverse from being treated as an admissible formula.

Comparative cost and preferred method

A rank-revealing SVD or QR supplies the solution and its complete nullspace certificate in one factorization. Normal-equation inversion is inadmissible when rank deficient and squares the condition number when full rank. A suitable conventional rank-revealing factorization is already the preferred algorithm; the gain is a complete characterization, not a new asymptotic improvement.

8.13 AC-121 | Kernel perturbation and the resolvent identity

Reference: [8], exercise 11.4.

Source task and conventions

Prove the dual-parameter perturbation bound and explicitly adjudicate the printed sign.

Conventional solution and proof

Put $A = K + \lambda I$, $A' = K' + \lambda I$, with $K, K' \succeq 0$, $\lambda > 0$, and $\alpha = A^{-1}y$, $\alpha' = A'^{-1}y$. Multiplying out gives

$$A'^{-1} - A^{-1} = A'^{-1}(A - A')A^{-1},$$

hence

$$\boxed{\alpha' - \alpha = -A'^{-1}(K' - K)A^{-1}y.}$$

This has the opposite sign from the displayed identity in part (a); the source hint has the correct minus sign. A scalar counterexample is $K = 0, K' = 1, \lambda = y = 1$: the actual difference is $-1/2$, while the printed right side is $+1/2$. For (b), spectral positivity gives $\|A^{-1}\|_2, \|A'^{-1}\|_2 \le 1/\lambda$. If $|y_i| \le M$, then $\|y\|_2 \le \sqrt{m}M$. Submultiplicativity therefore yields

$$\|\alpha' - \alpha\|_2 \le \lambda^{-2}\|K' - K\|_2\sqrt{m}M.$$

These arguments include singular unregularized K, since adding λI makes both matrices positive definite.

Structural or analytic derivation

The identity is an exact first-resolvent identity, not a first-order approximation. It also gives the sharper a posteriori bound $\|\alpha' - \alpha\| \le \|K' - K\|\,\|\alpha\|/\lambda$, and bounds using actual smallest eigenvalues can replace λ^{-2}. If $\Delta K = UV^\top$ has small rank, the Woodbury update follows by multiplying the proposed inverse and checking the identity, provided $I + V^\top A^{-1}U$ is invertible.

Comparative cost and preferred method

The norm bound certifies all perturbations of a declared size without refitting each one. It is a verification benefit; computing an exact spectral norm can itself be costly. A known perturbation-norm bound or a low-rank factor is required for a practical reduction. The sign correction is essential even though the norm inequality is unaffected by it.

8.14 AC-122 | Unregularized regression has no uniform stability bound

Reference: [8], exercise 14.3.

Source task and conventions

Examine the limit in Corollary 14.6 and construct a counterexample at $\lambda = 0$.

Conventional solution and proof

(a) With the source constants held fixed, the additive terms are

$$\frac{4Mr^2}{\lambda m} + \left(\frac{8Mr^2}{\lambda} + M\right)\sqrt{\frac{\log(1/\delta)}{2m}},$$

which diverge as $\lambda \downarrow 0$ when the relevant constants are nonzero. Thus this corollary gives no finite unregularized guarantee. Allowing a loss bound M to depend on λ cannot improve this conclusion without additional control. (b) Consider one-dimensional least squares with m examples, $m-1$ copies of $(x, y) = (0, 0)$, and one example $(\varepsilon, 1)$, $0 < \varepsilon \leq 1$. The unique empirical minimizer is $w = 1/\varepsilon$. Replace the last example by $(\varepsilon, 0)$; the unique minimizer is $w' = 0$. Both datasets have bounded inputs and labels. At the test point $(1, 0)$ their squared losses differ by $1/\varepsilon^2$, which is arbitrarily large. Therefore no finite uniform stability bound holds over this input/label class, even for fixed m. The failure is caused by arbitrarily small design singular values, not by choosing among nonunique minimizers.

Structural or analytic derivation

Regularization supplies the lower spectral bound $\lambda I \preceq XX^\top + \lambda I$. At zero regularization a bound is possible only on a restricted class with a uniform positive design-eigenvalue bound, or another hypothesis that controls prediction changes. The counterexample identifies the missing assumption exactly.

Comparative cost and preferred method

One scalar construction disproves a uniform theorem for every sample size. No amount of repeated numerical refitting proves stability over the unrestricted class. This is a necessary-condition result; it gives no speedup claim and does not say that all unregularized regressions are unstable.

8.15 AC-123 | Exact leave-one-out kernel ridge regression

Reference: [8], exercise 11.9.

Source task and conventions

Complete all four parts with a fixed unnormalised regularization convention and the corrected denominator.

Conventional solution and proof

Use the RKHS objective $J_T(h) = \sum_{(x,y)\in T}(h(x)-y)^2 + \lambda\|h\|^2$, $\lambda > 0$. It is strictly convex and coercive, with a unique minimizer. For (a), replace the ith label by $h_{S_i}(x_i)$ (the printed $h_{S_i}(y_i)$ is a type error). Adding the term $(h(x_i) - h_{S_i}(x_i))^2$ to J_{S_i} leaves its minimizer unchanged: the term is nonnegative and is zero there. Thus $h_{S_i} = h_{S_i'}$.

(b) Write $C = (K + \lambda I)^{-1}$ and $H = KC = CK$. The representer calculation, or first-order normal equations in the span of kernel sections, gives fitted values $Hy^{(i)}$ for the replaced-label vector. Symmetry of H gives

$$h_{S_i}(x_i) = e_i^\top H y^{(i)} = (y^{(i)})^\top CKe_i.$$

Let $t_i = h_{S_i}(x_i)$ and $\widehat{y}_i = h_S(x_i)$. Since $y^{(i)} = y + (t_i - y_i)e_i$,

$$t_i = \widehat{y}_i + H_{ii}(t_i - y_i), \qquad t_i - y_i = \frac{\widehat{y}_i - y_i}{1 - H_{ii}}.$$

The denominator is strictly positive: $I - H = \lambda C \succ 0$. Therefore (c) has the corrected expression

$$\boxed{\widehat{R}_{\mathrm{LOO}} = \frac{1}{m}\sum_i \left(\frac{\widehat{y}_i - y_i}{1 - H_{ii}}\right)^2 = \frac{1}{m}\sum_i \left(\frac{\alpha_i}{C_{ii}}\right)^2, \quad \alpha = Cy.}$$

The source prints H_{ii} in the denominator; with $K = 2I, \lambda = 1, y = \mathbf{1}$, the true LOO risk is 1, while that expression is 1/4.

(d) If all diagonal entries equal γ, then $\widehat{R}_{\mathrm{LOO}} = \widehat{R}_S/(1-\gamma)^2$. Since $0 \preceq H \prec I$, $0 \le \gamma < 1$. For positive empirical risk the errors coincide exactly at $\gamma = 0$. If the empirical risk is zero they coincide for every admissible γ; with $\lambda > 0$ and this strictly regularized formulation, zero training residual forces $y = 0$. Under averaged-loss conventions, deletion changes the effective coefficient unless λ is adjusted to keep the unnormalised penalty fixed.

Structural or analytic derivation

An independent derivation partitions $A = K + \lambda I$ around coordinate i. Its Schur complement $s = a - b^\top B^{-1} b$ is positive; block inversion gives $C_{ii} = 1/s$ and $\alpha_i = (y_i - b^\top B^{-1} y_{-i})/s$. The deleted prediction is $b^\top B^{-1} y_{-i}$, so the deleted residual is exactly α_i / C_{ii}. This proves the shortcut without solving m separate systems.

Comparative cost and preferred method

Independent dense refitting uses m factorizations of order $m - 1$, giving $O(m^4)$ arithmetic. One factorization plus all inverse-diagonal entries costs $O(m^3)$, including triangular solves against the identity; storage remains $O(m^2)$. Solving only for α is insufficient: the diagonal must also be computed. The matched benchmark checks every residual against explicit deletion. The identity does not apply exactly to logistic regression or to a preprocessing step that is re-estimated after each deletion without further analysis.

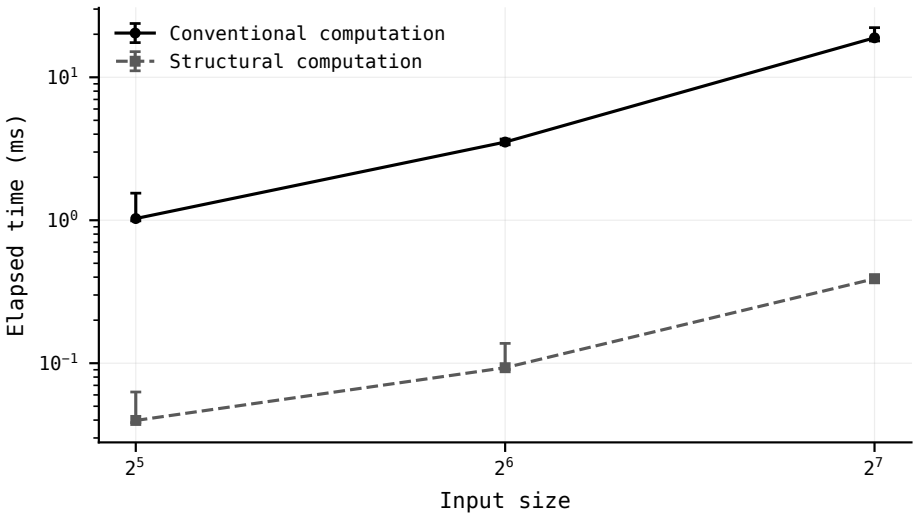


Figure 8.3: Kernel ridge LOO: medians and min/max ranges of seven complete calls; both methods return the same full output. Setup in each implementation is included.

8.16 AC-124 | Birthday collisions with one favoured floor

Reference: [8], exercise D.1.

Source task and conventions

Answer all three elevator-model questions, separating the requested approximation from exact threshold checks.

Conventional solution and proof

For independent passengers with floor probabilities $p_1, \ldots, p_F$, the exact no-collision probability for n passengers is $n!e_n(p_1, \ldots, p_F)$, where e_n is the elementary symmetric polynomial. This follows by summing products over ordered distinct floors. The expected number of colliding pairs is $\Lambda = \binom{n}{2} \sum_j p_j^2$. The usual rare-collision approximation is $P(\text{collision}) \approx 1 - e^{-\Lambda}$; it is not an identity for a heavily favoured floor.

(1) For uniform floors, $P(\text{no collision}) = (F)_n/F^n = \prod_{j=0}^{n-1}(1 - j/F)$. Expanding $\log(1-u) = -u + O(u^2)$ gives $\log P = -n(n-1)/(2F) + O(n^3/F^2)$ when n/F is small. Thus the approximate threshold solves $n(n-1) > 2F \log 2$. For $F = 30$, both the approximation and exact product give $n = 7$: collision probabilities at $n = 6, 7$ are 0.41355556 and 0.53084444.

(2) If one floor has probability q and all others have $p = (1-q)/(F-1)$, at most one passenger may choose the favoured floor in a collision-free sample. Hence

$$P_0(n) = (F-1)_n p^n + nq(F-1)_{n-1} p^{n-1},$$

with a falling factorial equal to zero when too many distinct floors are requested. The pair approximation uses $s_2 = q^2 + (1-q)^2/(F-1)$ and the first integer n satisfying $\binom{n}{2} s_2 > \log 2$. For $F = 30$, $q = 1/4, 7/20, 1/2$, the approximate and exact thresholds are respectively $5, 4, 3$. Exact collision probabilities at those thresholds are $0.51637260, 0.51051914, 0.52556480$; at one fewer passenger they are $0.36576044, 0.32480380, 0.25862069$. With

$q = 1/2, F = 1000, n = 3$ still suffices: the probability is $0.50075050 > 1/2$, while $n = 2$ gives 0.25025025. In the infinite-floor limit the three-passenger probability tends to $1/2$, so strictness matters.

(3) From guard data, estimate destination frequencies with explicit sampling periods, occupancy and missing-data rules; stratify by time, entry floor, group membership and building use. Assess dependence between passengers travelling together. A fitted joint/group model or a validated resampling model is preferable to assuming independent identical draws without checking. No numerical model can be fitted here because no guard observations are supplied.

Structural or analytic derivation

The generating polynomial $\prod_j (1 + p_j z)$ contains every no-collision probability as $n![z^n]$. Its coefficients can be computed by the recurrence $e_k^{(j)} = e_k^{(j-1)} + p_j e_{k-1}^{(j-1)}$. This also gives an exact algorithm for arbitrary nonuniform probabilities and avoids enumerating all F^n assignments.

Comparative cost and preferred method

Coefficient dynamic programming through order n costs $O(Fn)$ arithmetic and $O(n)$ storage, versus F^n assignments in direct enumeration. The special one-favoured-floor formula costs $O(n)$ with running products. A standard birthday-product calculation is already optimal for the uniform model. The rare-collision formula is reported as an approximation and independently checked at each requested threshold.

8.17 AC-125 | Label-bias estimation and an exponential certificate

Reference: [8], exercise D.2.

Source task and conventions

Construct the empirical estimate and prove the stated high-probability bound.

Conventional solution and proof

Let $B_i = \mathbf{1}\{f(X_i) = +1\}$ for i.i.d. $X_i \sim D$ and set $\widehat{p} = m^{-1} \sum_i B_i$. Then $E\widehat{p} = p_+$. For completeness, if $B \sim \text{Bernoulli}(p)$, define $\psi(t) = \log E e^{t(B-p)}$. Exponential tilting makes $\psi''(t)$ a Bernoulli variance, so $0 \le \psi''(t) \le 1/4$. Since $\psi(0) = \psi'(0) = 0$, integration twice gives $\psi(t) \le t^2/8$ for every real t. Independence and Markov's inequality imply

$$P(\widehat{p} - p \ge \varepsilon) \le \inf_{t>0} \exp(-tm\varepsilon + mt^2/8) = e^{-2m\varepsilon^2},$$

using $t = 4\varepsilon$. Apply the same argument to $-B_i$ and the union bound. Thus $P(|\widehat{p} - p| \ge \varepsilon) \le 2e^{-2m\varepsilon^2}$. For $0 < \delta < 1$, choosing $\varepsilon = \sqrt{\log(2/\delta)/(2m)}$ proves the requested confidence statement. Values $\delta \ge 1$ provide no substantive confidence claim.

Structural or analytic derivation

The moment-generating function packages all Bernoulli moments into one convex scalar function. Its curvature bound supplies a uniform tail certificate, while direct binomial summation provides an exact finite-m probability when p is known.

Comparative cost and preferred method

The estimator requires one pass, $O(m)$ time and $O(1)$ storage; an adversarial input can alter an uninspected label, so an exact count requires all labels in the ordinary input model. Exponential-moment analysis improves the distribution-free sample guarantee over variance-only bounds. It does not accelerate the already linear empirical count.

8.18 AC-126 | Optimal testing of two biased coins

Reference: [8], exercise D.3.

Source task and conventions

Prove the five lower-bound and optimality statements and give a sufficient sample size.

Conventional solution and proof

Let P be the m-toss law for coin A, whose head probability is $p = (1-\varepsilon)/2$, and Q that for coin B, whose head probability is $1-p$. Their prior probabilities are equal. The likelihood ratio for a sequence with N heads is

$$P(s)/Q(s) = \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{2N-m}.$$

It exceeds one exactly when $N < m/2$. Selecting the larger posterior at each sequence minimizes its contribution to the total error; summing proves (e), including arbitrary tie choices at $N = m/2$.

(a) For even m, symmetry gives

$$E_m = P_A(N > m/2) + \tfrac{1}{2}P_A(N = m/2) \geq \tfrac{1}{2}P_A(N \geq m/2),$$

which is the requested inequality.

(b) A complete affinity bound gives a slightly stronger result. Let $\rho = \sum_s \sqrt{P(s)Q(s)} = (\sqrt{1-\varepsilon^2})^m$. By Cauchy–Schwarz,

$$\mathrm{TV}(P,Q) = \tfrac{1}{2}\sum_s |\sqrt{P} - \sqrt{Q}|(\sqrt{P} + \sqrt{Q}) \leq \sqrt{1-\rho^2}.$$

The Bayes error is $E_m = (1 - \mathrm{TV})/2$. Since $\log(1-u) > -u/(1-u)$ for $0 < u < 1$ (differentiate their difference, which vanishes at zero),

$$E_m \geq \tfrac{1}{2}\{1 - \sqrt{1-(1-\varepsilon^2)^m}\} > \tfrac{1}{4}\{1 - \sqrt{1-e^{-m\varepsilon^2/(1-\varepsilon^2)}}\}.$$

This proves the displayed bound in (b).

(c) The optimal error cannot increase with more observations, because a rule can ignore the additional toss. For odd m, $E_m \geq E_{m+1}$, and the even-case bound with $m+1$ gives the stated exponent $2\lceil m/2\rceil\varepsilon^2/(1-\varepsilon^2)$. The affinity proof in fact also works directly for odd m.

(d) If $E_m \leq \delta < 1/4$, the strict bound forces

$$2\lceil m/2\rceil > \frac{1-\varepsilon^2}{\varepsilon^2}\log\frac{1}{8\delta(1-2\delta)}.$$

For fixed $\delta < 1/4$ this is $\Omega(\varepsilon^{-2})$; as $\delta \downarrow 0$ it is $\Omega(\varepsilon^{-2}\log(1/\delta))$ in the small-bias regime. This is a necessary bound, not sufficient. Hoeffding applied to the threshold rule gives $E_m \leq e^{-m\varepsilon^2/2}$, so $m \geq 2\varepsilon^{-2}\log(1/\delta)$ is sufficient. Together with (e), the lower bound applies to every decision rule.

Structural or analytic derivation

The sufficient statistic is the count of heads. Grouping the 2^m possible sequences by that count reduces exact error computation to a binomial tail with at most $m+1$ terms. The likelihood-ratio proof establishes that no more elaborate classifier improves the error for the declared equal-prior model.

Comparative cost and preferred method

Exact evaluation costs $O(m)$ recurrence steps for the binomial probabilities, versus 2^m sequence enumeration. Runtime to read the tosses is necessarily $\Omega(m)$ for an exact head count. The lower and upper sample-complexity bounds agree in order; they are not an exact least-integer sample threshold, which can be found from the binomial error.

8.19 AC-127 | Optimizing a tail-to-moment bound

Reference: [8], exercise D.4.

Source task and conventions

Integrate the specified sub-Gaussian tail and resolve the stated range $c > 0$.

Conventional solution and proof

For a nonnegative random variable, Tonelli's theorem applied to $X^2 = \int_0^\infty \mathbf{1}\{X^2 > t\}\, dt$ gives

$$EX^2 = \int_0^\infty P(X^2 > t)dt \le \int_0^\infty \min(1, ce^{-2mt})dt.$$

For $c \ge 1$, split at $u = (\log c)/(2m) \ge 0$. The integral is $u + ce^{-2mu}/(2m) = (\log c + 1)/(2m)$, proving the intended statement. Equivalently, differentiating $u + ce^{-2mu}/(2m)$ on $u \ge 0$ gives its unique minimizer at that point. For $0 < c < 1$, the minimizer over $u \ge 0$ is $u = 0$, and the valid bound is $c/(2m)$. The printed formula for all $c > 0$ is false. Let $X = 0$ with probability $1 - c$ and, with probability c, let $X = \sqrt{Y}$ for $Y \sim \mathrm{Exp}(2m)$. Then $P(X > t) = ce^{-2mt^2}$ for $t > 0$ and $EX^2 = c/(2m) > (1 + \log c)/(2m)$. Thus the sharp consequence of the given tail envelope is

$$\boxed{EX^2 \le \begin{cases} c/(2m), & 0 < c \le 1, \\ (1 + \log c)/(2m), & c \ge 1. \end{cases}}$$

Both branches are attainable by a distribution whose survival function is the displayed minimum.

Structural or analytic derivation

The probability cap $P \le 1$ is a constraint in the scalar optimization. Retaining it determines the breakpoint $c = 1$ and supplies an extremal distribution, which proves optimality of the bound within the stated tail class.

Comparative cost and preferred method

One analytic integration and a one-variable constrained minimum yield the best possible bound from the supplied envelope. Numerical quadrature would add approximation error and would not by itself certify optimality. This is an exact extremal result, not a measured speedup.

8.20 AC-128 | Hoeffding versus Chebyshev with the missing indepen-

Reference: [8], exercise D.5.

Source task and conventions

Compare the two bounds and produce the requested $\varepsilon = .05$ plot.

Conventional solution and proof

As printed, the source assumes only common mean and variance. Independence is needed for the usual $1/m$ variance reduction and Hoeffding product argument. Indeed, setting every $X_i = B$ for one Bernoulli(1/2) variable gives $P(|\bar{X} - 1/2| > .05) = 1$ for every m, contradicting the proposed exponential decay. The following solves the intended independent bounded-variable version explicitly.

(a) Independence gives $\mathrm{Var}\,\bar{X} = \sigma^2/m$. Markov's inequality applied to $(\bar{X} - \mu)^2$ gives

$$P(|\bar{X} - \mu| > \varepsilon) \le \min\{1, \sigma^2/(m\varepsilon^2)\}.$$

For $X \in [0,1]$, the tilted distribution still lies in $[0,1]$ and has variance at most $1/4$: $\operatorname{Var} X \le E(X-1/2)^2 \le 1/4$. The log-mgf curvature proof used in the label-bias solution therefore gives Hoeffding's bound $\min\{1, 2e^{-2m\varepsilon^2}\}$. Before clipping at one, Chebyshev is tighter exactly when

$$\sigma^2 < 2m\varepsilon^2 e^{-2m\varepsilon^2}.$$

After clipping, this strict comparison additionally requires the Chebyshev expression to be below one; otherwise the two displayed guarantees may simply tie at one.

(b) For Bernoulli variables $\sigma^2 = p(1-p) = 1/4 - (p-1/2)^2 \le 1/4$. At $\varepsilon = .05$, plot

$$C_m = \min(1, 100/m), \qquad H_m = \min(1, 2e^{-m/200}).$$

The enclosed graph uses integer $m = 1, \dots, 3000$ and a logarithmic vertical scale. It shows a finite moderate-m range in which the variance bound is smaller, followed by exponential superiority for sufficiently large m. With dependence retained, the general Chebyshev bound must instead use the actual covariance sum $m^{-2}\sum_{i,j}\operatorname{Cov}(X_i, X_j)$; the common-variance assumption alone only bounds it by σ^2.

Structural or analytic derivation

The comparison is a comparison of certified tail envelopes at the same tolerance. Independence is a mathematical input, and the counterexample proves that it cannot be introduced implicitly.

Comparative cost and preferred method

Both bound evaluations are $O(1)$ for a specified m. Their useful difference is sample complexity: the worst-Bernoulli Chebyshev guarantee requires $m \ge 1/(4\delta\varepsilon^2)$, while Hoeffding requires $m \ge \log(2/\delta)/(2\varepsilon^2)$. Neither is uniformly the smaller bound for all known variances and sample sizes.

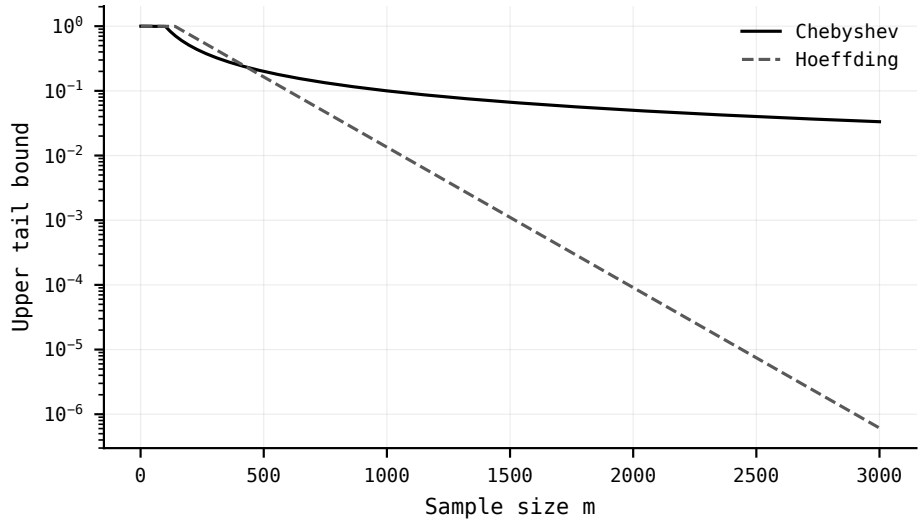


Figure 8.4: Independent Bernoulli observations at tolerance 0.05. The clipped bounds can exchange order; dependence invalidates the source's unqualified formulas.

8.21 AC-129 | Random projections, PCA and MNIST nearest-neighbour overlap

Reference: [8], exercise 15.6.

Source task and conventions

Complete parts (a)–(f) for the first 2,000 original MNIST test images: exact neighbour sets, Gaussian projections, overlap scores and PCA comparisons.

Conventional solution and complete derivation

Data and exact reference sets. Part (a) specifies the first $m = 2000$ test images, with all digit labels retained, not a binary classifier or a training subset. Parse the unsigned-byte IDX header, verify magic number 2051, count and shape 28×28, and place image i in column $x_i \in \mathbb{R}^{784}$ of X. The program uses a points-by-features transpose internally and records both conventions. For part (b), compute the squared distances $d_{ij}^2 = \sum_{r=1}^{784}(X_{ri} - X_{rj})^2$. Exclude $j = i$, order candidates by (d_{ij}^2, j) to specify ties, and retain the first 10 and 50 indices. Squared distance gives the same order as distance because the square root is increasing on nonnegative numbers. On the original integer pixels, the sums are at most $784 \cdot 255^2 < 2^{53}$ and are exactly representable in binary64 when accumulated by direct squared differences.

Gaussian projection and scores. For (c), generate $A \in \mathbb{R}^{100\times 784}$ with independent standard normal entries, set $\widetilde{X} = AX$, and repeat the ten-neighbour computation with the same self-exclusion and tie rules. The delivered program uses a recorded seed and common prefixes of one 500-row Gaussian matrix for the requested dimensions. Dividing A by $\sqrt{k}$ would not change any projected neighbour ordering, but the implementation retains the source's unscaled matrix. For (d), compute exactly

$$s_{10}(k) = \frac{1}{2000}\sum_i |N_{i,10} \cap \widetilde{N}_{i,10}|, \qquad s_{50}(k) = \frac{1}{2000}\sum_i |N_{i,50} \cap \widetilde{N}_{i,10}|.$$

Each summand in both scores is between zero and ten, and $s_{10}(k) \le s_{50}(k) \le 10$. The second score still uses *ten* projected neighbours; changing it to a fifty-versus-fifty comparison would answer a different question. Sets and every per-observation intersection count are saved, so both reported means are exactly reproducible from the integer outputs.

Dimension sweep and PCA. Part (e) repeats this construction for dimensions k in the set $\{1, 10, 50, 100, 250, 500\}$ and plots both scores against k. The program generates both plots from the actual results. For (f), centre the image columns, compute $X_c = U\Sigma V^{\mathsf{T}}$, and project with $U_k^{\mathsf{T}} X_c$. The same neighbour and score procedures then produce the PCA curves. Centring translates all original images equally and therefore leaves their pairwise distances unchanged. A principal basis is computed once, and all six ranks use its prefixes; all observations participate in this unsupervised empirical representation, as requested by the exercise.

PCA minimizes $\|(I - P)X_c\|_F^2$ among rank-k orthogonal projections: in an eigenbasis of $X_c X_c^{\mathsf{T}}$, $\operatorname{tr}(PX_cX_c^{\mathsf{T}}) = \sum_j \sigma_j^2 p_{jj}$, with $0 \le p_{jj} \le 1$ and $\sum p_{jj} = k$. Moving weight to the k largest eigenvalues maximizes that sum and proves the minimum residual $\sum_{j>k}\sigma_j^2$. This is an average reconstruction optimum, not a theorem that PCA maximizes nearest-neighbour overlap. Likewise Gaussian projections have a probabilistic distance-preservation guarantee, not an exact monotonicity guarantee for one sampled sequence of overlap scores. Greater dimension generally retains more distance information; the relative measured quality of PCA and Gaussian projection depends on the data's variance structure and local neighbour gaps. A claimed numerical winner must be obtained from the actual curves.

The original MNIST test image file was unavailable during this execution. The volume consequently supplies the complete original-data program and the mathematical analysis, with no invented overlap values or plots. The program is executed on a separately identified synthetic validation matrix to verify shapes, tie handling, set inclusions and score ranges; those values are

not substituted for the requested MNIST result. A source-specific empirical completion consists of running the supplied program with the original test file and retaining its two generated plots and records.

Structural or analytic formulation

A quantitative analytic certificate explains when a Gaussian projection preserves the source scores. For a fixed nonzero displacement v, $\|Av\|^2/\|v\|^2$ is chi-square with k degrees of freedom. Its moment-generating function is $(1-2t)^{-k/2}$ for $t < 1/2$: the one-dimensional identity follows by completing the square in a Gaussian integral, and independence multiplies the factors. Chernoff's bound with $t = \epsilon/[2(1+\epsilon)]$ gives the upper tail $\exp\{-k[\epsilon - \log(1+\epsilon)]/2\}$. Applying the negative-parameter version gives lower tail $\exp\{-k[-\epsilon - \log(1-\epsilon)]/2\}$. Integrating the geometric series, or differentiating the difference, shows for $0 < \epsilon < 1$ that both are bounded by $\exp\{-k(\epsilon^2/4 - \epsilon^3/6)\}$. A union bound over the $m(m-1)/2$ image pairs therefore bounds the probability that any squared distance leaves its $(1 \pm \epsilon)$ interval after division by k by

$$m(m-1)\exp\{-k(\epsilon^2/4 - \epsilon^3/6)\}.$$

No independence among pairwise-distance events is needed for this union bound. The bound is clipped at one if necessary. Let $d_{i,(j)}$ be the ordered original distances. On this distance-preservation event, if $(1-\epsilon)d^2_{i,(11)} > (1+\epsilon)d^2_{i,(10)}$, every original top-ten point remains closer than every outsider, so its overlap is exactly ten. If instead $(1-\epsilon)d^2_{i,(51)} > (1+\epsilon)d^2_{i,(10)}$, at least ten points inside the true fifty are projected closer than every outside point, giving the fifty-versus-ten overlap of ten. These inequalities prove the corresponding row-level score guarantees and explain why ties or small gaps can disrupt neighbour recovery despite small relative distance error. The sum of pairwise retained squared distances is $2m\|PX_c\|_F^2$; expansion of the double sum and $\sum_i x_{c,i} = 0$ prove this identity. Hence PCA optimizes total retained pairwise energy, but that objective does not order the individual neighbour gaps.

Comparative efficiency and scope

Direct original distances cost $O(m^2N)$ arithmetic and $O(m^2)$ storage if retained. After a dense Gaussian map, construction costs $O(mNk)$ and distances cost $O(m^2k)$, with $O(mk+m^2)$ working storage. The pair-distance arithmetic alone is reduced by N/k, namely $784/k$ here; projection setup and sorting must also be counted. The supplied fully sorted, deterministic tie rule costs $O(m^2 \log m)$ comparisons. A partial-selection implementation can reduce selection work while preserving its explicit tie contract. For $m = 2000$, one dense binary64 distance matrix uses $32,000,000$ bytes. The required original neighbour references are computed only once and reused for all twelve projected comparisons. A classical dense SVD costs $O(mN\min(m,N))$, followed by $O(mNk)$ projected coordinates; all ranks reuse that SVD. This setup can dominate small numbers of queries. No runtime or overlap superiority of PCA is claimed without its actual original-data run. The exact gap certificate and the proved operation/storage counts remain valid independently of that missing measurement.

Source qualification. This is exercise 15.6 on random projection, PCA and neighbours. No 3-versus-8 classification or SVM exercise is substituted.

Chapter 9

The Elements of Statistical Learning

Trevor Hastie, Robert Tibshirani, Jerome Friedman. Reference problems and derived analyses.

9.1 AC-130 | Population structure and the complete 200-observation experiment

Reference: [9], exercise 14.15.

Source task and conventions

Generate the prescribed Gaussian observations and compare the leading principal and factor-analysis directions.

Conventional solution and complete derivation

Let Z_1, Z_2, Z_3 be independent $N(0,1)$ and set $X_1 = Z_1$, $X_2 = Z_1 + .001Z_2$, $X_3 = 10Z_3$. The population covariance is

$$\Sigma = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1+10^{-6} & 0 \\ 0 & 0 & 100 \end{pmatrix}.$$

Its upper-block eigenvalues are $[2 + 10^{-6} \pm \sqrt{4 + 10^{-12}}]/2$, both less than 100. Hence the leading population principal component is exactly e_3. For any unit vector, its variance is the Rayleigh quotient of Σ; diagonalizing the upper block proves no other unit direction exceeds 100. By contrast, the factor representation is $X = Lf + \eta$ with $L = (1,1,0)^{\mathsf{T}}$, $\operatorname{Var} f = 1$, and $\operatorname{Cov} \eta = \operatorname{diag}(0, 10^{-6}, 100)$. Independence of the three original Gaussians proves the required independence and covariance properties. The factor loads on the common component $X_1 + X_2$; the large third-coordinate variance is unique noise in this model. The supplied execution generates 200 rows using NumPy's generator seed 20261006, centres them, and computes an SVD. Its leading sample principal direction is $(-0.01053713, -0.01053279, 0.99988901)^{\mathsf{T}}$. A one-factor Gaussian fit in unit-variance coordinates, using deterministic LAPACK SVD with tolerance 10^{-4}, takes 33 iterations. After transforming to the original units its loading is $(0.95598424, 0.95594867, -1.09494615)^{\mathsf{T}}$, and its uniqueness variances are approximately $(4.85811 \times 10^{-7}, 4.85802 \times 10^{-7}, 99.9536660)$. The standardized loadings are approximately $(0.9999997, 0.9999997, -0.108869)^{\mathsf{T}}$. Thus the sample common-factor variance fraction for the third variable is about $0.108869^2 = 1.19\%$, while the fractions for the first two are nearly one. A raw loading, standardized loading and posterior

factor-score coefficient are different objects. The apparent nonzero third loading reflects accidental finite-sample correlation. It does not contradict the exact population conclusion. The fitting routine reports a numerical likelihood stationary estimate; a global finite-sample likelihood optimum is not asserted. The full sample, loadings, uniquenesses and likelihood history are supplied.

Structural or analytic formulation

PCA maximizes total Euclidean variance; factor analysis separates shared covariance from diagonal uniqueness. The explicit decomposition proves the distinction before sampling. The subsequent computation tests the effect in the prescribed finite sample, using the same observations for both models. The original draft's 12-iteration figure is superseded by the reproducible deterministic 33-iteration run recorded here.

Comparative efficiency and scope

The population calculation reduces to a 2×2 eigenproblem and a separated scalar variance. The 200-by-3 sample is small; no meaningful asymptotic runtime advantage is claimed between its two fitted models. The improvement is correspondence between the chosen objective and common latent structure, not universal statistical superiority.

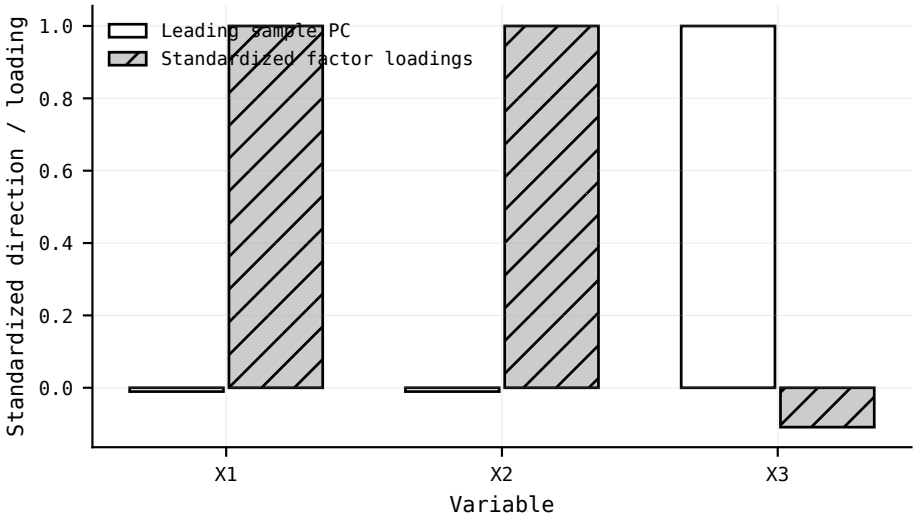


Figure 9.1: Population and finite-sample objectives differ: the PC follows the high-variance third coordinate; standardized common-factor loadings chiefly select the first two.

9.2 AC-131 | Kernel-PCA coordinates and out-of-sample centring

Reference: [9], exercise 14.16.

Source task and conventions

Prove rank, coordinate and new-point formulas, retaining the required centring convention.

Conventional solution and proof

Let $J = I - \mathbf{1}\mathbf{1}^\top/N$ and $K_c = JKJ = UD^2U^\top$, keeping only r positive eigenvalues d_m^2. The feature-centred data operator has SVD with these singular values, so exactly $r = \operatorname{rank} K_c$ nonzero principal components occur. Without centring this is $\operatorname{rank} K$; with centring it need not be. A constant kernel has rank one before centring and zero after centring. Set $\alpha_m = u_m/d_m$. Then $K_c\alpha_m = d_m u_m = z_m$, establishing the coordinate formula. To see its meaning, write $K_c = \Phi_c\Phi_c^*$, where the rows of Φ_c are the centred feature vectors. The feature-space direction $v_m = \Phi_c^* u_m/d_m$ has norm one, and $\Phi_c v_m = d_m u_m$. A new point has coordinate $z_{0m} = \langle \phi(x_0) - \bar{\phi}, v_m\rangle = k_c(x_0)^\top u_m/d_m$, where

$$[k_c(x_0)]_i = K(x_i, x_0) - \frac{1}{N}\sum_j K(x_j, x_0) - \frac{1}{N}\sum_j K(x_i, x_j) + \frac{1}{N^2}\sum_{j,l} K(x_j, x_l).$$

All terms use the training mean. Zero eigenvalues are omitted, so no division by zero is performed. Eigenvalue multiplicities permit rotations within the corresponding principal subspace.

Structural or analytic derivation

This is the finite-vector/SVD argument applied to an implicit Hilbert-space feature map. The kernel supplies the inner products needed by the projection without constructing all feature coordinates. The new-point centring formula is necessary to preserve the same affine origin.

Comparative cost and preferred method

A finite $N \times N$ eigensystem replaces a possibly infinite-dimensional covariance operator. Dense cost is $O(N^3)$ with $O(N^2)$ storage; a new projection onto r components requires N kernel evaluations and $O(Nr)$ arithmetic. If an explicit low-dimensional feature representation is available, its ordinary PCA can be cheaper. The rank reduction is exact for the empirical feature span.

9.3 AC-132 | Variational proof of kernel principal functions

Reference: [9], exercise 14.17.

Source task and conventions

Solve the RKHS constrained-variance problem and its successive orthogonal versions.

Conventional solution and proof

Let $\phi_i = K(\cdot, x_i)$, $\bar{\phi} = N^{-1}\sum_i \phi_i$, and define the empirical covariance operator

$$Tg = \frac{1}{N}\sum_i \langle g, \phi_i - \bar{\phi}\rangle(\phi_i - \bar{\phi}).$$

It is self-adjoint, positive, finite rank, and $\mathrm{Var}_T\, g(X) = \langle g, Tg\rangle$. Components orthogonal to the centred feature span contribute no variance. On that span the nonzero eigenvalues of T are d_m^2/N, where $JKJ = UD^2U^\top$. Define

$$g_m = \frac{1}{d_m}\sum_i u_{im}(\phi_i - \bar{\phi}) = \frac{1}{d_m}\sum_i u_{im}\phi_i.$$

The second equality holds because positive-eigenvalue u_m is orthogonal to $\mathbf{1}$. Direct inner products give $\langle g_l, g_m\rangle = \delta_{lm}$ and $Tg_m = (d_m^2/N)g_m$. For any unit $g = \sum_m a_m g_m + g_\perp$, Pythagoras gives $\sum_m a_m^2 \le 1$ and

$$\langle g, Tg\rangle = \frac{1}{N}\sum_m d_m^2 a_m^2 \le d_1^2/N.$$

Equality is attained by g_1, proving $\widehat{c}_i = u_{i1}/d_1$. Under orthogonality to $g_1, \ldots, g_{m-1}$, the same calculation bounds the variance by d_m^2/N and gives g_m. Repeated eigenvalues give nonunique bases; when all centred eigenvalues vanish, the variance is zero for every unit function and the displayed division formula has no nonzero component to represent.

Structural or analytic derivation

The coefficient calculation can also be written as maximizing $c^\top K_c^2 c/N$ subject to $c^\top K_c c = 1$, after discarding null directions. Diagonalizing K_c reduces this to a weighted average of its eigenvalues. The operator proof additionally explains the full RKHS and the unobserved orthogonal complement.

Comparative cost and preferred method

The empirical problem is exactly finite-dimensional. The eigenproblem supplies all mutually orthogonal components in one factorization, with no search over functions. This is the conventional kernel-PCA solution; the functional-analytic perspective proves its global optimality and the exceptional zero-variance case.

9.4 AC-133 | Reproducing identities and the finite representer theorem

Reference: [9], exercise 5.15.

Source task and conventions

Prove all four RKHS identities and the equality condition.

Conventional solution and proof

Use the source expansion $K(x,y) = \sum_{j:\gamma_j>0} \gamma_j \phi_j(x)\phi_j(y)$, and write $f = \sum_j c_j \phi_j$ with $\|f\|_H^2 = \sum_j c_j^2/\gamma_j < \infty$. Zero-eigenvalue terms are omitted. At each evaluation point assume the stated RKHS kernel is finite, $K(x,x) < \infty$. Cauchy–Schwarz gives

$$\sum_j |c_j \phi_j(x)| \le \left(\sum_j c_j^2/\gamma_j\right)^{1/2} \left(\sum_j \gamma_j \phi_j(x)^2\right)^{1/2} < \infty.$$

Thus the evaluation and the following inner products are justified. (a) The coefficients of $K(\cdot, x_i)$ are $\gamma_j \phi_j(x_i)$, giving $\langle K(\cdot,x_i), f\rangle_H = \sum_j c_j \phi_j(x_i) = f(x_i)$. (b) Apply (a) to $f = K(\cdot, x_j)$. (c) Bilinearity for the finite expansion $g = \sum_i \alpha_i K(\cdot, x_i)$ gives $J(g) = \|g\|_H^2 = \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$. (d) Orthogonality implies $\rho(x_i) = 0$ by (a), and $\langle g, \rho\rangle = 0$. Hence all loss terms are unchanged and $J(g+\rho) = J(g) + \|\rho\|^2$. The difference of objectives is exactly $\lambda\|\rho\|^2$. For $\lambda > 0$ it is nonnegative, with equality exactly when $\rho = 0$. At $\lambda = 0$ the source's exclusive equality condition would fail.

Structural or analytic derivation

Let S be the finite span of the evaluation representers. Every f has an orthogonal decomposition $g + \rho$ with $g \in S$, and part (d) shows that any attained minimizer with positive regularization lies in S. Convexity of the loss is unnecessary for this reduction; existence or uniqueness of a minimizer still requires its own assumptions.

Comparative cost and preferred method

An optimization over a Hilbert space reduces exactly to at most N coefficients. The reduction preserves every sampled loss and decreases the penalty by a known nonnegative quantity. It is an exact dimension and optimality statement, not an accuracy assertion about an arbitrary finite truncation. The squared-eigenvalue summability alone must not be used as a replacement for finite point-evaluation norms.

9.5 AC-134 | Centring with an unpenalized intercept

Reference: [9], exercise 3.5.

Source task and conventions

Derive the centred ridge and lasso formulations and their coefficient correspondence.

Conventional solution and proof

Write $Z_{ij} = x_{ij} - \bar{x}_j$. The bijection between parameters is $\beta_j^c = \beta_j$ for slopes and $\beta_0^c = \beta_0 + \sum_j \bar{x}_j \beta_j$. It gives the same fitted value $\beta_0 + x_i^\top \beta = \beta_0^c + z_i^\top \beta^c$ and leaves the slope penalty unchanged. Because every centred column sums to zero, differentiating with respect to the unpenalized intercept yields $\widehat{\beta}_0^c = \bar{y}$. The ridge slopes solve

$$(Z^\top Z + \lambda I)\widehat{\beta}^c = Z^\top (y - \bar{y}\mathbf{1}),$$

so for $\lambda > 0$ the unique slopes are its inverse applied to the right side. The original intercept is $\bar{y} - \bar{x}^\top \widehat{\beta}^c$. Replacing the squared penalty by $\lambda \sum_j |\beta_j|$ preserves the same bijection and the same intercept calculation. The lasso slopes are the minimizers of the centred penalized criterion; they need not be unique without further design assumptions. Penalizing the intercept would invalidate this equivalence as stated.

Structural or analytic derivation

The centring matrix J is the orthogonal projector off the constant vector. Eliminating the intercept gives $\min_\beta \|Jy - JX\beta\|^2 + \lambda P(\beta)$ with the original slope penalty P. This is an exact elimination of a nuisance coordinate.

Comparative cost and preferred method

Centring costs $O(Np)$ once. It removes one unpenalized variable from each subsequent fit and simplifies the normal equations. It does not change the asymptotic cost of a dense slope solve. Reusing centring across a cross-validation split requires the same affine model and unpenalized intercept; data-dependent scaling has separate requirements.

9.6 AC-135 | Gaussian posterior and ridge regression

Reference: [9], exercise 3.6.

Source task and conventions

Compute the full posterior and identify its mean, mode and regularization parameter.

Conventional solution and proof

Assume the source covariance convention $\beta \sim N(0, \tau I)$, $\tau > 0$, and $y \mid \beta \sim N(X\beta, \sigma^2 I)$, $\sigma^2 > 0$. Up to a constant, minus twice the posterior log density is

$$\sigma^{-2}\|y - X\beta\|^2 + \tau^{-1}\|\beta\|^2.$$

Let $A = X^\top X/\sigma^2 + I/\tau \succ 0$ and $b = X^\top y/\sigma^2$. Completing the square gives $\beta^\top A\beta - 2b^\top\beta = (\beta - A^{-1}b)^\top A(\beta - A^{-1}b) - b^\top A^{-1}b$. Thus

$$\beta \mid y \sim N(A^{-1}b, A^{-1}), \qquad E(\beta \mid y) = \text{mode}(\beta \mid y) = (X^\top X + \lambda I)^{-1}X^\top y, \quad \lambda = \sigma^2/\tau.$$

The mean equals the mode because the Gaussian density is symmetric about its centre and its positive-definite quadratic exponent is uniquely minimized there. If the prior is parameterized with covariance $\tau^2 I$, the denominator is τ^2 instead; that is not the source convention.

Structural or analytic derivation

The posterior precision is the sum of data precision and prior precision. This makes the regularizing effect transparent even when X is rank deficient, because I/τ remains positive definite.

Comparative cost and preferred method

One factorization gives the posterior mean and permits covariance actions. Forming the entire posterior covariance has additional $O(p^3)$ classical work and $O(p^2)$ output storage. Completing the square is already the standard optimal derivation; an analytic reinterpretation supplies no separate computational gain.

9.7 AC-136 | Ridge as augmented least squares

Reference: [9], exercise 3.12.

Source task and conventions

Prove the augmentation identity and its computational implications.

Conventional solution and proof

For centred data and $\lambda \geq 0$, define $\widetilde{X} = \begin{pmatrix} X \\ \sqrt{\lambda} I_p \end{pmatrix}$ and $\widetilde{y} = \begin{pmatrix} y \\ 0 \end{pmatrix}$. Then

$$\|\widetilde{y} - \widetilde{X}\beta\|^2 = \|y - X\beta\|^2 + \lambda\|\beta\|^2.$$

The objectives agree for every coefficient vector, so their minimizer sets agree. If $\lambda > 0$, $\widetilde{X}$ has full column rank, and its normal equations are $(X^\top X + \lambda I)\beta = X^\top y$. If $\lambda = 0$, augmentation adds zero rows and does not remove any nonuniqueness. An unpenalized intercept must be excluded from the identity block, or eliminated by centring.

Structural or analytic derivation

The artificial observations encode a quadratic penalty as Euclidean residuals. Orthogonal QR on the augmented matrix computes the same solution without explicitly forming $X^\top X$, avoiding the condition-number squaring of normal equations.

Comparative cost and preferred method

Naively storing the added rows costs p^2 entries; structured implementations can account for the identity without materializing it. QR costs $O((N + p)p^2)$ classically. A dual solve is preferable when $p \gg N$, while augmentation can improve numerical robustness in other regimes. Objective equality, not a universal speedup, is the proved result.

9.8 AC-137 | Orthogonal subset, ridge and lasso estimators

Reference: [9], exercise 3.16.

Source task and conventions

Derive every entry of Table 3.4, including ties and penalty normalization.

Conventional solution and proof

Assume $X^\top X = I$ and set $b = X^\top y$, the unrestricted least-squares coefficients. Orthogonal decomposition gives

$$\|y - X\beta\|^2 = \|y - Xb\|^2 + \sum_j (\beta_j - b_j)^2.$$

For best subset of size at most M, fixing a support makes $\beta_j = b_j$ on that support and zero elsewhere. The residual reduction is $\sum_{j\in S} b_j^2$, maximized by selecting M largest absolute coefficients. At a boundary tie, choose any required number of tied coordinates; the table's indicator with $\geq$ can include too many unless ties are resolved.

For ridge with objective SSE$+\lambda\sum_j \beta_j^2$, each coordinate minimizes $(\beta_j - b_j)^2 + \lambda\beta_j^2$, giving $\widehat{\beta}_j = b_j/(1+\lambda)$ by setting its derivative to zero; the second derivative is positive.

For the lasso normalization $\frac{1}{2}\text{SSE} + \lambda\sum_j |\beta_j|$, a positive solution has derivative $\beta_j - b_j + \lambda = 0$, and a negative solution has derivative $\beta_j - b_j - \lambda = 0$. At zero the subgradient condition is $|b_j| \leq \lambda$. Thus

$$\widehat{\beta}_j = \operatorname{sign}(b_j)(|b_j| - \lambda)_+.$$

These three cases exhaust the convex one-dimensional problem. With SSE$+\lambda\|\beta\|_1$ the threshold is $\lambda/2$; Table 3.4 absorbs this scaling into its tuning parameter.

Structural or analytic derivation

Orthogonality converts the objective into independent scalar problems. Hard selection, linear shrinkage and soft thresholding follow from three different feasible-set/penalty contracts; none can be substituted for another without changing the optimization problem.

Comparative cost and preferred method

After b is formed, ridge and lasso require $O(p)$ scalar operations. Subset selection can use a linear-time order-statistic algorithm plus a pass, or sorting at $O(p \log p)$, instead of enumerating $\binom{p}{M}$ supports. The orthogonal-design assumption is indispensable; general best-subset regression does not decompose this way.

9.9 AC-138 | Norm monotonicity: ridge, lasso and PLS

Reference: [9], exercise 3.19.

Source task and conventions

Prove the ridge property and determine the corresponding Euclidean-norm claims for lasso and successive PLS steps.

Conventional solution and proof

For $X = UDV^\top$ and $c = U^\top y$, ridge coefficients are $\beta_\lambda = \sum_j [d_j c_j/(d_j^2 + \lambda)]v_j$. Therefore

$$\|\beta_\lambda\|_2^2 = \sum \frac{d_j^2 c_j^2}{(d_j^2 + \lambda)^2},$$

which is nonincreasing in λ, and strictly decreases whenever a nonzero fitted direction is present. Its limit at zero is the minimum-norm least-squares coefficient norm.

For lasso, the ℓ_1 norm is monotone as regularization weakens, but the Euclidean norm need not be. Here is a standardized-Gram counterexample for objective $\frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$:

$$G = X^\top X = \begin{pmatrix} 1 & .8 & .8 \\ .8 & 1 & .3 \\ .8 & .3 & 1 \end{pmatrix}, \quad c = X^\top y = (3.7, 3, 3)^\top.$$

The leading principal minors are $1, .36, .014 > 0$, so $G \succ 0$, and $G^{-1}\mathbf{1} = (-15, 10, 10)^\top$. On $.1 \le \lambda \le .11$, the positive vector

$$\beta_\lambda = (2 + 15(\lambda - .1),\ 1 - 10(\lambda - .1),\ 1 - 10(\lambda - .1))^\top$$

satisfies $G\beta_\lambda - c + \lambda\mathbf{1} = 0$, hence is the unique lasso minimizer. Its squared norm is 6.2425 at .11 and 6 at .1: weakening the penalty decreases the Euclidean norm. Such a centred unit-column design exists by taking $X = QG^{1/2}$ with $Q^\top Q = I$, $Q^\top\mathbf{1} = 0$, and $y = XG^{-1}c$. Monotonicity of the ℓ_1 norm follows by adding the two optimality inequalities at two penalty values.

For univariate-response PLS with exact arithmetic, the coefficients are the conjugate-gradient iterates for $G\beta = c$, started at zero, in the Krylov spaces generated by c. Restrict to $\operatorname{ran} G$ if necessary. Write $r_k = c - G\beta_k$, $p_0 = r_0$, $a_k = \|r_k\|^2/(p_k^\top G p_k) > 0$,

$$\beta_{k+1} = \beta_k + a_k p_k, \quad r_{k+1} = r_k - a_k G p_k, \quad p_{k+1} = r_{k+1} + b_k p_k, \quad b_k = \|r_{k+1}\|^2/\|r_k\|^2 \ge 0.$$

For completeness these recurrences give orthogonal residuals and G-conjugate directions by induction. The step size makes $r_{k+1} \perp p_k$; previous conjugacy gives orthogonality to all earlier directions. For $j < k$, $Gp_j = (r_j - r_{j+1})/a_j$ gives $r_{k+1}^\top G p_j = 0$. For $j = k$, the contribution $-\|r_{k+1}\|^2/a_k$ is cancelled by $b_k p_k^\top G p_k$. This closes the induction. In particular $r_k \perp \beta_k$, and

$$\beta_k^\top p_k = b_{k-1}\{\beta_{k-1}^\top p_{k-1} + a_{k-1}\|p_{k-1}\|^2\} \ge 0$$

starting from zero. Hence $\|\beta_{k+1}\|^2 - \|\beta_k\|^2 = 2a_k\beta_k^\top p_k + a_k^2\|p_k\|^2 > 0$ until convergence. The norm property therefore holds for this standard PLS sequence, although its individual spectral factors need not all be shrinkage factors.

Structural or analytic derivation

Ridge diagonalization proves monotonicity termwise; lasso needs its own norm and active-set analysis; PLS uses Krylov orthogonality. The three estimators cannot be placed under one unproved scalar-shrinkage argument.

Comparative cost and preferred method

The ridge path can be evaluated after one SVD, with $O(r)$ spectral work per parameter and $O(pr)$ work if full coefficient vectors are required. The lasso counterexample is exact rational arithmetic. The PLS proof concerns iteration count and norm, not a guarantee that every further step improves prediction. No universal preference among the three statistical procedures follows.

9.10 AC-139 | Duplicate predictors and the effective ridge penalty

Reference: [9], exercise 3.29.

Source task and conventions

Derive the two-copy coefficients and prove equality for any number of identical predictors.

Conventional solution and proof

Assume $\lambda > 0$ and centred one-predictor data. Put $s = X^\top X$ and $c = X^\top y$, so the original coefficient is $a = c/(s+\lambda)$. With two identical columns, strict convexity and exchange symmetry force $b_1 = b_2 = b$. Differentiation gives $s(2b) - c + \lambda b = 0$, hence

$$\boxed{b = \frac{c}{2s+\lambda} = a\frac{s+\lambda}{2s+\lambda}.}$$

It is generally not $a/2$. More generally, for m duplicate coefficients with aggregate $t = \sum_j b_j$, the prediction depends only on t and

$$\sum_j b_j^2 \geq t^2/m,$$

by Cauchy–Schwarz, with equality exactly when every $b_j = t/m$. This remains true in the presence of other predictors because their coefficients can be held fixed in this argument. Every ridge minimizer therefore has equal duplicate coefficients. In the one-variable case $b_j = c/(ms+\lambda)$ and $t = c/(s+\lambda/m)$. At $\lambda = 0$, equality is a minimum-norm choice but is not required of every least-squares minimizer.

Structural or analytic derivation

Duplicating a feature divides its effective quadratic penalty by the duplication count when aggregate prediction is held fixed. The norm-minimization identity gives the result directly and identifies why duplicate columns change the fitted aggregate.

Comparative cost and preferred method

Identical columns can be compressed to one aggregate variable with penalty λ/m, reducing a redundant block from m parameters to one exactly. Naive duplicate removal with unchanged penalty solves a different problem. The benefit is an exact reduction in linear-system dimension when duplicates are known.

9.11 AC-140 | Feature-space and kernel-space ridge equivalence

Reference: [9], exercise 5.16.

Source task and conventions

Derive the source basis change, explain its identifiability, and prove the fitted-value identity.

Conventional solution and proof

Let $h(x) \in \mathbb{R}^M$ be the stated feature vector and fix the measure underlying the source's orthonormal eigenfunctions. Define $T = \int h(x)h(x)^\top \, d\mu(x)$ and diagonalize $T = VD_\gamma V^\top$. For $\gamma_j > 0$, put $\phi_j(x) = v_j^\top h(x)/\sqrt{\gamma_j}$. Then $\int \phi_j \phi_l \, d\mu = \delta_{jl}$ and

$$h(x) = VD_\gamma^{1/2}\phi(x), \qquad K(x,y) = \sum_j \gamma_j \phi_j(x)\phi_j(y),$$

on the nonnull feature span under the Mercer representation. Null directions are omitted; equality there is understood with the regularity/point-evaluation assumptions of the source. The kernel integral operator determines the γ_j and ϕ_j. Relative to a specified feature map, $v_j = \gamma_j^{-1/2}\int h(x)\phi_j(x)\, d\mu(x)$. The kernel alone does not determine the orientation V of an unspecified feature map: replacing h by an orthogonal rotation leaves K unchanged. Nor do N sample kernel values determine a global Mercer spectrum without a measure.

For coefficients β, let $c = D_\gamma^{1/2}V^\top\beta$. Then $\beta^\top h(x) = c^\top\phi(x)$ and the least attainable coefficient penalty is $\|\beta\|^2 = \sum_{\gamma_j>0} c_j^2/\gamma_j$, with null coefficients set to zero. Substitution proves equivalence of the source criteria (5.53) and (5.63).

For (b), the finite evaluation matrix is $H \in \mathbb{R}^{N\times M}$ and $K = HH^\top$. With $\lambda > 0$, $(H^\top H+\lambda I)H^\top = H^\top(K+\lambda I)$, hence

$$\widehat{\beta} = H^\top(K + \lambda I)^{-1}y, \qquad \boxed{\widehat{f} = K(K + \lambda I)^{-1}y.}$$

Both shifted matrices are invertible by positive definiteness, so all multiplications are legitimate even when H is rank deficient. The empirical SVD can be obtained from an eigendecomposition of K; this is a finite-sample alternative to the global Mercer construction.

Structural or analytic derivation

The push-through identity shows that the solution lies in the span of the observed feature vectors. The infinite or very large basis disappears from the prediction calculation once its inner products are known. The two meanings of eigen-decomposition—global kernel operator and empirical Gram matrix—are kept separate.

Comparative cost and preferred method

For $M \gg N$, a dense feature normal-equation solve has $O(M^3)$ factorization cost and $O(M^2)$ storage. The dual system has N unknowns and $O(N^3)$ factorization cost, plus kernel construction and prediction. With explicit H, constructing its Gram matrix costs $O(N^2M)$; this setup cost is included. A thin-SVD conventional implementation achieves the same dimensional reduction.

9.12 AC-141 | The large-penalty ridge direction

Reference: [9], exercise 18.1.

Source task and conventions

Identify the normalized limit and its one-component PLS interpretation.

Conventional solution and proof

Let $G = X^\top X$ and $c = X^\top y \neq 0$. Since $\beta_\lambda = (G + \lambda I)^{-1}c$, spectral diagonalization gives $\lambda\beta_\lambda \to c$ as $\lambda \to \infty$. Continuity of normalization away from zero therefore yields

$$\frac{\beta_\lambda}{\|\beta_\lambda\|} \longrightarrow \frac{c}{\|c\|}.$$

The first PLS score is $t = Xc$, and least squares on that single score has coefficient multiplier $a = t^\top y/(t^\top t) = c^\top c/(c^\top Gc) > 0$. Thus its feature coefficient is $\beta_{\mathrm{PLS},1} = ac$, with the same normalized direction. Because $c \in \operatorname{ran} X^\top$, $c^\top Gc > 0$ when $c \neq 0$. If $c = 0$, both coefficient vectors vanish and normalization is undefined; no directional limit is asserted.

Structural or analytic derivation

The resolvent expansion $(G + \lambda I)^{-1} = \lambda^{-1}[I - \lambda^{-1}G + O(\lambda^{-2})]$ identifies the first Krylov vector c as the leading term. The spectral proof supplies the limit without requiring a formal series outside its convergence range.

Comparative cost and preferred method

The limiting direction requires only the matrix-vector product $X^\top y$, costing $O(Np)$, instead of solving many large-λ systems. This computes a limit, not a finite-penalty ridge fit. An error bound follows from $\|\lambda\beta_\lambda - c\| \leq \|G\|\|c\|/\lambda$; normalization error additionally requires that the denominator stay away from zero.

9.13 AC-142 | High-dimensional ridge through the observed row space

Reference: [9], exercise 18.4.

Source task and conventions

Derive equation (18.15) with $X = RV^\top$ and $V^\top V = I$.

Conventional solution and proof

The normal equation is $(X^\top X + \lambda I)\widehat{\beta} = X^\top y$. Projecting onto the orthogonal complement of $\operatorname{ran} V$ gives $\lambda\widehat{\beta}_\perp = 0$, so for $\lambda > 0$, $\widehat{\beta} = V\theta$. Substitution gives $V(R^\top R + \lambda I)\theta = VR^\top y$, and multiplication by $V^\top$ yields

$$\boxed{\widehat{\beta} = V(R^\top R + \lambda I)^{-1} R^\top y.}$$

Equivalently, the original first-order equation implies $\widehat{\beta} = \lambda^{-1} X^\top (y - X\widehat{\beta})$, proving the source hint directly. The reduced shifted matrix is positive definite, so its solution is unique.

Structural or analytic derivation

The data rows occupy at most N dimensions of $\mathbb{R}^p$. Every orthogonal coefficient component changes only the penalty, so it is zero at the optimum. This is the finite-dimensional form of the representer argument.

Comparative cost and preferred method

A thin factorization costs $O(pN^2)$ classically when $p \gg N$, followed by an N-dimensional solve. This replaces the unnecessary $p \times p$ normal matrix and its $O(p^3)$ factorization. Constructing the final p-vector still costs $O(pN)$ and cannot be omitted when it is the requested output.

9.14 AC-143 | Exact reduction for arbitrary loss and quadratic regu-

Reference: [9], exercise 18.5.

Source task and conventions

Prove the equivalence of optimization problems (18.16)–(18.17), including existence and uniqueness qualifications.

Conventional solution and proof

Assume $X = RV^\top$, $V^\top V = I$, and $\lambda > 0$. Every β has a unique decomposition $\beta = V\theta + \eta$, $V^\top \eta = 0$. Then $X\beta = R\theta$ and $\|\beta\|^2 = \|\theta\|^2 + \|\eta\|^2$. The full objective is therefore exactly

$$\sum_i L(y_i, \beta_0 + r_i^\top \theta) + \lambda\|\theta\|^2 + \lambda\|\eta\|^2.$$

For every fixed (β_0, θ) it is uniquely minimized over η at zero. Thus its infimum equals the reduced problem's infimum, and its attained minimizers correspond bijectively to the reduced minimizers by $(\beta_0, \beta) = (\theta_0, V\theta)$. This proves the theorem as a statement about minimizer sets without assuming differentiability or convexity of L. If no minimum is attained, the equality of infima still holds but a hatted minimizer is not defined. If the reduced loss is not strictly convex in the unpenalized intercept, that intercept need not be unique. At $\lambda = 0$, orthogonal components may be arbitrary; the lifted reduced solution is a minimum-norm representative.

Structural or analytic derivation

Projection removes directions invisible to all observations. A quadratic norm is orthogonally invariant, which is precisely why this reduction is exact. An ℓ_1 penalty has no corresponding rotation invariance; substituting it invalidates the argument.

Comparative cost and preferred method

The parameter dimension drops from p to $r = \operatorname{rank} X \leq N$. The projection costs $O(pN^2)$ for a dense thin factorization; subsequent loss evaluations and gradients depend on r, not p. The theorem certifies equivalence for any supplied loss, but it does not make a nonconvex reduced loss globally easy to optimize.

9.15 AC-144 | Zero-residual fits and the minimum-norm interpolant

Reference: [9], exercise 18.7.

Source task and conventions

Complete parts (a)–(c) for full row-rank $X = UDV^\top = RV^\top$.

Conventional solution and proof

(a) Full row rank implies $R \in \mathbb{R}^{N\times N}$ is invertible. Every vector

$$\beta = VR^{-1}y + \eta, \qquad \eta \in \ker X,$$

has $X\beta = y$, and every zero-residual solution has this form by subtracting the displayed representative. Since $p > N$, the nullspace dimension is $p - N > 0$, giving infinitely many solutions. (b) Decompose $\beta = V\theta + \eta$. The ridge objective is $\|y - R\theta\|^2 + \lambda\|\theta\|^2 + \lambda\|\eta\|^2$. For $\lambda > 0$, $\eta = 0$ and differentiation gives $\theta = (R^\top R + \lambda I)^{-1}R^\top y$, the stated expression. (c) At zero penalty choose $\beta_0 = VR^{-1}y = VD^{-1}U^\top y$. It interpolates because $X\beta_0 = UDD^{-1}U^\top y = y$. Every other interpolant is $\beta_0 + \eta$ with $\eta \perp \beta_0$, so $\|\beta_0 + \eta\|^2 = \|\beta_0\|^2 + \|\eta\|^2$. This proves the claimed unique minimum-norm property.

Structural or analytic derivation

Interpolation of the observed response and uniqueness of the feature coefficients are separate claims. The pseudoinverse selects a canonical coefficient vector by orthogonal projection; it does not establish predictive validity away from the observed row space.

Comparative cost and preferred method

The thin SVD gives the full solution family and the minimum-norm representative in $O(pN^2)$ dense work. Enumerating nullspace coefficients is unnecessary when only the minimum-norm solution is required. The exact lower-dimensional solve is also the appropriate conventional method.

9.16 AC-145 | The ridged discriminant limit and maximal data piling

Reference: [9], exercise 18.10.

Source task and conventions

Adjudicate the literal limit and prove the correct directional result.

Conventional solution and proof

Let $d = \bar{x}_1 - \bar{x}_{-1}$, let P_0 be the orthogonal projector onto $\ker W$, and write $q = P_0 d$. Spectral decomposition gives

$$(W + \lambda I)^{-1} d = \lambda^{-1} q + W^{\dagger} d + O(\lambda).$$

Therefore the literal finite limit of the source's unscaled $\delta_\lambda(x)$ generally does not exist: $W = 0, d = x = 1$ gives $\delta_\lambda(x) = 1/\lambda$. The valid limits are

$$\lambda \delta_\lambda(x) \to x^\top q, \qquad \frac{(W + \lambda I)^{-1} d}{\|(W + \lambda I)^{-1} d\|} \to \frac{q}{\|q\|} \quad (q \neq 0).$$

To prove the data-piling claim, write W as a positive multiple of the sum of within-class centred outer products. Then $v \in \ker W$ exactly when $v^\top (x_i - \bar{x}_{y_i}) = 0$ for every training point. Thus each class projects to a single value, and the gap for unit $v \in \ker W$ is $|v^\top d| = |v^\top q| \leq \|q\|$. Equality holds at $v = q/\|q\|$, uniquely up to sign. For prescribed class responses ± 1 and a free intercept, coefficients must satisfy $v^\top d = 2$; Cauchy–Schwarz then gives the unique minimum-norm coefficient $2q/\|q\|^2$, with the intercept chosen to centre the two projected values. This is the maximal piling direction of Exercise 18.8 under its separating-data assumptions. If $q = 0$, no positive-gap zero-within-class-variance direction exists, and the asserted separating conclusion fails.

Structural or analytic derivation

The singular resolvent identifies the nullspace projection as the residue of the $1/\lambda$ pole. The residue, not an unscaled divergent score, is the geometrically meaningful limiting discriminant direction. The extremal gap follows from one Hilbert-space Cauchy–Schwarz inequality.

Comparative cost and preferred method

A rank-revealing factorization computes q directly, avoiding a sequence of ill-conditioned solves at decreasing λ. The proof gives the maximum separation among data-piling directions only; it does not equate that restricted problem with the general maximum-margin support-vector problem.

Source qualification

The source writes an unscaled limit. The divergence counterexample and the normalized/scaled replacement are explicit corrections.

9.17 AC-146 | One exact row-space reduction across cross-validation

Reference: [9], exercise 18.12.

Source task and conventions

Prove reuse of R for all ten-fold fits without changing the penalized model.

Conventional solution and proof

Compute $X = RV^\top$ once with V an orthonormal basis containing the span of every data row. For a training-index subset T, $X_T = R_T V^\top$. Every coefficient decomposes as $V\theta + \eta$ with $V^\top \eta = 0$, so the fold objective equals

$$\sum_{i \in T} L(y_i, \beta_0 + r_i^\top \theta) + \lambda \|\theta\|^2 + \lambda \|\eta\|^2.$$

For $\lambda > 0$, the minimizer has $\eta = 0$ and is exactly the fit on R_T. Held-out predictions also agree because $X_i V \theta = r_i^\top \theta$. This argument applies independently to every fold and every positive λ, proving reuse. The reduction is an exact orthogonal coordinate change on the complete row span, not a rank truncation. It preserves even each fold's directions that are unobserved in its training rows, since positive regularization makes their fitted coefficient zero.

Structural or analytic derivation

Using held-out covariates to define this full isometry does not alter any fitted function or held-out prediction in the declared model. A data-dependent feature selection, truncated PCA, bandwidth choice or nonlinear preprocessing step is a different operation and must be fitted according to the validation protocol. An unpenalized intercept is retained unchanged in the proof.

Comparative cost and preferred method

With F folds and L penalty values, one thin factorization costs $O(pN^2)$ once, instead of repeating that feature-space reduction FL times. Reduced fits still have their own solver cost. A strong dual-Gram implementation can obtain the same reuse without explicitly storing V when only kernel predictions are required.

9.18 AC-147 | Kernel logistic regression from an empirical square-root

Reference: [9], exercise 18.13.

Source task and conventions

Prove both coefficient and prediction formulas, including a singular kernel.

Conventional solution and proof

Let $K = UD^2U^\top$ on its positive eigenspace, with $U \in \mathbb{R}^{N\times r}$, $D \succ 0$, and $R = UD$. Let $X = UDV^\top$ be a compatible thin SVD. The quadratic-penalty projection theorem makes the feature coefficient $w = V\beta$, where β minimizes

$$\sum_i \{\log(1 + e^{r_i^\top \beta}) - y_i r_i^\top \beta\} + \lambda\|\beta\|^2$$

for the 0/1 logistic convention. A free intercept can be appended consistently. The new inner-product vector is $k_0 = Xx_0 = UDV^\top x_0$. Consequently

$$\alpha = UD^{-1}\beta, \qquad k_0^\top \alpha = x_0^\top VDU^\top UD^{-1}\beta = x_0^\top V\beta = x_0^\top w.$$

This proves (a)–(b) and the new-point linear predictor. Applying the sigmoid gives the predicted probability. For a general PSD kernel, R is an empirical feature matrix and the same optimization defines kernel logistic regression. For singular K, retain only positive eigenvalues or use the pseudoinverse on that support. A legitimate extended kernel has $k_0 \in \operatorname{ran} K$: if $a \in \ker K$, positivity of the block Gram matrix on $(ta, 1)$ for every real t forces $a^\top k_0 = 0$. Thus the omitted zero eigenvalues cause no prediction ambiguity.

Structural or analytic derivation

The nonlinear part of the problem remains the logistic loss; the kernel transformation is linear algebra in a Hilbert-space feature span. Positive regularization makes the slope objective strictly convex, while an unpenalized intercept may need both classes present for a finite optimum.

Comparative cost and preferred method

The Newton or first-order optimization uses at most $r \le N$ coefficients, replacing a p-parameter system. Eigensystem construction costs $O(N^3)$ densely; each Newton Hessian depends on iteration-specific logistic weights. This does not supply the exact one-factorization leave-one-out identity available for squared loss.

9.19 AC-148 | Leave-one-out local polynomial regression

Reference: [9], exercise 6.7.

Source task and conventions

Derive the complete local-polynomial cross-validated residual expression.

Conventional solution and proof

At target x_i, fix the bandwidth, degree and basis $b_{ij} = (1, x_j - x_i, \dots, (x_j - x_i)^d)^\top$, with nonnegative weights w_{ij}. Put $M_i = \sum_j w_{ij} b_{ij} b_{ij}^\top$ and $a_i = \sum_j w_{ij} b_{ij} y_j$. Assume the full and deleted local design matrices are invertible. The fitted value is $\widehat{f}_i = e_1^\top M_i^{-1} a_i$. Since $b_{ii} = e_1$, its self-weight is $S_{ii} = w_{ii} e_1^\top M_i^{-1} e_1$. Deletion gives $M_i^{(-i)} = M_i - w_{ii} e_1 e_1^\top$ and $a_i^{(-i)} = a_i - w_{ii} e_1 y_i$. The rank-one inverse formula follows by multiplying

$$(M - uu^\top)^{-1} = M^{-1} + \frac{M^{-1} u u^\top M^{-1}}{1 - u^\top M^{-1} u}$$

by $M - uu^\top$. Substituting $u = \sqrt{w_{ii}} e_1$ and simplifying gives

$$\boxed{y_i - \widehat{f}_i^{(-i)} = \frac{y_i - \widehat{f}_i}{1 - S_{ii}}, \qquad \mathrm{CVRSS} = \sum_i \left(\frac{y_i - \widehat{f}_i}{1 - S_{ii}} \right)^2 .}$$

Divide by N for the mean error convention. The denominator is positive under the stated positive-definite deleted-design assumption. If $w_{ii} = 0$, the observation never enters its own fit and the formula reduces correctly to the ordinary residual.

Structural or analytic derivation

The local smoother may not be a symmetric global matrix, but each target's deletion is still a rank-one change to its local normal equations. The fixed bandwidth and basis are part of the theorem: adaptive bandwidth reselection after deletion requires a separate computation.

Comparative cost and preferred method

An already computed local fit and its inverse self-leverage yield the deleted residual with $O(1)$ additional scalar work per target. Computing all local fits is not eliminated: naive dense local design construction costs $O(N^2 d^2)$ and N small factorizations cost $O(Nd^3)$. Repeating a complete smoother fit for each deletion would unnecessarily multiply the target work; a strong conventional implementation fitting only the held-out target has a smaller baseline.

9.20 AC-149 | When the PRESS identity holds for a linear smoother

Reference: [9], exercise 7.3.

Source task and conventions

Prove (a)–(c), including denominator conditions and the exact deletion-consistency criterion.

Conventional solution and proof

(a) For a fixed penalized least-squares rule, augment a deleted-data minimizer by observation $(x_i, \widehat{f}^{(-i)}(x_i))$. The added squared residual is zero at that minimizer and nonnegative elsewhere, so it remains a minimizer. Under uniqueness, fitted values are identical. Because the full rule is linear, replacing y_i by $t = \widehat{f}^{(-i)}(x_i)$ gives $t = \widehat{f}(x_i) + S_{ii}(t - y_i)$. If $1 - S_{ii} \neq 0$, rearrangement yields the claimed identity. Least-squares projections and fixed-parameter cubic smoothing splines satisfy this self-consistency property, provided the deleted fit is well defined and the unnormalised penalty remains fixed.

(b) For a projection, $S = S^\top = S^2$, so $0 \leq S_{ii} \leq 1$. For a smoothing spline, the fitted-value problem can be represented by minimizing $\|y - f\|^2 + \lambda f^\top \Omega f$ over the feasible evaluation space, with $\Omega \succeq 0$; its smoother eigenvalues lie in $[0, 1]$, including unpenalized components. Thus $0 \leq S_{ii} < 1$ whenever the nondegenerate deletion identity applies, and

$$|y_i - \widehat{f}^{(-i)}(x_i)| = |y_i - \widehat{f}(x_i)|/(1 - S_{ii}) \geq |y_i - \widehat{f}(x_i)|.$$

At $S_{ii} = 1$, the fraction is undefined and the conclusion needs a separate interpolation/deletion analysis.

(c) Let the deleted linear prediction be $\widehat{f}^{(-i)}(x_i) = \sum_{j \neq i} a_{ij} y_j$, with both rules independent of response values. For $S_{ii} \neq 1$, the identity holds for every y if and only if

$$\boxed{a_{ij} = S_{ij}/(1 - S_{ii}) \quad (j \neq i).}$$

Necessity follows by comparing coefficients of each y_j; sufficiency follows by substitution. Equivalently the rule reproduces the deleted prediction after replacing the omitted response by that prediction. Linearity alone is insufficient: the full and deleted smoother families must have this relationship. Response-dependent parameter tuning generally destroys it.

Structural or analytic derivation

The rank-one normal-equation argument and the self-consistency argument are independent routes to the same deletion formula. The latter also covers spline penalties with an unpenalized nullspace when the fit remains unique.

Comparative cost and preferred method

For a fixed global smoother, one full fit plus its diagonal provides all leave-one-out residuals; the extra work after those quantities are known is $O(N)$. A diagonal is not free when the algorithm only supplies a matrix-vector product, and parameter selection inside each fold remains separate. The exact algebra establishes a computational reduction for compatible smoothers, not for every linear predictor family.

Chapter 10

An Introduction to Statistical Learning: with Applications in Python

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor. Reference problems and derived analyses.

10.1 AC-150 | Polynomial regression, exact PRESS and the Python-seeded sample

Reference: [10], exercise 8.

Source task and conventions

Answer all six simulation, plotting, cross-validation and significance subparts.

Conventional solution and complete derivation

The source's `default_rng(1)` draws 100 independent standard-normal predictors and then 100 independent errors. Thus (a) $n = 100$, with one original predictor $p = 1$, and the model is $Y_i = X_i - 2X_i^2 + \epsilon_i$, X_i, ϵ_i independent $N(0,1)$. The original predictor count remains one, although the degree-d design has d nonconstant polynomial columns. For (b), the scatterplot shows a noisy concave parabola, whose conditional mean has vertex at $x = 1/4$ and value $1/8$; this follows by completing the square $x - 2x^2 = 1/8 - 2(x - 1/4)^2$. Points are not expected to lie exactly on that curve because the errors have variance one.

For any fixed degree d, let $A_{ij} = x_i^j$, $j = 0, \dots, d$, including the intercept, and $H = A(A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}$. Denote the full residual by $e = y - A\widehat{\beta}$ and row vector by a_i^{T}. Deleting observation i changes the normal matrix to $M - a_i a_i^{\mathsf{T}}$ and its right side to $A^{\mathsf{T}}y - a_i y_i$. If $1 - h_{ii} > 0$, the Sherman–Morrison identity (verified by multiplying its two sides by $M - a_i a_i^{\mathsf{T}}$) gives

$$y_i - a_i^{\mathsf{T}}\widehat{\beta}_{-i} = \frac{e_i}{1 - h_{ii}}.$$

The condition holds for the reported designs; distinct sample abscissas give full column rank even after deleting one point. Thus the exact leave-one-out mean square is $100^{-1}\sum_i [e_i/(1 - h_{ii})]^2$. A QR factorization computes the same diagonal by $h_{ii} = \sum_j Q_{ij}^2$, avoiding explicit normal-matrix inversion. The supplied script also fits all 100 deleted models separately for each degree and checks their residuals against this formula. For (c), the degree-one through degree-four leave-one-out MSE values are, in order, $(6.6330298392, 1.1229368563, 1.3017965489, 1.3323942694)$. All coefficients,

standard errors, two-sided t p-values and leverages are printed in the empirical table. For (d), changing a seed used only by the validation routine does not change leave-one-out partitions: every row is excluded once, deterministically. Consequently the four scores are unchanged for the same stored sample. Regenerating the data with another seed would change the data and generally the scores; that is a different experiment. For (e), degree two has the smallest score, consistent with the quadratic conditional mean. Adding higher terms increases variance without changing the true conditional mean, although any single realized sample can show a different ranking. For (f), conditional on the predictor matrix, the classical Gaussian linear-model variance estimate is $\widehat{\sigma}^2 = \|e\|^2/(100 - d - 1)$ and coefficient covariance is $\widehat{\sigma}^2(A^\mathsf{T}A)^{-1}$. Dividing each coefficient by its standard error and using the $t_{100-d-1}$ distribution gives the reported nominal two-sided significance tests. These tests assess individual conditional coefficient hypotheses; leave-one-out assesses held-out squared prediction error. They need not agree for every extra term. In particular the R-seeded sample can give a nominally significant cubic coefficient while the quadratic model still has smaller held-out error. Polynomial collinearity and multiple testing further prevent interpreting one p-value as a model-selection proof. The table supplies the exact numerical evidence for each source edition instead of conflating their different random samples.

Structural or analytic formulation

The residual identity extracts every deleted least-squares fit from one QR factorization. The proof shows precisely why it is exact for this linear-in-parameters squared-loss problem. The logistic question has a different nonlinear score and must not borrow this exact identity. The plotted polynomial curve and residuals are computed from the same stored sample as the table.

Comparative efficiency and scope

At fixed degree with $k = d + 1$ columns, 100 separate fits require $O(n^2k^2 + nk^3)$ arithmetic in a direct implementation. A single QR fit, leverage calculation and residual division need $O(nk^2+k^3)$ and $O(nk)$ storage. The source output is one error per model, so the full n deleted coefficient vectors need not be formed. The numerical residual agreement is checked independently; the asymptotic reduction follows from the exact formula, not timing alone.

Coefficient estimates and significance

Degree	Power	Estimate	Std. error	t-test p
1	0	-1.46496301	0.246752605	4.4096e-08
1	1	1.94936857	0.288689866	1.0365e-09
2	0	-0.0727552908	0.119133593	0.54282
2	1	0.966272759	0.126353175	1.4949e-11
2	2	-2.00470902	0.0908258146	1.3401e-39
3	0	-0.0571966912	0.119928804	0.6345
3	1	1.1145842	0.187471569	4.4485e-08
3	2	-2.04709357	0.0990222967	3.9689e-37
3	3	-0.0643003271	0.0600811737	0.2872
4	0	0.100847655	0.135818386	0.45961
4	1	0.904997864	0.204613555	2.5913e-05
4	2	-2.50592308	0.221057793	2.396e-19
4	3	0.0337683719	0.0725098661	0.64249
4	4	0.104216994	0.0451336459	0.023107

Degree	LOO MSE	Max PRESS/refit discrepancy
1	6.633029839	1.97e-15
2	1.122936856	1.29e-14
3	1.301796549	2.93e-14
4	1.332394269	3.73e-14

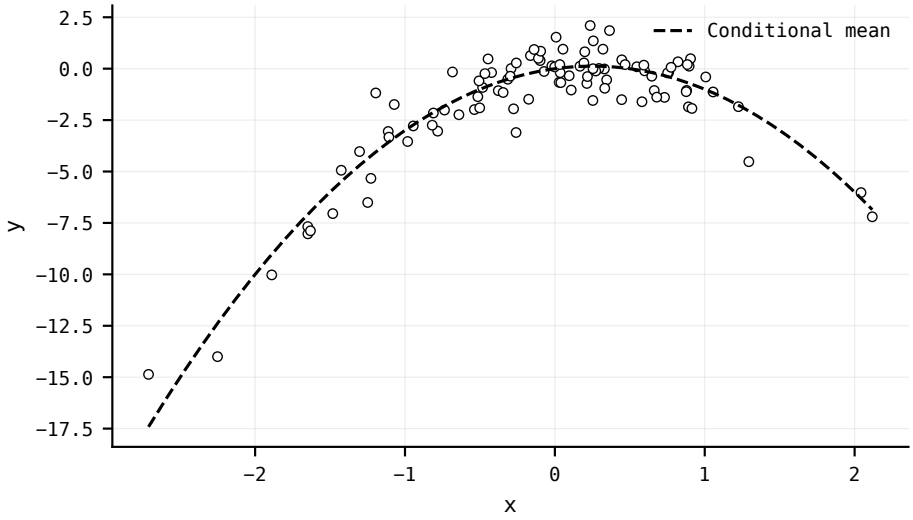


Figure 10.1: The exact stored predictor and response sample for Python_default_rng_1. The dashed curve is the specified conditional mean.

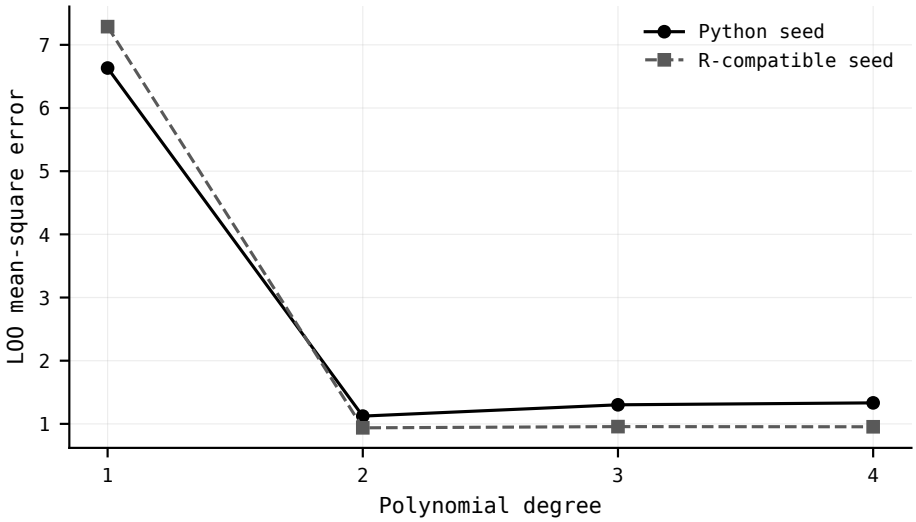


Figure 10.2: Both source-specific samples select degree two among the four candidates; identical response data are used for explicit refits and PRESS.

10.2 AC-151 | Complete Weekly logistic leave-one-out calculation: Python edition

Reference: [10], exercise 7.

Source task and conventions

Fit the full and first-row-deleted models, classify the first observation, and execute every leave-one-out fit.

Conventional solution and complete derivation

Write $x_i = (1, \mathrm{Lag1}_i, \mathrm{Lag2}_i)^\mathsf{T}$ and $y_i = 1$ for Up, zero for Down. The model probability is $p_i(\theta) = 1/(1 + e^{-x_i^\mathsf{T}\theta})$, and its negative log likelihood is

$$L(\theta) = \sum_i \{\log(1 + e^{x_i^\mathsf{T}\theta}) - y_i x_i^\mathsf{T}\theta\}.$$

Differentiation gives $\nabla L = X^\mathsf{T}(p - y)$ and $\nabla^2 L = X^\mathsf{T}\,\mathrm{diag}(p_i(1 - p_i))X$. For a finite parameter vector all weights are positive; full column rank makes the Hessian positive definite, because $u^\mathsf{T}\nabla^2 L u = \sum_i p_i(1-p_i)(x_i^\mathsf{T}u)^2 > 0$ for $u \neq 0$. Hence a finite zero of the score is the unique global minimizer. The executed Newton solver uses stable log-likelihood evaluation and a line search, and reports score and Hessian diagnostics for every fit. The complete source data contain 1,089 observations. The retrieved compressed R object was decoded only after its Git blob SHA-1 matched the source value; labels, row counts, lag alignment and first-record values were checked. The full-data coefficient vector (intercept, Lag1, Lag2) is $(0.2212240543, -0.03872222247, 0.06024830276)$. After omitting the first row it is $(0.2232430465, -0.03843316878, 0.06084762641)$. The omitted point has predicted Up probability $0.5713923205 > 1/2$, so it is classified Up; its actual label is Down. This answers (a)–(c). For (d), loop over $i = 1, \ldots, 1089$, fit the same unpenalized model to all rows except i, predict that excluded row using the 0.5 threshold, and store $e_i = \mathbf{1}\{\widehat{y}_{-i} \neq y_i\}$. No excluded response is used by its fitted model. The supplied program prints the first error and writes all 1,089 held-out probabilities, labels and errors. Their sum is 490, so (e)

$$\boxed{\widehat{R}_{\mathrm{LOO}} = 490/1089 = 0.4499540863\ldots.}$$

This is the requested empirical classification-error estimate, not a confidence interval or a prospective forecast guarantee for dependent time-series observations. It uses leave-one-out sampling as specified; chronological validation would answer a different deployment question.

Structural or analytic formulation

A full-data fit is an effective initialization for each deleted fit. It does not make logistic leave-one-out an exact PRESS identity: deleting one observation changes the optimum-dependent weights throughout the nonlinear Hessian. The recorded experiment performed every deleted nonlinear optimization to its convergence criterion. Cold starts and full-fit warm starts produce held-out probabilities differing by at most 1.76×10^{-11} and exactly the same 490 error labels. The smallest observed distance from threshold 0.5 is 3.4376×10^{-4}, far larger than that numerical discrepancy. These are floating-point diagnostics, not interval enclosures.

Comparative efficiency and scope

Cold starts used 4,398 Newton iterations across the deleted fits; warm starts used 3,288, saving 1,110 iterations (25.24%) on this dataset. Single-run timings were approximately 0.499 s and 0.386 s in the recorded environment; they are not presented as a repeated timing experiment. Each iteration computes a three-column weighted Hessian in $O(m)$ work, so this is a measured iteration-count reduction with matched outputs. Both algorithms remain $O(m^2)$ for all m deleted fits at bounded iteration count; no one-factorization exact logistic shortcut is asserted.

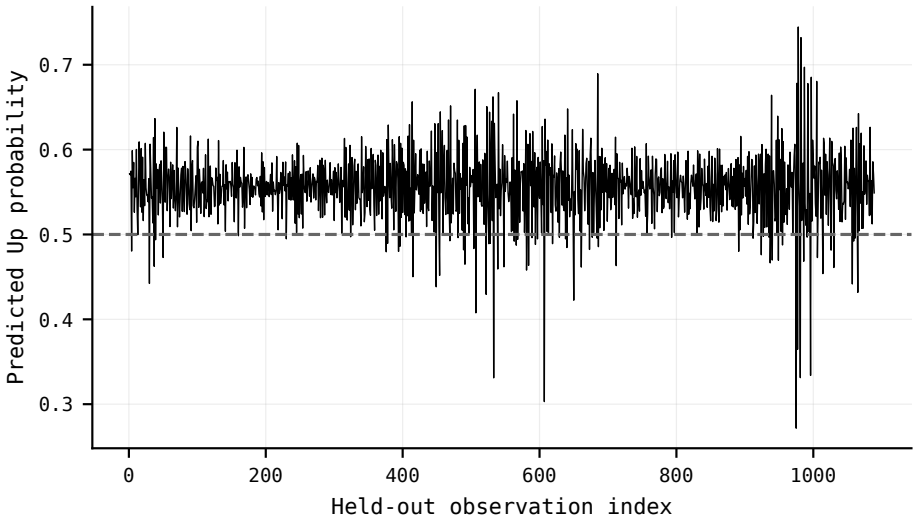


Figure 10.3: All 1,089 held-out probabilities, each from a model trained without its row. Exact prediction/error arrays are supplied.

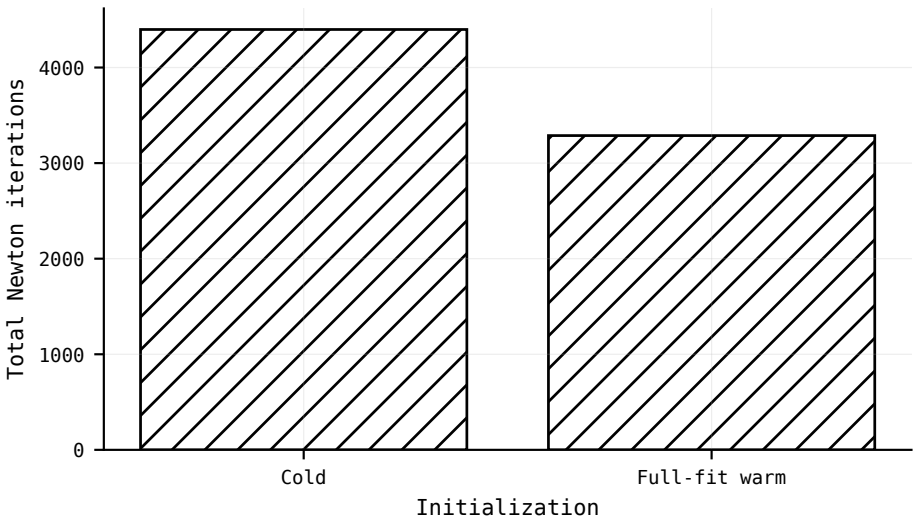


Figure 10.4: All deleted nonlinear fits converge under both starts; the warm starts save 1,110 iterations while preserving all 490 error labels.

Chapter 11

An Introduction to Statistical Learning: with Applications in R

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. Reference problems and derived analyses.

11.1 AC-152 | Polynomial regression, exact PRESS and the R-seeded sample

Reference: [11], exercise 8.

Source task and conventions

Answer all six simulation, plotting, cross-validation and significance subparts.

Conventional solution and complete derivation

The source's `set.seed(1); rnorm(100)` stream is reconstructed using R's Mersenne–Twister initialization and inversion-normal convention. The first five values are checked against the documented seed-one sequence; this is a compatible reconstruction, not a native R execution. The exact generator implementation and the resulting data arrays are included. Thus (a) $n = 100$, with one original predictor $p = 1$, and the model is $Y_i = X_i - 2X_i^2 + \epsilon_i$, X_i, ϵ_i independent $N(0, 1)$. The original predictor count remains one, although the degree-d design has d nonconstant polynomial columns. For (b), the scatterplot shows a noisy concave parabola, whose conditional mean has vertex at $x = 1/4$ and value $1/8$; this follows by completing the square $x - 2x^2 = 1/8 - 2(x - 1/4)^2$. Points are not expected to lie exactly on that curve because the errors have variance one.

For any fixed degree d, let $A_{ij} = x_i^j$, $j = 0, \dots, d$, including the intercept, and $H = A(A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}}$. Denote the full residual by $e = y - A\widehat{\beta}$ and row vector by a_i^{T}. Deleting observation i changes the normal matrix to $M - a_i a_i^{\mathsf{T}}$ and its right side to $A^{\mathsf{T}}y - a_i y_i$. If $1 - h_{ii} > 0$, the Sherman–Morrison identity (verified by multiplying its two sides by $M - a_i a_i^{\mathsf{T}}$) gives

$$y_i - a_i^{\mathsf{T}}\widehat{\beta}_{-i} = \frac{e_i}{1 - h_{ii}}.$$

The condition holds for the reported designs; distinct sample abscissas give full column rank even after deleting one point. Thus the exact leave-one-out mean square is $100^{-1}\sum_i [e_i/(1 - h_{ii})]^2$. A QR factorization computes the same diagonal by $h_{ii} = \sum_j Q_{ij}^2$, avoiding explicit normal-matrix inversion. The supplied script also fits all 100 deleted models separately for each degree and checks

their residuals against this formula. For (c), the degree-one through degree-four leave-one-out MSE values are, in order, $(7.2881616067, 0.9374236376, 0.9566218301, 0.9539048927)$. All coefficients, standard errors, two-sided t p-values and leverages are printed in the empirical table. For (d), changing a seed used only by the validation routine does not change leave-one-out partitions: every row is excluded once, deterministically. Consequently the four scores are unchanged for the same stored sample. Regenerating the data with another seed would change the data and generally the scores; that is a different experiment. For (e), degree two has the smallest score, consistent with the quadratic conditional mean. Adding higher terms increases variance without changing the true conditional mean, although any single realized sample can show a different ranking. For (f), conditional on the predictor matrix, the classical Gaussian linear-model variance estimate is $\widehat{\sigma}^2 = \|e\|^2/(100 - d - 1)$ and coefficient covariance is $\widehat{\sigma}^2(A^\mathsf{T}A)^{-1}$. Dividing each coefficient by its standard error and using the $t_{100-d-1}$ distribution gives the reported nominal two-sided significance tests. These tests assess individual conditional coefficient hypotheses; leave-one-out assesses held-out squared prediction error. They need not agree for every extra term. In particular the R-seeded sample can give a nominally significant cubic coefficient while the quadratic model still has smaller held-out error. Polynomial collinearity and multiple testing further prevent interpreting one p-value as a model-selection proof. The table supplies the exact numerical evidence for each source edition instead of conflating their different random samples.

Structural or analytic formulation

The exact residual identity and its QR-based implementation are established in AC-150. They apply unchanged to this different realized predictor/error sample. The numerical data, fitted coefficients and significance results below are specific to the R-compatible stream; the identity does not identify the two samples or their error values.

Comparative efficiency and scope

The shared arithmetic and storage comparison is proved in AC-150. The present instance changes the observations, not the number of design-matrix columns or the output contract. Its independent refit comparison tests the same exact identity on a second random-number stream.

Coefficient estimates and significance

Degree	Power	Estimate	Std. error	t-test p
1	0	-1.62542674	0.261936634	1.3093e-08
1	1	0.692497014	0.290941778	0.019238
2	0	0.0567150089	0.117655494	0.63086
2	1	1.01716087	0.107982694	2.4033e-15
2	2	-2.1189212	0.084765703	4.5843e-44
3	0	0.0615071798	0.11950374	0.60795
3	1	0.97528027	0.18728149	1.0893e-06
3	2	-2.12379099	0.0870025131	5.8734e-43
3	3	0.0176385754	0.0642903659	0.7844
4	0	0.156702953	0.139461925	0.264
4	1	1.03082564	0.191336549	5.1743e-07
4	2	-2.40989818	0.234855057	4.5752e-17
4	3	-0.00913290388	0.0672288073	0.89223
4	4	0.0697854209	0.0532400556	0.1931

Degree	LOO MSE	Max PRESS/refit discrepancy
1	7.288161607	1.78e-15
2	0.9374236376	6.88e-15
3	0.9566218301	1.02e-14
4	0.9539048927	2.55e-14

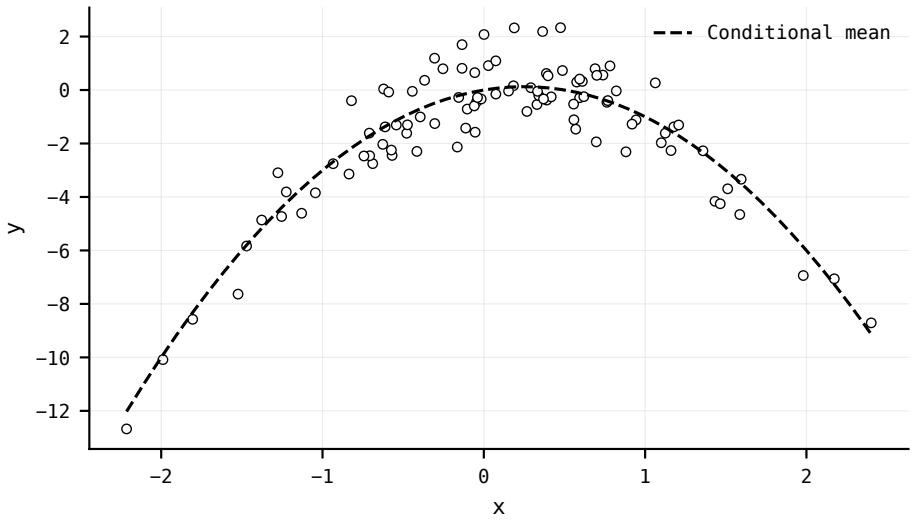


Figure 11.1: The exact stored predictor and response sample for R_seed1_reconstructed. The dashed curve is the specified conditional mean.

Graphical reference. See Figure 10.2 for the shared construction.

11.2 AC-153 | Complete Weekly logistic leave-one-out calculation: R edition

Reference: [11], exercise 7.

Source task and conventions

Fit the full and first-row-deleted models, classify the first observation, and execute every leave-one-out fit.

Conventional solution and complete derivation

The likelihood, score, Hessian, full-data estimate and first deleted-observation calculation are given in AC-151. This edition uses the same observations and the same intercept-plus-two-lags model. Consequently its mathematical solution and its 490 leave-one-out errors are shared with that case. The language-specific obligation is to reproduce the data interpretation, fit criterion, held-out prediction and threshold convention in the supplied R program. A native R execution has not been recorded.

Structural or analytic formulation

A full-data fit is an effective initialization for each deleted fit. It does not make logistic leave-one-out an exact PRESS identity: deleting one observation changes the optimum-dependent

weights throughout the nonlinear Hessian. The recorded experiment performed every deleted nonlinear optimization to its convergence criterion. Cold starts and full-fit warm starts produce held-out probabilities differing by at most 1.76×10^{-11} and exactly the same 490 error labels. The smallest observed distance from threshold 0.5 is 3.4376×10^{-4}, far larger than that numerical discrepancy. These are floating-point diagnostics, not interval enclosures. The accompanying R script uses `glm(Direction~Lag1+Lag2,family=binomial)` with explicit factor levels Down, Up and `type="response"`. The numerical results above were independently executed from the identical R-package data in Python; a native R interpreter was not run. The R code is complete and the model, objective and held-out sets are identical.

Comparative efficiency and scope

The 1,089-fit warm-start and cold-start comparison is the single experiment reported in AC-151; it is not a second independent performance sample. Reusing its mathematical and numerical evidence avoids counting the same work twice. A separate R runtime comparison would require recorded native execution under an equivalent stopping rule.

Graphical reference. See Figure 10.3 for the shared construction.

Graphical reference. See Figure 10.4 for the shared construction.

Chapter 12

Introduction to Natural Language Processing

Jacob Eisenstein. Reference problems and derived analyses.

12.1 AC-154 | An exact two-ReLU XOR network

Reference: [12], exercise 3.4.

Source task and conventions

Specify all hidden and output weights and offsets for the source sign convention.

Conventional solution and proof

For $x_1, x_2 \in \{0, 1\}$, take

$$z_1 = \mathrm{ReLU}(x_1 - x_2), \qquad z_2 = \mathrm{ReLU}(x_2 - x_1), \qquad y = \mathrm{Sign}(z_1 + z_2 - 1/2).$$

The hidden weight rows are $(1, -1)$ and $(-1, 1)$, both hidden biases are zero, the output weights are $(1, 1)$, and the output bias is $-1/2$. For inputs $(0, 0), (1, 0), (0, 1), (1, 1)$ the hidden pairs are respectively $(0, 0), (1, 0), (0, 1), (0, 0)$; the output preactivations are $-1/2, 1/2, 1/2, -1/2$. Applying the stated sign convention gives exactly $-1, 1, 1, -1$, with no boundary ambiguity. Two hidden units are also minimal for a network of the specified form without skip connections. With one ReLU unit, the sign of $a\,\mathrm{ReLU}(w^\top x + b) + c$ defines a halfspace, its complement, or a constant set along the scalar $w^\top x + b$. XOR is not linearly separable: if an affine function is positive at $(1, 0), (0, 1)$ and nonpositive at $(0, 0), (1, 1)$, adding the first two inequalities contradicts the latter two. Therefore one hidden unit cannot realize this truth table.

Structural or analytic derivation

The hidden sum equals $|x_1 - x_2|$, so the network tests whether the two Boolean inputs differ. On the Boolean domain the same function is the polynomial $2x_1 + 2x_2 - 4x_1x_2 - 1$. That polynomial is a verification identity for the four inputs; it is not a substitution for the required ReLU architecture off the Boolean domain.

Comparative cost and preferred method

The construction uses two hidden ReLUs and one sign node, and the lower-bound argument proves minimal hidden width in the stated architecture. A four-row truth-table check is exact. Complex differentiability is irrelevant: ReLU and sign are not holomorphic functions and a complex-step gradient argument would not verify this network.

12.2 AC-155 | A complete certificate for a dead ReLU on the Boolean

Reference: [12], exercise 3.8.

Source task and conventions

Answer all three parts, including the derivative convention at zero and the optimizer assumptions.

Conventional solution and proof

Write $a_i(x) = b_i + \sum_{j=1}^{D} w_{ij}x_j$ and $z_i = \max(0, a_i)$. (a) A linear function on $\{0,1\}^D$ is maximized by setting $x_j = 1$ for $w_{ij} > 0$ and zero for $w_{ij} < 0$. Hence

$$\boxed{z_i \text{ is zero for every input } \iff b_i + \sum_j \max(w_{ij}, 0) \le 0.}$$

The same maximum holds on the full cube $[0,1]^D$.

(b) Let the outgoing weight be $v_i = \theta_i^{(z \to y)}$ and $\partial L/\partial y = 1$. At $a_i \neq 0$, the chain rule gives

$$\frac{\partial L}{\partial b_i} = v_i \mathbf{1}\{a_i > 0\}, \qquad \frac{\partial L}{\partial w_{ij}} = v_i x_j \mathbf{1}\{a_i > 0\}, \qquad \frac{\partial L}{\partial v_i} = z_i.$$

At $a_i = 0$, ReLU has no classical derivative. A chosen subgradient $\eta \in [0,1]$ replaces the indicator; the common convention is $\eta = 0$.

(c) Under that zero-at-zero convention, a dead unit has zero incoming-weight, bias and outgoing-weight data gradients on every permitted input. Plain gradient descent or stochastic gradient descent without an additional update term therefore leaves all these parameters unchanged. Induction over optimization steps proves it remains dead. With strictly negative maximum preactivation, all input points are differentiable inactive points. If equality occurs in (a) and a nonzero subgradient is chosen at zero, the permanence statement is not automatic. Stored momentum, injected noise or other parameter updates also fall outside the pure zero-gradient argument.

Structural or analytic derivation

The cube support function $h_{[0,1]^D}(w) = \sum_j (w_j)_+$ replaces enumeration of every possible input. It gives a global activation certificate and identifies precisely the boundary at which derivative conventions matter.

Comparative cost and preferred method

The certificate takes $O(D)$ arithmetic and one pass over the weights, compared with evaluating 2^D inputs at $O(D)$ each. The pass is worst-case necessary in an unrestricted coefficient-access model: an uninspected coefficient can be changed to a sufficiently large positive value and alter the answer. This is a genuine exact complexity reduction; it proves activation behaviour, not a universal assertion about every optimizer called gradient-based.

Chapter 13

Certified roots, continuation and partial computation

13.1 CSE-A: roots as typed algebraic data

13.1.1 A root is not only a complex number

For the equation $z^2 = w$, a robust branch-sensitive object has at least the form

$$(p(z, w),\ \Omega_w,\ w_0,\ Z_0,\ \gamma,\ \mathcal{C}),$$

where p is the equation, Ω_w excludes the discriminant, Z_0 is an isolating neighbourhood of the initial root, γ is the continuation path and $\mathcal{C}$ contains local continuation certificates. A rounded complex number omits all but an approximate state. A principal-root routine instead implements a particular slit-domain convention; it cannot promise continuity across its own branch cut.

Theorem 13.1 (C01: A residual needs separation to become a root error)**.** *If r is a root of $z^2 - w$ and $\widehat{z}$ is a candidate, then*

$$|\widehat{z} - r| = \frac{|\widehat{z}^2 - w|}{|\widehat{z} + r|}$$

when the denominator is nonzero. If a certificate proves $|\widehat{z} + r| \geq s > 0$, then the residual divided by s bounds the forward error.

Proof. Factor $\widehat{z}^2 - r^2 = (\widehat{z} - r)(\widehat{z} + r)$ and take absolute values. The assumed separation permits division. Without it, a small residual gives no comparable linear bound. □

Near $w = 0$, the two roots approach each other and the inverse derivative diverges. At $w = \varepsilon^2$, choosing $\widehat{z} = 0$ gives residual ε^2 but root distance ε. Calling the residual itself a root error is false. Conversely choosing the other exact root gives zero residual while being wrong for a branch-sensitive request.

13.1.2 Quotient-algebra representation and repeated fibres

For $p(z, w) = z^2 - w$, represent the algebra $\mathbb{C}[z]/(p)$ in the basis $(1, z)$, with multiplication matrix

$$M(w) = \begin{pmatrix} 0 & w \\ 1 & 0 \end{pmatrix}.$$

The relation $M^2 = wI$ is exact. At $w = 0$, M is nonzero nilpotent. Replacing the fibre by the set $\{0\}$ would lose this multiplicity information. The characteristic polynomial is still $\lambda^2 - w$, and its two-dimensional algebra does not collapse when its reduced support has one point.

For a monic degree-d polynomial, multiplication by z on its quotient algebra gives a $d \times d$ companion representation. Reduction modulo the polynomial converts higher powers to length-d coefficient vectors. This can be preferable to storing all roots when the requested output is a trace, norm or polynomial remainder. It does not imply that the companion matrix is well conditioned in every basis.

Worked problem: A branch-free exponential observable

Since $M^2 = wI$,

$$e^{tM} = a_t(w)I + b_t(w)M, \quad a_t(w) = \sum_{n\geq 0} \frac{t^{2n}w^n}{(2n)!}, \quad b_t(w) = \sum_{n\geq 0} \frac{t^{2n+1}w^n}{(2n+1)!}.$$

Both coefficients are entire in w. Writing them as $\cosh(t\sqrt{w})$ and $\sinh(t\sqrt{w})/\sqrt{w}$ is convenient away from zero but can conceal the removable singularity of the second expression. At zero, $e^{tM} = I + tM$. The finite algebra supplies an analytic representation without eigenvalue sorting. This develops the matrix-monodromy investigation XM06 instead of relabelling a discontinuous eigenvector plot as a physical singularity.

13.1.3 A specification for implementation

The input polynomial must be normalized before degree is read. Coefficients can be exact rationals, intervals, balls, or floating-point values with a separate coefficient-error model; those are different input types. The returned certificate must bind to the exact input coefficients, not merely to the same filename. A result for a rounded polynomial is not automatically a result for the original real-valued model.

For repeated roots, use multiplicity-aware algebra or a cluster representation. A distinct-root API should reject a discriminant-zero input instead of manufacture separated roots at an arbitrary numerical tolerance. For symbolic expressions, simplification must preserve the declared domain: cancelling a denominator can extend a function across a removable hole, which is a different partial-function contract unless the extension is explicit.

13.2 CSE-B: an exact root-count checker for rational contours

13.2.1 From an argument-principle theorem to finite data

Let p have coefficients in $\mathbb{Q}(i)$ and let Γ be a positively oriented rational polygon avoiding its zeros. On one segment $z(t) = a + (b - a)t$, $0 \leq t \leq 1$, the image $q(t) = p(z(t))$ is a polynomial in one real variable. In its degree-m Bernstein form,

$$q(t) = \sum_{k=0}^{m} b_k \binom{m}{k} t^k (1-t)^{m-k},$$

the weights are nonnegative and sum to one. Thus the entire image lies in the convex hull of the complex control points b_k.

Theorem 13.2 (C02: A half-plane is a whole-arc certificate)**.** *If all b_k have strictly positive real part, strictly negative real part, strictly positive imaginary part, or strictly negative imaginary part, the whole polynomial arc avoids zero and is homotopic to its endpoint chord through arcs avoiding zero, with endpoints fixed.*

Proof. A convex combination preserves the selected strict coordinate sign. Both the polynomial arc and its chord lie in the same open convex half-plane. Their pointwise straight interpolation lies there too and fixes the common endpoints, giving the required homotopy. □

The coordinate-axis half-planes are intentionally simple witnesses. More general rational separating directions can reduce subdivision, but they are not needed for correctness. A producer that finds no witness must subdivide or return "not certified"; it must not infer a zero merely from a failed sufficient test.

13.2.2 Exact conversion and subdivision

If $q(t) = \sum_{j=0}^{m} a_j t^j$, the Bernstein coefficients are

$$b_k = \sum_{j=0}^{k} a_j \frac{\binom{k}{j}}{\binom{m}{j}}.$$

This follows from the identity

$$t^j = \frac{1}{\binom{m}{j}} \sum_{k=j}^{m} \binom{k}{j}\binom{m}{k} t^k (1-t)^{m-k},$$

which is the binomial theorem after extracting t^j. All coefficients remain rational complex numbers. Restricting to a dyadic subinterval can be done either by exact de Casteljau subdivision or by substituting its affine parametrization and reconverting. The shipped implementation uses the second form to let the checker recompute each piece directly from the original polynomial.

Theorem 13.3 (C03: Certificate acceptance implies the stated polynomial root count)**.** *Suppose a certificate partitions every oriented contour edge exactly into ordered subintervals, supplies a valid half-plane witness for each polynomial image arc, and states the exact winding of the resulting endpoint polygon. Then the stated integer equals the number of zeros of p inside Γ, with multiplicity, provided Γ is a simple positive contour.*

Proof. C02 supplies endpoint-fixed homotopies on every subinterval. They patch to a zero-free homotopy of the entire image loop to the endpoint polygon. Winding is unchanged by M11. The exact polygon crossing formula computes that winding. Since p is holomorphic and has no contour zero, M13 identifies it with the zero count. Simplicity/orientation of the *input* contour supplies the intended inside domain; the image polygon itself may self-intersect. □

The supplied examples use axis-aligned rectangles whose simplicity and orientation are immediate. The general checker accepts a polygonal loop but does not silently prove that an arbitrary input loop is Jordan. For a nonsimple source contour, its winding still has meaning but counts zeros with the contour's local winding weights, not necessarily zeros in one simple interior.

13.2.3 Polygon winding without trigonometry

For consecutive nonzero rational complex vertices $a = (x_0, y_0)$ and $b = (x_1, y_1)$, let $c = x_0 y_1 - y_0 x_1$. The oriented crossing count of the positive real ray is

$$+1 \text{ if } y_0 \le 0 < y_1 \text{ and } c > 0; \quad -1 \text{ if } y_1 \le 0 < y_0 \text{ and } c < 0.$$

All other edges contribute zero. Half-open vertical inequalities count a ray vertex exactly once. The half-plane certificates already exclude an edge through the origin. Summing yields the winding. There is no inverse tangent, phase unwrapping threshold or rounding to the nearest integer in acceptance.

13.2.4 What an independent checker recomputes

The certificate contains the coefficient vector, rational source contour, edge number and endpoints of every subinterval, selected coordinate/sign, and claimed winding. The checker binds the coefficients and contour to the expected task, proves that the intervals tile each edge without overlap or omission, recomputes every Bernstein control point, checks strict signs and computes winding from scratch. It does not trust a claimed lower bound on clearance, a producer's cached control points, or a declared root count.

```
for each oriented contour edge:
    require its certificate intervals to tile [0,1]
    for each interval:
        derive p(a + (b-a)t) from original coefficients
        convert exactly to Bernstein coefficients
        require one strict common coordinate sign
recompute exact crossing winding of all image endpoints
require equality with the certificate's claimed integer
```

13.2.5 A continuum of polynomial inputs can share one certificate

Theorem 13.4 (C04: Convex-family extension)**.** *Suppose $p_0, \dots, p_r$ are rational complex polynomials and every control point of every p_j on a certified arc lies in the* same *open half-plane. Then every polynomial in their coefficientwise convex hull is nonzero on that arc. On a closed certified contour the entire convex family has the same winding.*

Proof. Evaluation and Bernstein conversion are linear in polynomial coefficients. A convex combination of polynomials produces the same convex combination of each control point. The shared half-plane is convex, so all those control points and their arcs stay in it. Along a straight parameter path to p_0, the whole loop remains nonzero; M11 makes winding constant. □

This is a genuinely universal parameter certificate, although its finite witness is rational. It must not be described as merely testing the endpoints of an uncertainty interval: the argument explicitly encloses every interpolating coefficient vector. Conversely, independently nonzero endpoint polynomials do not suffice when their control points lack a common separating half-plane.

The executed gain-family certificate covers

$$p_K(s) = s^3 + 6s^2 + 11s + 6 + K, \qquad 0 \le K \le 50,$$

on the rectangle $[0, 10] + i[-10, 10]$. It uses 20 certified arcs and 160 recomputed control points and has winding zero. A separate certificate at $K = 61$ has winding two. At $K = 60$, the factorization $(s + 6)(s^2 + 11)$ places two zeros on the imaginary-axis contour; the producer does not certify that forbidden boundary case.

13.2.6 Executed examples and the cost boundary

Case	Image arcs	Control points	Winding
$z^3 - 3z$, large square	20	80	3
$z^3 - 3z$, central square	4	16	1
$z^3 - 3z$, right-root rectangle	4	16	1
z^2, repeated-root square	16	48	2
p_K, all $0 \le K \le 50$	20	160	0
p_{61}, right-half-plane rectangle	24	96	2

These values are read from newly executed certificate checks, not from a numerical root plot. Four deliberately altered certificates per case were rejected: a wrong integer, an inverted sign witness, an omitted interval and changed coefficients. Twenty-four rejected mutations do not establish that every conceivable malformed certificate is rejected; they test specific obligations.

For a degree-m polynomial and s accepted pieces, direct affine substitution and conversion cost $O(sm^2)$ rational arithmetic operations in this implementation, before bit complexity. Coefficient numerator/denominator sizes grow with input bits, degree and subdivision depth. A cost measured only in arithmetic operations is not a bit-complexity claim. The minimum nonzero separation can force many subdivisions. Single-run construction and checking times are retained as diagnostics; no broad speed superiority is inferred from six small cases.

Theorem 13.5 (C05: Termination in the fixed, zero-free polynomial case)**.** *For a fixed polynomial and compact polygonal contour on which it never vanishes, repeated sufficiently fine subdivision eventually admits coordinate half-plane witnesses everywhere.*

Proof. The image has positive minimum modulus. At each parameter value, at least one real or imaginary coordinate has magnitude at least this minimum divided by $\sqrt{2}$. By continuity it retains its sign on a small neighbourhood. Under restriction to a shrinking interval, every Bernstein control point of the fixed-degree polynomial converges uniformly to its value at the interval's limiting point; this follows directly from the affine-substitution formula. A finite subcover of the compact parameter interval and sufficiently fine uniform subdivision therefore produce the witnesses. This is an existence proof of termination, not a parameter-independent depth bound. □

For a convex family, absence of zeros for each parameter does not automatically make one coordinate witness work over a large parameter cell. Parameter subdivision may be necessary. The PG CSE branch gives the tensor-product formulation instead of making that stronger conclusion from the fixed-polynomial proof.

13.3 CSE-C: continuation with topology and conditioning in one record

13.3.1 Local root tubes

For $p(z, w) = z^2 - w$, suppose a candidate centre $c \neq 0$ is selected and the parameter lies in a set W. On $|z - c| = r$,

$$p(z, w) = 2c(z - c) + (c^2 - w) + (z - c)^2.$$

Theorem 13.6 (C06: A certified quadratic sheet tube)**.** *If* $\sup_{w \in W} |c^2 - w| + r^2 < 2|c|r$*, then each* $w \in W$ *has exactly one root in* $D(c, r)$*, and this selected root is holomorphic in the interior of* W *when* W *is a domain.*

Proof. The linear term dominates the other terms on the boundary, so Rouché gives one root counted with multiplicity. A count of one excludes a multiple root there. The implicit function theorem supplies local analytic branches, and uniqueness of the root in the same disk makes them agree on overlaps. □

A continuation algorithm covers a parameter path by such tubes and verifies an overlap/fibre correspondence at consecutive steps. The initial isolating disk determines the selected branch. The final root need not have the same label as the initial one after a closed path; its permutation is the monodromy output. A nearest-root heuristic without an exclusion margin is not an overlap certificate.

13.3.2 Predictor, corrector and independent verifier

A floating-point predictor uses $z' = w'/(2z)$. A corrector solves the polynomial approximately. The verifier then establishes a Rouché tube or exact contour count using rational enclosures. Predictor failure can reduce efficiency without invalidating an accepted result; verifier failure must prevent acceptance. This separation permits fast numerical construction while retaining an exact acceptance boundary.

The cost depends on both path geometry and distance from the discriminant. A path of short Euclidean length close to $w = 0$ can be harder than a longer path far away. Rescaling the variables can improve arithmetic conditioning but does not remove the topological obstruction to a global square-root label.

13.3.3 Why preserving a final value is insufficient

For two loops around distinct branch values, monodromy permutations may not commute. The endpoint of the base path can be identical while the selected root differs. A cache keyed only by the final parameter is therefore invalid for a path-dependent branch API. For invariant outputs, a cache keyed by the final parameter can be legitimate, but that is a different output contract. The source lifting theory identifies exactly which information the software must retain.

13.4 CSE-D: finite passports, Galois closure and state-space cost

13.4.1 A passport validator

Input consists of a degree, ordered branch loops, their permutations, a distinguished sheet and target topology. Validate permutation bijectivity, transitivity, the required product relation, cycle defects and the Euler formula. A source disk needs total defect $d - 1$ and a single boundary cycle. A compact sphere covering needs the product identity including the loop at infinity. These checks reject impossible or mislabelled data before any numerical surface realization.

The two degree-five examples in M18 both have disk source and total defect four, but their generated groups have orders 60 and 120 in the executed closure computation. Thus even with identical degree, number of critical points and Euler characteristic, regular-closure memory can differ. The group orders are finite calculations for those tuples; the existence of their holomorphic disk realizations still uses the written surface construction.

Theorem 13.7 (C07: An exact finite closure algorithm)**.** *Breadth-first closure of a finite generating set of permutations, starting at the identity and adjoining products by generators, terminates with precisely the generated subgroup. It stores at most $d!$ elements of S_d.*

Proof. Every produced element is a word in the generators, hence belongs to the generated subgroup. Termination follows from finiteness of S_d. At termination the set contains the identity and is closed under multiplication by each generator. In a finite group, inverse powers of each generator are positive powers, so all group words occur. Thus the set is exactly the subgroup. □

A straightforward tuple representation needs $O(d)$ memory per element and $O(d)$ work per composition. With r generators, this gives $O(r|G|d)$ elementary entry operations and $O(|G|d)$ storage, apart from hashing and integer representation. The bound is not an efficient algorithm for all large-degree groups; stabilizer-chain methods are a different algorithmic improvement, not implemented or benchmarked here.

13.4.2 Branch moves and the order of operations

An elementary Hurwitz move changes a pair (σ, τ) to $(\sigma\tau\sigma^{-1}, \sigma)$; the ordered product is preserved:

$$(\sigma\tau\sigma^{-1})\sigma = \sigma\tau.$$

This finite identity is checked for all pairs in the executed S_3 example. Preserving the product alone is not equivalence of the whole covering: loop bases, simultaneous conjugacy and the rest of the tuple must also be tracked. The PG CSE branch treats this as a state-rewriting problem with invariants, not a claim that every braid-equivalence question is solved by one product test.

13.5 CSE-E: invariant outputs can reduce computation without losing the question

13.5.1 Trace recurrence instead of explicit roots

For the cubic family in M23, compute s_n from three initial values and the recurrence $s_n = 3s_{n-2} + ws_{n-3}$. At a fixed parameter, computing through order N takes $O(N)$ scalar arithmetic and constant working storage when only the last three values are needed. Computing a selected root remains a different task. If polynomial coefficients in w are retained symbolically, polynomial degree and coefficient size enter the cost; calling that symbolic calculation constant cost per step would be false.

Theorem 13.8 (C08: Representation-independent trace under similarity)**.** *If A and $S^{-1}AS$ represent the same finite algebra element in different bases, then $\operatorname{tr} p(S^{-1}AS) = \operatorname{tr} p(A)$ for every polynomial p.*

Proof. Powers telescope to $(S^{-1}AS)^k = S^{-1}A^kS$. Trace is cyclic, so $\operatorname{tr}(S^{-1}A^kS) = \operatorname{tr} A^k$. Sum with the polynomial coefficients. □

The exact output is basis invariant; numerical conditioning need not be. An ill-conditioned S can amplify floating-point errors before the trace is formed. Algebraic invariance therefore motivates a good representation but does not certify a floating-point implementation without an error model.

13.5.2 A finite covering automaton

For the $d = 3$ power cover, encode each positive/negative turn of a base loop by a step $+1$ or -1 modulo three. The sheet state is a three-state automaton. Its adjacency matrix has zeros on the diagonal and ones elsewhere, with eigenvalues $2, -1, -1$. The number of length-n signed-turn words returning to a fixed sheet is

$$a_n = \frac{2^n + 2(-1)^n}{3}.$$

This counts a particular language of returning lifts; it does not prove equivalence of arbitrary automata. A direct dynamic-programming count is the baseline and the formula is an independent structural check. Its exact values were compared for $1 \le n \le 30$.

13.6 CSE-F: composition of partial programs

13.6.1 The domain is part of the result type

A partial analytic map can be stored as $(\Omega, f, \mathcal{E}, \mathcal{B})$, where Ω is its allowed domain, $\mathcal{E}$ is a validated evaluation rule and $\mathcal{B}$ gives branch/denominator information. A composable chain additionally proves that each enclosure is contained in the next allowed domain. The target-aware pullback sets of M24 give the exact mathematical semantics of that type.

Theorem 13.9 (C09: A correct final expression can be an incorrect partial program)**.** *Cancelling factors or composing formulas without preserving their original domains can change the partial function even when every returned value agrees wherever both programs are defined.*

Proof. The expression $(z^2 - 1)/(z - 1)$ is undefined at $z = 1$, while $z + 1$ is defined there. They agree at every other point. An API promising the original domain must retain that exclusion; an API intentionally returning the analytic extension must state the changed domain. The same logical distinction applies at each stage of a composition. □

The example is elementary, but it is exactly the kind of issue that becomes hard to detect after a chain of branch transformations. A symbolic simplifier can preserve an algebraic identity and still violate a domain-sensitive specification.

13.6.2 Error propagation and rejection

Use M27 with L_i bounded over the full enclosure. At each step, reject if the enclosure meets a forbidden pole/cut or leaves the next domain. An output tolerance is met only when the sum-of-products error bound is below the requested tolerance *and* all guards pass. Unknown is a legitimate result; silently replacing it by success destroys the certificate contract.

Theorem 13.10 (C10: A two-layer acceptance rule)**.** *If every stage's domain/evaluation certificate is sound and the arithmetic used for the bound is outward-enclosing, accepting only after all domain guards and the final error inequality hold proves the requested final enclosure. A final small residual alone does not imply this conclusion.*

Proof. Induction preserves the assertion that the exact state lies in the current enclosure. The stage certificate makes evaluation admissible and bounds its local error. M27 propagates the enclosure to the next stage. After the last step, its error inequality gives the requested output enclosure. The counterexample of an exact but wrong sheet has zero residual and fails the branch contract. □

Chapter 14

Finite certificates and exact representations

14.1 CU1: a square-root API needs a branch contract

For an input $w \neq 0$, the residual equation $z^2 - w = 0$ has two solutions. A routine called `sqrt` may specify a principal branch; a routine called `continue_root` must instead specify a starting root and a path. These are different APIs even when they call the same floating-point primitive internally.

An adequate root record contains the defining polynomial, the parameter cell, an isolating disk, multiplicity, and the predecessor cell or initial condition selecting the branch. A failure state should distinguish invalid input, boundary collision, unresolved precision and exhausted computation budget. Returning a complex number alone hides these distinctions.

Exercise 14.1 (CU1a)**.** For $w(t) = e^{it}$, begin at $z(0) = 1$. Give the continued root at $t = \pi, 2\pi, 4\pi$. At which endpoint does the first loop close in the base but not in the root state?

Solution. Continuity gives $z(t) = e^{it/2}$, so the values are $i, -1, 1$. The first base loop closes at 2π, while the lift ends at the other root. A nearest-root matching method can approximate this continuation, but without a separation and step-size condition it is a heuristic, not its proof.

Exercise 14.2 (CU1b)**.** Explain why a residual tolerance alone cannot ensure the correct branch. Use the exact pair $w = 9$, $z = 3$ and $z = -3$.

Solution. Both residuals are exactly zero. An initial condition selecting the positive branch has additional information that the equation alone does not encode. A residual checker validates equation satisfaction; it cannot recover a missing branch label. The same distinction appears in an implicit layer or an equilibrium solver.

14.2 CU2: a three-entry representation can retain a repeated root

Take the quotient algebra $\mathbb{C}[z]/((z-2)^3)$ and the basis $1, \varepsilon, \varepsilon^2$ with $\varepsilon = z - 2$. Multiplication by z is

$$M = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} = 2I + N, \qquad N^3 = 0.$$

For any polynomial h, Taylor expansion truncates exactly:

$$h(M) = h(2)I + h'(2)N + \tfrac{1}{2}h''(2)N^2.$$

The state dimension is three even though there is only one distinct root position. The representation contains first- and second-order information that a deduplicated list $\{2\}$ has thrown away. This is an algebraic-geometric fibre represented through finite linear algebra, not an instruction to diagonalize a defective matrix.

Exercise 14.3 (CU2a)**.** Compute M^2 and its trace. Compare the result with summing the squares of distinct root positions.

Solution. $M^2 = 4I + 4N + N^2$ has trace twelve. The sum over distinct positions is only four. The correct power sum includes the root twice more through its multiplicity. A program must say which quantity it computes before its answer can be judged correct.

14.3 CU3: permutation data have checkable invariants

Represent a permutation of $\{0, \dots, d-1\}$ by a tuple of length d. It must contain every label once. Composition is $(ab)[j] = a[b[j]]$. The following small routine computes its cycle count without confusing fixed points with absent cycles.

```
def cycle_count(p: tuple[int, ...]) -> int:
    if sorted(p) != list(range(len(p))):
        raise ValueError("not a permutation")
    seen: set[int] = set()
    count = 0
    for start in range(len(p)):
        if start in seen:
            continue
        count += 1
        j = start
        while j not in seen:
            seen.add(j)
            j = p[j]
    return count
```

For local monodromy p, the ramification defect is $d - \texttt{cycle_count}(p)$. A transposition has defect one; a three-cycle has defect two; the identity has defect zero. Counting only nontrivial cycles is wrong: the fixed sheets still contribute to the fibre count.

Exercise 14.4 (CU3a)**.** At degree four, compare (12)(34) and (123) with label four fixed. Compute their defects and explain why equal defects do not make the permutations conjugate.

Solution. Both have two cycles in total and hence defect two. Their cycle types are $(2, 2)$ and $(3, 1)$, respectively. Conjugation relabels cycles but preserves their lengths, so they are not conjugate. A total ramification number is a compressed invariant that may lose precisely the information a lifting problem needs.

14.4 CU4: a root count carried by four exact inequalities

Consider $p(z) = z^2 - 2$ on the positively oriented rectangle with vertices

$$1 - \tfrac{i}{2}, \quad 2 - \tfrac{i}{2}, \quad 2 + \tfrac{i}{2}, \quad 1 + \tfrac{i}{2}.$$

On the bottom edge, $\operatorname{Im} p(x - i/2) = -x < 0$. On the right edge, $\operatorname{Re} p(2 + iy) = 2 - y^2 \geq 7/4 > 0$. On the top edge, $\operatorname{Im} p(x + i/2) = x > 0$. On the left edge, $\operatorname{Re} p(1 + iy) = -1 - y^2 < 0$. Each complete image edge therefore lies in a strict coordinate half-plane avoiding zero.

The polynomial image can be deformed to the chord joining its endpoints inside that same half-plane. The endpoint polygon is

$$(-5/4, -1), \quad (7/4, -2), \quad (7/4, 2), \quad (-5/4, 1).$$

It contains zero once with positive orientation, as is checked by its ray crossing on the right edge. Thus its winding is one, and the polynomial has one root inside the rectangle. The argument principle is an imported analytic theorem; every finite inequality needed by this particular certificate is rational.

14.4.1 Why the checker is independent of a plotted curve

A visual trace could miss a small additional turn. Here the half-plane property excludes such a hidden homotopy change. The verifier rederives the polynomial restriction on each edge and checks the exact inequalities; it does not trust a claimed winding number stored in the file. The CSE application volume generalizes this hand calculation with Bernstein control polygons.

Exercise 14.5 (CU4a)**.** A certificate file changes the reported winding from one to zero while keeping every endpoint and edge inequality. What should the checker do? What if the polynomial coefficients are also changed without changing the expected task?

Solution. Recomputing the polygon winding detects the first change. Comparing the certificate's polynomial with the externally specified input detects the second. A certificate must be bound to the task; a correct proof about a different polynomial does not verify the requested computation.

Exercise 14.6 (CU4b)**.** Could a zero on a contour be handled by merely ignoring the offending sample? Explain using $p(z) = z - 1$ and the unit circle.

Solution. The zero occurs at $z = 1$, on the boundary. The entire-boundary nonvanishing hypothesis fails. Omitting that sample changes neither the polynomial nor the curve. One must change the contour with a justified exclusion or treat the boundary zero separately. A rounded winding from the remaining samples is not a valid argument-principle count.

14.5 CU5: a covering lift becomes a finite automaton

Consider $z^4 = w$ over $\mathbb{C}^*$. A positive loop around zero advances the sheet label by one modulo four. A binary word describes a sequence of actions: digit zero stays on the current sheet; digit one performs one loop. The final sheet is the number of ones modulo four. A lifted path closes exactly when that count is divisible by four.

Let $v_n(j)$ be the number of length-n action words ending on sheet j, starting on zero. The exact recurrence is

$$v_{n+1}(j) = v_n(j) + v_n(j - 1 \bmod 4), \qquad v_0 = (1, 0, 0, 0).$$

It uses four integer states and $O(n)$ arithmetic operations, without drawing paths or computing roots. A generating-function or Fourier representation is an optional mathematical compression, not a substitute for this semantic derivation.

For $n \geq 1$, a roots-of-unity filter gives

$$v_n(0) = \frac{1}{4}\sum_{k=0}^{3}(1+i^k)^n = 2^{n-2} + 2^{n/2-1}\cos(n\pi/4).$$

The right side is exactly an integer despite its analytic representation. For implementation, the integer recurrence avoids floating-point cancellation and transcendental evaluation.

Exercise 14.7 (CU5a)**.** Compute v_4 and $v_8(0)$ by the recurrence or binomial coefficients. Explain the exceptional case $n = 0$ in the displayed formula.

Solution. At length four, grouping by the number of ones gives $(2, 4, 6, 4)$. At length eight, the accepted counts are $\binom{8}{0} + \binom{8}{4} + \binom{8}{8} = 72$. For $n = 0$, the Fourier term $(1-1)^n$ represents one in the empty-word count, whereas the simplified formula omitted it as zero for positive n. The correct initial value is $v_0(0) = 1$.

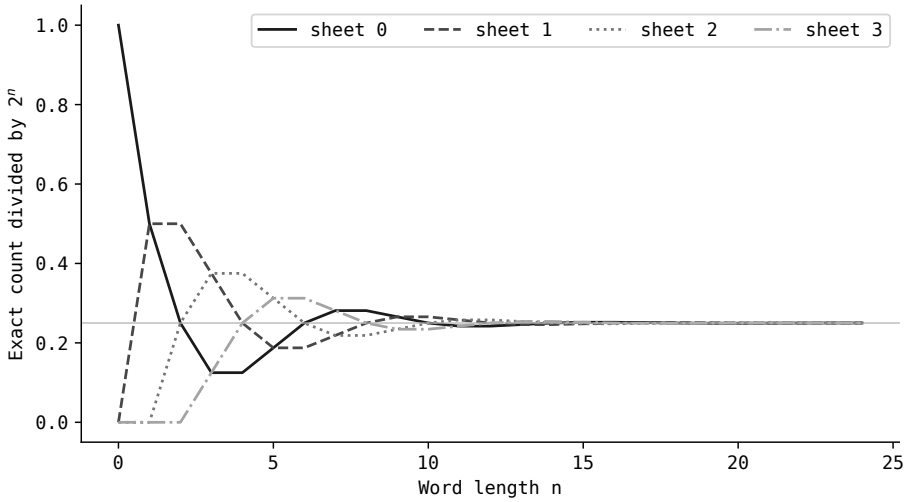


Figure 14.1: Exact finite-state counts normalized by the number of action words. The data are generated by the integer recurrence, not by numerical root tracking. The limiting balance does not discard the exceptional empty-word initial condition.

14.6 CU6: group enumeration and its cost

Breadth-first search from the identity under a finite list of permutation generators enumerates the generated subgroup. Each discovered permutation has a witness word. Closure follows when every generator edge from every discovered state returns to the set. The search terminates because there are at most $d!$ permutations, but this is not a polynomial-time bound in d.

A single-sheet orbit can be much smaller. For the natural action of S_3, there are six group states and three possible positions of label zero. Quotienting by the position of that label forgets the two-element stabilizer. Whether that loss is acceptable depends on the query: following one root needs less state than reconstructing a full root ordering.

Exercise 14.8 (CU6a)**.** For the cyclic group generated by (1234), compare group size and orbit size. For the group generated by (12) and (23), do the same. What does equality mean here?

Solution. The cyclic example has four group elements and an orbit of four, with trivial point stabilizer; its action is regular. The second example has six group elements and a three-element orbit, with stabilizer size two. Equal group and transitive orbit sizes mean that no nonidentity group element fixes the tracked sheet. A transitive action alone does not imply regularity.

14.7 CU7: guard a partial composition, then propagate its error

Suppose a routine A accepts inputs in U and guarantees outputs in V. A routine B accepts inputs only in $W \subset V$. The composite $B(A(x))$ is guaranteed only for $x \in A^{-1}(W)$, not for every $x \in U$. Source 3.6 constructs these pullback domains for univalent analytic stages; the same logical pattern is familiar in typed program contracts.

Algebraic simplification does not erase partiality. The expression $(z^2 - 1)/(z - 1)$ agrees with $z + 1$ for $z \neq 1$, but the original program may reject one. A rewrite that changes behavior at one is not semantics-preserving unless the contract explicitly permits filling a removable singularity.

Exercise 14.9 (CU7a)**.** Let $A(z) = 2z$ accept $|z| < 1$, and let $B(w) = 1/w$ accept $1/2 < |w| < 1$. Determine the guaranteed composite domain. State its output range.

Solution. The pullback condition is $1/2 < |2z| < 1$, so $1/4 < |z| < 1/2$. The output has $1 < |1/(2z)| < 2$. The excluded center is a genuine pole, not a removable algebraic artifact. The map's domain is annular; treating it as a disk changes its topology and its branch/continuation properties.

A numerical certificate adds an error disk around every intermediate value. The whole disk must lie inside the next input domain. Checking only the center is insufficient near a pole or branch cut. If an error bound is propagated through a Lipschitz constant, that constant must be valid on the entire disk. This joins source domain geometry to ordinary numerical-program verification without claiming that every software task is complex analytic.

14.8 Submission and machine-checking boundary

The branch's required artifact is a small proof-carrying root-count program with an exact input format, replayable certificate and rejected mutation tests. It must also report whether it counts distinct roots or multiplicities and whether it selects a branch. The finite recurrence and permutation checks can be exhausted for a declared range; exhaustion over that range is not an infinite theorem. The published written arguments justify the general forms. Lean/Coq candidates in the package illustrate universal algebraic identities but have not been compiled in this runtime. The exact rational checker was executed and its status is separately recorded.

14.9 Undergraduate comparison laboratory: exact input, exact claim

The purpose of this laboratory is to distinguish a program that produces a plausible answer from one that exports a checkable reason for that answer. The native undergraduate tools are arrays, rational arithmetic, loops, graph traversal and a clearly specified input/output contract. The complex geometry enters through a small number of proved invariants. These are proposed exercises linked to discrete mathematics, algorithms and numerical-computing material, not replacements for compiler, operating-system or security correctness.

14.9.1 CU8: a certificate has a producer and an independent consumer

A root solver may use floating approximations, heuristics and an adaptive search to propose a candidate count. A certificate consumer should instead receive a polynomial, an oriented contour, a complete ordered partition, a nonzero half-plane witness on every segment and an integer count. The consumer binds the polynomial and contour to the caller's task instead of trusting the certificate to redefine the input.

A minimal example. For $p(z) = z^2 + 1$ on the square with vertices $\pm 2 \pm 2i$, the expected root count is two. Both roots lie inside, but the checker must establish the result from its accepted contour witness instead of hard-code these roots. If one segment of the contour partition is deleted, the certificate is incomplete even when the reported count remains two. A correct final integer is not enough to validate its claimed proof.

Traditional versus proposed. A unit test compares one output with a known answer. A certificate verifies a theorem-specific condition at runtime. Both are useful: tests challenge the implementation, while the certificate explains the individual accepted result. Neither alone proves that the whole program is correct for every input. The v2 corpus contains mutation tests that deliberately corrupt otherwise valid certificates.

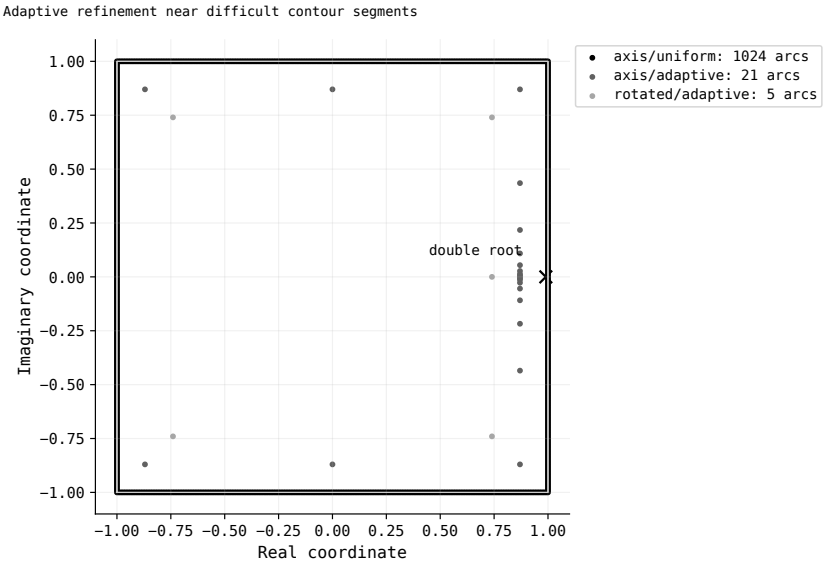


Figure 14.2: The same contour can be certified by a large uniform partition or a small locally refined one. The inward scaling is only for visibility; it does not change the actual task.

Exercise 14.10 (CU8a)**.** Explain why a certificate for one polynomial cannot be accepted for another polynomial merely because both happen to have the same count. Give one field the consumer must recompute.

Solution. The witness inequalities concern values of the declared polynomial on the declared contour. Reusing them for another polynomial is a different proposition. The consumer must compare the exact input coefficients with the certificate's coefficients and recompute the Bernstein controls or other segment enclosure from those coefficients. Matching the final count is not input binding.

14.9.2 CU9: control points are not curve samples

For a quadratic curve written in Bernstein form,

$$q(t) = (1-t)^2 v_0 + 2t(1-t)v_1 + t^2 v_2, \quad 0 \le t \le 1,$$

all coefficients are nonnegative and add to one. Thus $q(t)$ lies in the triangle spanned by the three controls. The middle control is usually not $q(1/2)$; in fact $q(1/2) = (v_0 + 2v_1 + v_2)/4$.

A cheap check with a global conclusion. If all three controls have positive dot product with a fixed normal, the entire curve lies in the same open half-plane. Checking three sample points would not establish the same fact for a general curve. The power of the finite check comes from the exact representation theorem, not from the number three.

Exercise 14.11 (CU9a)**.** Let $v_0 = (1, 0)$, $v_1 = (1/4, 1/4)$ and $v_2 = (0, 1)$. Find a strict rational separating normal and a rational Euclidean lower bound for $|q(t)|$.

Solution. Take $n = (1, 1)$. The dot products are $1, 1/2, 1$, so $n \cdot q(t) \ge 1/2$. Since $\|n\|_1 = 2$, C11 gives $|q(t)| \ge 1/4$. This is conservative but valid on the entire parameter interval. Strict positive-coordinate tests fail at the endpoints; the oblique normal succeeds.

14.9.3 CU10: arrays can represent an algebra, not only a root list

In $\mathbb{Q}[z]/(z^3 - 2)$, represent $a + bz + cz^2$ by the length-three array (a, b, c). Multiplication reduces every z^3 to two. Multiplication by z sends (a, b, c) to $(2c, a, b)$, so the matrix shown earlier can be implemented without any complex root approximation.

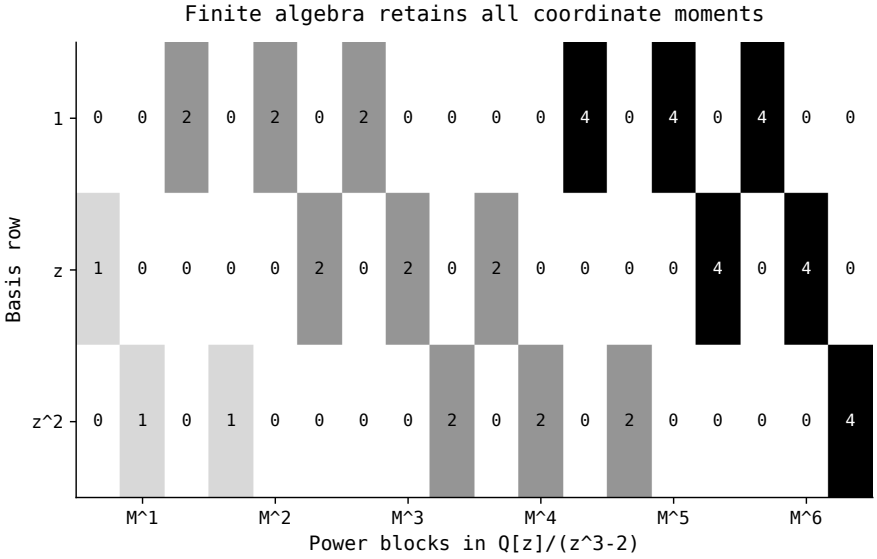


Figure 14.3: Six exact multiplication-power blocks. The displayed entries can be checked with integer arithmetic; no root ordering or numerical eigenvector is involved.

Compare the outputs. For the trace $\mathrm{tr}(M_z^6)$, the relation $M_z^3 = 2I$ gives $M_z^6 = 4I$ and trace twelve. Computing three cube roots and summing their sixth powers is an alternative route, but introduces unnecessary branch and arithmetic issues for this exact-integer task. If a plot needs the individual root coordinates, the quotient trace does not provide that output.

Exercise 14.12 (CU10a)**.** Give a recurrence for $\mathrm{tr}(M_z^n)$ and compute it for $n = 0, 1, 2, 3, 4, 5, 6$. State which values vanish for structural reasons.

Solution. Set $t_n = \operatorname{tr}(M_z^n)$. Then $t_0 = 3$, $t_1 = t_2 = 0$, and $t_{n+3} = 2t_n$. The values are $3, 0, 0, 6, 0, 0, 12$. All indices not divisible by three vanish. This follows from the algebraic relation and initial traces, not from numerical cancellation among roots.

14.9.4 CU11: a group search need not allocate the factorial bound

Given generators as permutations of three symbols, a breadth-first closure search stores only reached permutations. With generators (12) and (23), six permutations are reached. With only the three-cycle (123), three are reached. The degree-three fibre is the same size in both cases, but the generated ordering state spaces differ.

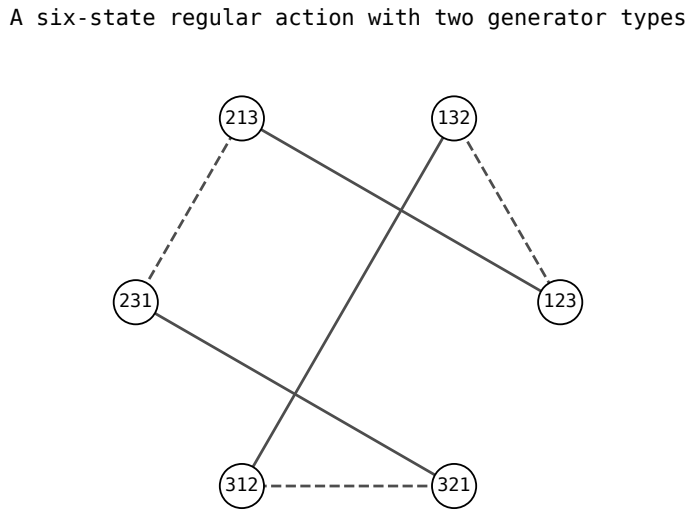


Figure 14.4: The two-generator action on six orderings. Solid and dashed edges represent different generators; product order and edge convention are part of the data.

Metric. Count generated states, stored entries and composition operations. Reporting only "the bound is $d!$" hides the savings available when the actual subgroup is smaller. Conversely, using only d sheet states may be wrong when the task asks for the full root ordering or the regular cover. A smaller data structure is useful only if it retains the observable requested by the caller.

14.9.5 CU12: the claim determines the benchmark

The laboratory submission should contain one valid input, one almost-invalid input, one invalid input and one corrupted certificate. Record candidate construction time separately from consumer verification time. Count exact rational control points and serialized bytes alongside runtime. An optimization that simply stops checking difficult inputs will look fast unless accepted coverage and rejection reasons are also reported.

The lecture now places a short prerequisite card before each content slide. These cards define the exact mathematical objects needed on the next slide, give one small calculation, and provide a self-check answer. They are support for the argument, not independent new mathematical results counted toward the corpus's novelty or theorem totals.

Chapter 15

Algorithmic geometry and continuation

15.1 CG1: Bernstein coefficients are a proof representation

For a polynomial $q(t) = \sum_{j=0}^{m} a_j t^j$, write

$$q(t) = \sum_{k=0}^{m} b_k \binom{m}{k} t^k (1-t)^{m-k}, \qquad b_k = \sum_{j=0}^{k} a_j \frac{\binom{k}{j}}{\binom{m}{j}}.$$

For real $0 \le t \le 1$, the basis weights are nonnegative and sum to one. Thus a complex polynomial path lies in the convex hull of its complex control points b_k. If one real linear functional $\ell(z) = \alpha \operatorname{Re} z + \beta \operatorname{Im} z$ is strictly positive on all b_k, it is strictly positive on the whole path. This is the link from finite arithmetic to nonvanishing on a continuum.

15.1.1 Degree elevation preserves the certificate

The same polynomial expressed in degree $m+1$ has coefficients

$$b'_k = \frac{k}{m+1} b_{k-1} + \left(1 - \frac{k}{m+1}\right) b_k,$$

with the natural endpoint conventions. To prove this, multiply each Bernstein term by $t + (1-t)$ and regroup adjacent terms using the binomial identities. Each new coefficient lies on a segment between old coefficients. Therefore a valid strict half-plane certificate survives degree elevation. A failed convex-hull exclusion may become successful after subdivision, but simply changing basis degree cannot validate an originally nonzero value that lies in a genuinely wrong domain.

Exercise 15.1 (CG1a)**.** For $q(t) = 1 + it - t^2$, find its degree-two Bernstein coefficients and a strict half-plane certificate, or explain why the coordinate real half-plane test alone fails.

Solution. The coefficients are $b_0 = 1$, $b_1 = 1 + i/2$, $b_2 = i$. Their real parts include zero, so the strict positive-real test fails. Their imaginary parts also include zero. The functional $\ell(z) = \operatorname{Re} z + \operatorname{Im} z$ is positive on all three, with values $1, 3/2, 1$. It certifies the path in a rotated half-plane. The release's implemented checker uses coordinate half-planes only; this example specifies a sound extension beyond that implementation, not an executed capability already present.

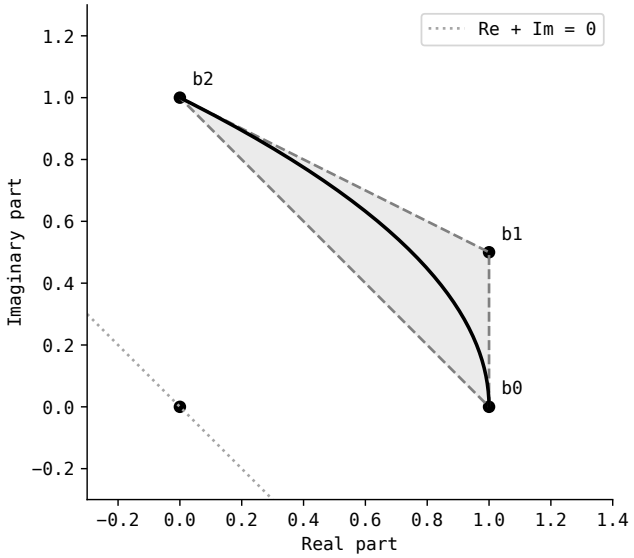


Figure 15.1: The curve and its exact Bernstein hull. Neither coordinate half-plane supplies a strict witness for all control points, but the rational direction Re+Im does. The currently implemented checker does not yet accept arbitrary rotated directions.

15.2 CG2: subdivision and a sound checker

The de Casteljau recursion gives an exact subdivision of a Bernstein curve at any rational $s \in [0, 1]$:

$$b_k^{(0)} = b_k, \qquad b_k^{(r+1)} = (1 - s)b_k^{(r)} + sb_{k+1}^{(r)}.$$

The left and right boundary diagonals of this triangle are the coefficients of the two subcurves. The recursion follows from repeated affine interpolation; induction on degree verifies equality of the represented polynomial. With rational inputs and rational s, every coefficient remains rational complex.

A verifier need not trust the producer's subdivision arithmetic. The delivered checker derives the restricted polynomial directly from its coefficients and each declared rational source segment, reconverts to Bernstein form and verifies the half-plane witness. It also checks that segments tile each original edge exactly, in order, without overlaps or gaps. This redundancy makes a malformed subdivision certificate detectable even when its advertised winding looks plausible.

Exercise 15.2 (CG2a)**.** Why does checking every declared segment independently not suffice if the certificate can omit an interval? Give the missing logical implication.

Solution. Local exclusion proves nonvanishing only on the union of the declared segments. The argument principle requires nonvanishing on the entire closed boundary. Exact tiling supplies the equality between that union and the requested contour. Without it, the missing interval can contain a zero or an extra winding. Certificate soundness has both arithmetic and combinatorial premises.

15.3 CG3: separate root counting from branch following

Suppose $p(z, w)$ has exactly one zero in an isolating disk for each w in an open connected parameter cell, and that the root remains simple. The implicit-function theorem supplies local analytic roots; uniqueness in the same disk makes the local branches agree on overlaps. This

constructs one analytic branch on that cell. It does not supply a global label around a parameter loop whose monodromy is nontrivial.

For $z^2 = w$, use the exact test

$$|c^2 - w_0| + h + r^2 < 2|c|r$$

when $|w - w_0| \le h$ and the proposed root disk is $|z - c| < r$. The left side bounds the nonlinear remainder relative to $2c(z - c)$. It is a quantitative Rouché condition, not a fitted convergence rate.

Exercise 15.3 (CG3a)**.** Take $c = 3/2$, $w_0 = 9/4$, $r = 1/2$ and $h = 1$. Certify the branch and a uniform derivative bound.

Solution. The center residual is zero. The remainder bound is $1 + 1/4 = 5/4 < 3/2$. Exactly one root lies in the disk for every parameter in the cell. Its modulus exceeds one, so $|dz/dw| = 1/(2|z|) < 1/2$. The strict inequality gives an open neighborhood on which the same branch exists. The other root is excluded by the disk instead of by a floating-point sign comparison.

15.3.1 A continuation-step policy

Choose a candidate center c by any efficient predictor. The checker evaluates its residual enclosure, selects a rational r and validates the above inequality on the next parameter cell. With the conservative choice $r = |c|/2$, the allowable parameter radius satisfies

$$h < \tfrac{3}{4}|c|^2 - |c^2 - w_0|.$$

This exposes the shrinking step budget near the branch point. In exact rational implementations, enclose $|c|$ from below and use squared inequalities where appropriate. A predictor failure should lead to refinement or a classified unresolved cell, not to silently switching to a different root.

15.4 CG4: multiplication matrices avoid unstable root labels

For a monic polynomial $p(z) = z^d + c_{d-1}z^{d-1} + \cdots + c_0$, multiplication by z in $\mathbb{C}[z]/(p)$ has a companion matrix. Its characteristic polynomial is p. A polynomial observable can be represented by matrix reduction without computing individual roots.

For $p(z) = z^3 - w$, the basis $1, z, z^2$ yields

$$A(w) = \begin{pmatrix} 0 & 0 & w \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad A(w)^3 = wI.$$

Hence powers reduce to three residue classes modulo three. An entire function $f(z) = \sum a_n z^n$ has

$$f(A) = F_0(w)I + F_1(w)A + F_2(w)A^2, \quad F_j(w) = \sum_{n \ge 0} a_{3n+j} w^n.$$

For the exponential, all three F_j are entire. Their representation remains analytic at $w = 0$, where the eigenvalue labels collide and the matrix is nilpotent. This does not eliminate all matrix conditioning, but it removes artificial branch choices from a symmetric observable.

Exercise 15.4 (CG4a)**.** Compute $\operatorname{tr} A^n$ for all $n \ge 0$ and explain the case $n = 0$.

Solution. The trace is zero when n is not divisible by three. For $n = 3m$, $A^n = w^m I$, so the trace is $3w^m$, including three at $n = 0$. This is an exact recurrence. Numerically summing independently labelled roots can add branch-matching complexity to a quantity that never required the labels.

15.5 CG5: a regular closure is an automaton with algebraic invariants

Given permutations $\sigma_1, \ldots, \sigma_r$, enumerate the generated group with BFS. Store a parent edge and generator for every new state. A checker validates the identity, each witness word, distinctness, and closure under every generator. Closure plus reachability proves that the listed set is exactly the generated subgroup: no listed element is spurious and no reachable element is missing.

If the group has g elements represented as arrays of length d, elementary composition costs $O(d)$, so the basic enumeration takes $O(rgd)$ array operations plus dictionary overhead and $O(gd)$ storage. The worst-case $g = d!$ must remain visible. Reporting the number of root sheets as the group-state count can understate the closure by a large factor.

Exercise 15.5 (CG5a)**.** Why are the two checks "every element has a witness word" and "the set is closed under generators" both needed?

Solution. Witnesses alone permit an incomplete prefix of the generated group. Closure alone permits a larger group containing all generators, including unreachable elements under an incorrectly initialized state set. Together with the identity and distinct valid permutations, they establish both inclusions. This is analogous to the paired containment checks needed in a certificate for a geometric image.

15.5.1 A chain-complex consistency check

A candidate triangulated cover can store boundary matrices ∂_2 and ∂_1. The exact identity $\partial_1 \partial_2 = 0$ is necessary for a chain complex. Euler characteristic can be computed from cell counts and checked against alternating homology ranks over a field. These finite checks harden a proposed combinatorial model, but do not by themselves prove that its geometric realization is a surface, that the map is locally a power, or that no branch data were omitted. Source 3.2 supplies those additional topological obligations.

15.6 CG6: truncation error in a Hardy-space computation

For an analytic $f(z) = \sum a_n z^n$ with $|a_n| \le M R^{-n}$, $R > 1$, the tail after degree N satisfies

$$\left\| f - \sum_{n=0}^{N} a_n z^n \right\|_{H^2}^2 \le M^2 \frac{R^{-2(N+1)}}{1 - R^{-2}}.$$

This follows by Parseval and summing a geometric series. The hypotheses concern a whole coefficient family, not a finite observed decay pattern. The estimate is an H^2 mean-square boundary bound; it is not a uniform bound on the closed unit disk without additional reasoning.

For $|z| \le r < 1$, Cauchy–Schwarz converts a known H^2 tail norm δ to

$$|\mathrm{tail}(z)| \le \frac{\delta\, r^{N+1}}{\sqrt{1 - r^2}}$$

where $\delta^2 = \sum_{n>N} |a_n|^2$; shifting the coefficient sequence does not change this norm. Ignoring the tail's zero of order $N+1$ gives the coarser bound $\delta/\sqrt{1-r^2}$. The sharper factor comes from the known vanishing order, not from redefining the same norm.

Exercise 15.6 (CG6a)**.** For $f(z) = 1/(1-z/2)$, calculate the exact H^2 tail squared norm after degree N and the exact pointwise tail at real $z = r \in [0, 1)$.

Solution. The coefficients are 2^{-n}. The squared tail norm is $4^{-(N+1)}/(1-1/4) = 4^{-N}/3$. The pointwise tail is $(r/2)^{N+1}/(1-r/2)$. The first is an average-energy statement, the second a pointwise formula; neither should be substituted for the other without its stated conversion.

15.7 CG7: error balls must fit the next domain

Let an approximate intermediate value be $\hat{z}$ with error radius ε. Before evaluating an inverse branch defined on V, verify $\overline{D}(\hat{z}, \varepsilon) \subset V$. If $V = \{z : |z-a| > \rho\}$, the sufficient exact condition is $|\hat{z}-a| - \varepsilon > \rho$. If V is a slit domain, the distance to both the cut and its endpoint matters. The derivative bound used for propagation must hold on the entire input ball.

Exercise 15.7 (CG7a)**.** For inversion $\psi(z) = 1/z$, assume $|\hat{z}| \geq m > \varepsilon$. Derive a bound on $|\psi(z) - \psi(\hat{z})|$ for $|z - \hat{z}| \leq \varepsilon$.

Solution. Direct subtraction gives $|z-\hat{z}|/(|z||\hat{z}|)$. Since $|z| \geq m - \varepsilon$ and $|\hat{z}| \geq m$, the error is at most $\varepsilon/(m(m-\varepsilon))$. This is sharper than using an unnecessarily large global derivative bound and explicitly fails as the certified ball approaches the pole.

15.8 Deliverable and verification standard

A graduate submission should implement one producer and a deliberately smaller verifier, prove the verifier's implication, and provide malformed, boundary and near-critical tests. It should report exact arithmetic costs separately from plotting and the construction heuristic. The release already executes one such polynomial contour verifier. Rotated half-plane witnesses, tensor-parameter cells, general rational-function poles and a fully formalized verifier are described extensions, not claimed executed features. The claim ledger makes that implementation boundary explicit.

15.9 Graduate comparison: algorithmic work and certificate strength

The graduate computational objective is to preserve a common semantic contract while changing the representation. A fair comparison records accepted problem scope and error or proof guarantees before measuring runtime. The new experiments are reproducible but deliberately small; they do not establish performance on every polynomial, matrix or parameter family.

15.9.1 CG8: adaptive geometry has two separable effects

The first effect is partition locality: refining only difficult contour arcs. The second is enclosure direction: accepting arbitrary rational separating normals instead of four coordinate directions. The benchmark varies both independently. This four-way design avoids attributing all gains to the more visually striking geometric change.

Observed size result. For $(z - 99/100)^2$ on the unit square, axis/uniform, axis/adaptive, rotated/uniform and rotated/adaptive use $1024, 21, 8, 5$ arcs. Thus adaptivity alone changes 1024 to 21, rotation alone at a uniform depth changes 1024 to 8, and their combination reaches 5. Those effects are not additive percentages. For the cubic workload, rotated uniform and rotated adaptive coincide at eight arcs.

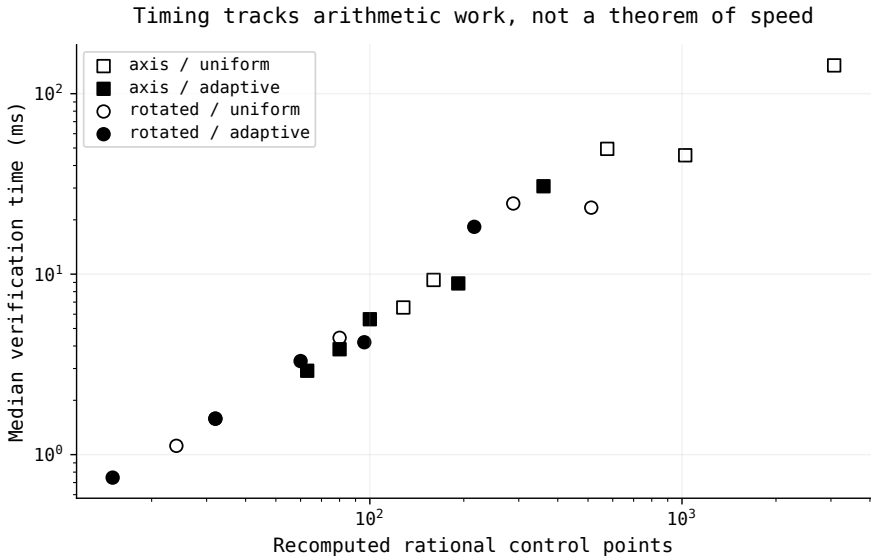


Figure 15.2: Verifier cost versus recomputed control points. Open markers are uniform and filled markers adaptive; circle and square shapes distinguish normal languages. Each point is an accepted certificate for a fixed task.

Cost derivation. Let L be the number of accepted arcs, d the polynomial degree and V the number of coefficient-family vertices. A direct control recomputation uses $O(LVd^2)$ rational operations. If all numerators and denominators have at most b bits, a corresponding bit estimate must include the multiplication and normalization cost at size b. Since subdivision can increase b, plotting L alone does not prove the bit complexity. The current exact implementation exports the certificate itself so these sizes are inspectable.

Exercise 15.8 (CG8a)**.** Two methods have $(L, b) = (20, 100)$ and $(5, 1000)$. Under the toy multiplication model $M(b) = b^2$, which has smaller $LM(b)$? What does this say about an arc-only claim?

Solution. The costs are 200,000 and 5,000,000. The five-arc method is more expensive under this model despite its smaller partition. The toy model is not a runtime measurement; it demonstrates that arc count alone cannot logically determine cost when coefficient sizes change.

15.9.2 CG9: invariant matrix evaluation with a controlled truncation error

For the family $A^2 = \lambda I$, the quotient series in C15 has a uniform exact-arithmetic tail certificate on $|\lambda| \leq 1$. Horner evaluation of the even and odd coefficient polynomials costs $2m$ scalar multiply-add steps plus a few matrix-entry assignments. This avoids a matrix diagonalization and is straightforward to vectorize across many parameters. The upfront proof of the relation and the domain is part of the setup cost.

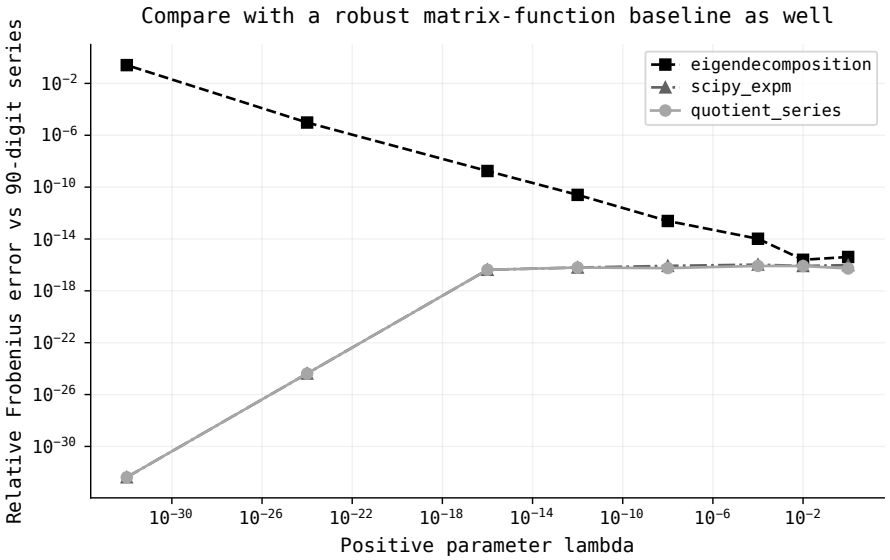


Figure 15.3: Actual float64 errors against an independent 90-digit reference. The general matrix-exponential routine is included to avoid a comparison only against a weak near-defective eigenbasis method.

Roundoff is a second theorem. The coefficient truncation bound below 7.55×10^{-33} is far below double precision. A practical total bound must add coefficient conversion and arithmetic errors. A generic Horner forward bound of the form $\gamma_{2m} \sum |c_k \lambda^k|$ requires a stated floating-point model, no underflow/overflow pathologies, and suitable treatment of complex arithmetic. The release does not claim a formal float64 proof of that model; it reports observed errors separately.

Exercise 15.9 (CG9a)**.** Why can the quotient method remain correct at $\lambda = 0$ even though a formula containing $\sinh(\sqrt{\lambda})/\sqrt{\lambda}$ appears undefined? State the value of the scalar quotient there.

Solution. Its series is $\sum_{k \geq 0} \lambda^k/(2k+1)!$, which has value one at zero. The quotient is an entire function with a removable singularity in the square-root notation. Evaluating it through its series avoids a spurious division by zero. The resulting matrix exponential is $I + A(0)$.

15.9.3 CG10: contour certificates and root tubes are different interfaces

A contour certificate proves a count inside a region. A root tube additionally preserves the identity of a particular root along a parameter path. The count can remain two while the labels exchange. A complete continuation record therefore includes a starting root, a path cell, an enclosure, uniqueness or cluster multiplicity, and overlap or monodromy information between cells.

A quantitative tube. For $p(z,t) = z^2 - t$ and a known base root $z_0^2 = t_0$, write $z = z_0 + u$ and $t = t_0 + v$. On $|u| = r$,

$$p(z,t) = 2z_0 u + u^2 - v.$$

The strict condition $r^2 + |v| < 2|z_0|r$ certifies one root in the disk around z_0. This is the same inequality already used in CG3, now interpreted as a parameter-step policy. It degenerates as $z_0 \to 0$; that is genuine local branch geometry, not just a slow implementation.

Comparison. Predictor-corrector iteration is a construction method. The inequality is a posteriori verification. Combining them can be superior to replacing the solver with a fully

symbolic formula, but the cost must include both construction and checking. An unverified predictor that rarely fails does not satisfy the same acceptance contract as a certified tube.

15.9.4 CG11: joint parameter geometry versus a list of point tests

A polynomial family affine in a real parameter is a convex combination of its endpoints. The joint Bernstein control list can therefore give a whole-interval certificate. Testing one hundred parameter values only establishes one hundred instances; it does not give the same output. The advantage is coverage of a continuum, often with one finite witness. The limitation is conservatism of the joint hull: a too-large parameter region can defeat one common separator even if each member is admissible.

For nonlinear parameter dependence, use a valid tensor-product Bernstein enclosure, a proved convex relaxation, or a subdivision with an explicit remainder. Sampling endpoint coefficients alone is not generally a valid enclosure. The CSE research branch isolates the statement that the parameter representation must satisfy before any contour checker is called.

15.9.5 CG12: a reproducible result is not just a median

The benchmark stores exact input coefficients and contour vertices, the full certificate, raw timing repetitions, generator and verifier costs, and acceptance diagnostics. Warmup and repeated measurements reduce some noise; they do not make a confidence interval without a statistical design. The reported minimum and maximum are observed ranges. Changing the interpreter, rational arithmetic backend, degree distribution or normal-search strategy can change rankings.

A substantive improvement has two routes: a proved inequality or capability result for a family, or a measured result on a clearly stated workload. The former needs a theorem; the latter needs repeatable data. The revised lecture comparisons label these routes so a graph of measured speed is not mistaken for an asymptotic proof.

Chapter 16

Uniform certificate construction and complexity

16.1 CP1: a quantitative termination condition for one polynomial

Fix a polynomial p of degree m and a compact piecewise-linear contour on which $|p| \geq \mu > 0$. On a segment starting at z_0 with increment h, its power coefficients under $z = z_0 + ht$ are $a_j = p^{(j)}(z_0)h^j/j!$. The degree-m Bernstein coefficient b_k differs from $p(z_0)$ by at most

$$R(h) = \sum_{j=1}^{m} \frac{M_j |h|^j}{j!}, \qquad M_j \geq \sup_{\Gamma} |p^{(j)}|.$$

Indeed the conversion factors $\binom{k}{j}/\binom{m}{j}$ lie between zero and one. At least one of $|\operatorname{Re} p(z_0)|$ and $|\operatorname{Im} p(z_0)|$ is at least $\mu/\sqrt{2}$. If $R(h) < \mu/\sqrt{2}$, all control points retain the strict sign of that coordinate. A coordinate-half-plane witness therefore exists for every sufficiently short segment.

This proves termination of recursive subdivision for a fixed zero-free polynomial, assuming exact arithmetic and no externally imposed depth cap. It does not give a parameter-independent bound when $\mu \downarrow 0$. A practical implementation can replace $\mu/\sqrt{2}$ by an exact lower bound on the chosen endpoint coordinate and verify the local inequality directly.

Exercise 16.1 (CP1a)**.** For $p(z) = z^2 - 2$ on a region $|z| \leq R$, bound $R(h)$ using only R and $|h|$.

Solution. $|p'| = 2|z| \leq 2R$ and $|p''| = 2$, with higher derivatives zero. Thus $R(h) \leq 2R|h| + |h|^2$. A strict comparison with the endpoint coordinate margin gives a completely algebraic local subdivision rule. It is conservative, but its conclusion covers the whole segment.

16.2 CP2: a nonzero family can defeat an axis-only witness

Consider the constant-in-z family

$$p_t = (-2 + i)(1 - t) + (1 - 2i)t, \qquad 0 \leq t \leq 1.$$

Every value lies on the line $\operatorname{Re} p_t + \operatorname{Im} p_t = -1$, so zero is uniformly excluded. Yet across the entire parameter interval the real parts have both signs and the imaginary parts have both signs. No single coordinate half-plane contains both endpoint values. Subdividing the source contour cannot help because the polynomial is constant there.

A rational separating functional does help: $\ell(z) = -\operatorname{Re} z - \operatorname{Im} z$ equals one throughout the family. Alternatively split the parameter interval at $1/2$. The first half has strictly negative real

part, and the second has strictly negative imaginary part. Thus coordinate witnesses become complete after suitable joint parameter subdivision, even though source-only subdivision is not.

Implementation consequence. The delivered checker is sound for its coordinate witnesses and succeeds on the particular gain family included in the tests. It is not claimed to certify every nonvanishing convex polynomial family. This counterexample is a positive control against overstating its completeness. Extending witness directions to rational (α, β) is straightforward for checking, but the producer's search strategy and bit complexity still require analysis.

Exercise 16.2 (CP2a)**.** Show that the family stays at Euclidean distance at least $1/\sqrt{2}$ from zero, and identify where equality occurs.

Solution. By Cauchy–Schwarz, $|x + y| \le \sqrt{2}\sqrt{x^2 + y^2}$. Since $x + y = -1$, the norm is at least $1/\sqrt{2}$. Equality occurs at $x = y = -1/2$, namely $t = 1/2$. The failure of a coordinate witness is therefore not caused by a tiny nonvanishing margin.

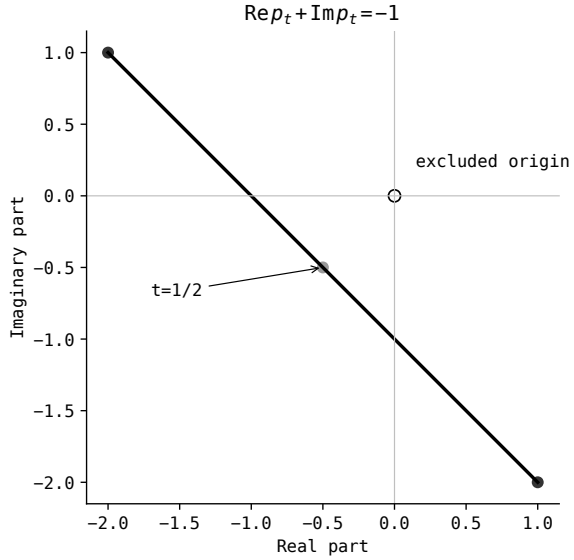


Figure 16.1: A uniformly zero-free parameter segment that defeats an axis-only global witness. The obstruction belongs to the certificate representation, not to the mathematical nonvanishing property. Parameter subdivision or a rational rotated witness repairs it.

16.3 CP3: joint Bernstein certificates for parameter boxes

Let $q(t, u)$ be polynomial on $[0, 1]^2$, with tensor-product Bernstein expansion

$$q(t, u) = \sum_{j=0}^{m} \sum_{k=0}^{n} b_{jk} B_j^m(t) B_k^n(u).$$

The product weights are nonnegative and sum to one. A single strict rational half-plane witness for all b_{jk} therefore certifies nonvanishing over the entire two-dimensional cell. The same statement holds for any finite number of parameters. It is a finite convexity argument; the complexity grows with the tensor-product degree and the required subdivision.

Local certificates can use different witness directions on neighboring cells. To infer a single root count over the entire parameter region, one also needs connectedness and a covering by valid cells whose union is the claimed domain. To infer a labelled branch, one additionally needs compatible

isolating regions and monodromy control. These are increasingly strong conclusions and must not be merged into one success flag.

Exercise 16.3 (CP3a)**.** For $q(t, u) = 1 + t + iu$, give a uniform certificate over the unit square and explain why checking only its four corners would not be a general proof for an arbitrary nonlinear polynomial.

Solution. The real part is $1 + t \geq 1$. Since this polynomial is affine in each variable, its tensor Bernstein control points are its corner values and have real parts one or two. For higher degree, corner values need not bound the interior; an interior zero can occur while all corners are nonzero. The certificate uses all control points of the declared degree, not merely convenient samples.

16.4 CP4: separating arithmetic, topology and implementation soundness

The implemented certificate format contains the expected polynomial vertices, an oriented polygon, an ordered rational subdivision, a coordinate witness for every piece and a claimed integer. The verifier rederives the piece coefficients, validates strict signs and exact tiling, and recomputes the polygon winding. Its logical implication is conditional on the source contour being the intended positively oriented Jordan boundary. For the supplied axis-aligned rectangles, that fact is direct from their construction; an arbitrary user polygon would require a separate exact simplicity/orientation check.

A full formal proof should have three interfaces. The algebraic layer shows that computed coefficients represent the restricted polynomial. The geometric layer shows that a certified convex hull gives a zero-free homotopy to a chord. The topological layer relates the exact polygon crossing number to winding and then to polynomial zeros. A proof of the first interface alone does not prove the final root count.

The recorded executable universal-identity checker uses a sparse rational coefficient dictionary. It accepts only rational constants, variables, addition, subtraction, multiplication and nonnegative integer powers. Its soundness argument is structural induction on that syntax. This is stronger than checking an identity at finitely many points, but it remains dependent on the Python interpreter, integer/rational implementation and the program's own correctness. It is not described as a proof-assistant kernel.

16.4.1 Malformed input is part of the theorem's boundary

A production parser should reject duplicate or missing fields, invalid rational denominators, nonfinite values, out-of-order cells, unbounded AST sizes and integer ranges that would exhaust resources. Mathematical soundness and denial-of-service resistance are distinct properties. The included polynomial AST imposes a node budget and exponent cap; it does not constitute a general secure document parser. The release's adversarial tests target incorrect mathematical certificates, not hostile internet deployment.

Exercise 16.4 (CP4a)**.** Why should a certificate's embedded polynomial not be accepted as the sole source of the task identity?

Solution. Otherwise a malicious or mistaken producer can replace the requested polynomial with an easier one and prove a correct fact about it. The verifier must compare the certificate with externally supplied expected coefficients, domains and conventions, or with an authenticated immutable task record. A checksum detects changes only relative to a trusted expected checksum; self-hashing does not authenticate a task.

16.5 CP5: a condition-aware complexity account

Let s be the number of accepted contour pieces, m the polynomial degree, v the number of coefficient-family vertices and B a bound on rational bit lengths. A direct coefficient restriction and Bernstein conversion costs $O(vm^2)$ rational operations per piece; the full check is $O(svm^2)$ such operations plus polygon crossings. A rational operation is not constant-time when B grows. Under a multiplication cost $\mathcal{M}(B)$, numerator/denominator normalization and coefficient growth must be included in a bit-complexity estimate.

The construction cost can greatly exceed checking cost because failed trial subdivisions are not part of the final certificate. The current timings are single-run diagnostics on a small fixed corpus, not a general benchmark or a proof of asymptotic superiority over interval Newton, Routh or established polynomial solvers. A valid performance study would freeze input families, tolerances, arithmetic precision, failure policy and machine configuration, then report both successes and unresolved cases.

Exercise 16.5 (CP5a)**.** Does a certificate with twice as many segments necessarily take twice as long to verify? Give two independent reasons why the answer can be no.

Solution. Its coefficients may have different degrees or numbers of parameter vertices, changing the operation count per segment. Its rational numbers may also have larger bit lengths, changing the cost of each arithmetic operation. Cache effects and interpreter overhead are further empirical factors, but the first two already show that segment count alone is an incomplete cost metric.

16.6 CP6: labelled continuation and label-free observables

For an analytic matrix family with a root collision, individual eigenvalue branches can permute around a loop. A symmetric observable may remain analytic. The quotient-algebra representation in the graduate branch computes traces and entire matrix functions without choosing roots, but it cannot answer a query that genuinely asks for one transported mode without additional branch information.

A branch-aware data structure should therefore offer two distinct types: a symmetric fibre observable and a based continuation section. Converting the latter to the former forgets labels. Reconstructing a labelled section from the symmetric value is generally impossible. This is an information-loss statement, not a numerical defect.

Exercise 16.6 (CP6a)**.** For roots of $z^2 = w$, compare the information in their sum, their unordered pair and a root continued from $+1$ around a loop based at $w = 1$.

Solution. The sum is always zero and contains no branch information. The unordered pair determines w but is unchanged after the loop. The continued root records whether the loop's winding is even or odd, ending at $+1$ or -1. The three outputs solve distinct problems. A model that predicts only symmetric features cannot be credited with recovering the transported label.

16.7 CP7: a research specification with auditable novelty

A focused extension of the implemented checker is a certified branch-tracking engine for a bounded polynomial family with parameter boxes. It should combine exact or interval Bernstein exclusion, isolating root clusters, monodromy permutations, composition-domain checks and a compact replayable certificate. Its theorem must say precisely which accepted outputs are sound, whether the search terminates on separated inputs, and how failure is classified near actual branch values.

A possible new result would be an improved uniform complexity bound, a smaller independently checked certificate, or a demonstrated capability on a family that existing certified methods cannot handle under the same assumptions. Merely adding a monodromy plot, using a proof assistant for integer arithmetic, or moving a solver to a GPU does not establish that contribution. The proposed experiments must include the axis-witness counterexample CP2, contour boundary roots, multiple roots, branch-label swaps and parameter cells with genuine topology.

16.8 Research extension: completeness, lower bounds and the audit itself

The source section supplies the topological and analytic reasons that a certificate can be sound. The computational research question asks how much of the admissible input space a particular finite witness language can actually certify, and at what bit cost. Soundness, termination, completeness relative to a representation, and practical performance are different properties.

16.8.1 CP8: distinguish three quantifiers in a family certificate

For a family p_λ and contour Γ, the assertions

$$(\forall\lambda)(\forall z \in \Gamma)\ p_\lambda(z) \neq 0, \quad (\exists n)(\forall\lambda, z)\ n \cdot p_\lambda(z) > 0, \quad (\forall \text{ arcs } I)(\exists n_I)(\forall\lambda, z \in I)\ n_I \cdot p_\lambda(z) > 0$$

are not equivalent. The second implies the first but excludes nonzero loops of nonzero winding. The third permits different half-planes along the contour and can certify such loops. Even the third can fail for a parameter relaxation whose joint value hull contains zero.

Rotated-language improvement. The constant family $1 - \lambda + i\lambda$ separates axis-language incompleteness from genuine nonvanishing failure. Its arbitrarily fine source partitions do not change the family values. A rational rotated witness succeeds immediately. This is a clean negative control because degree, arithmetic precision and source curvature cannot explain the old failure.

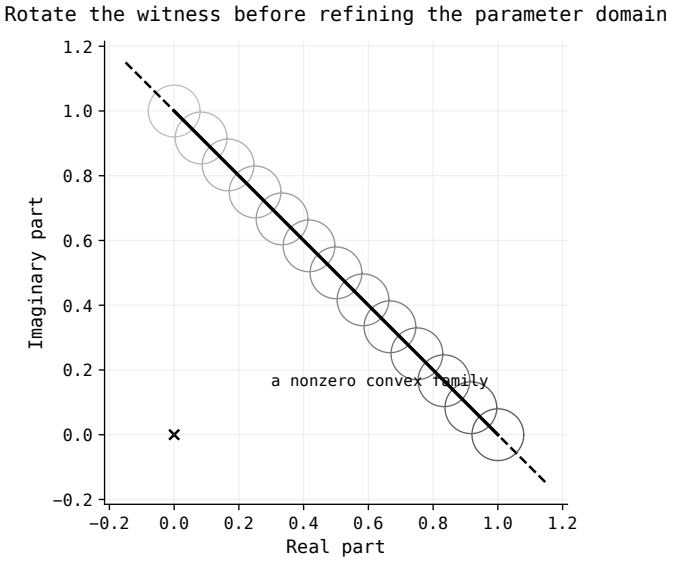


Figure 16.2: A witness-language obstruction has a different repair from numerical precision. Rotation solves this declared convex-family example; it is not a universal substitute for parameter subdivision.

Exercise 16.7 (CP8a)**.** Give a nonconvex family of nonzero constants whose convex relaxation contains zero. What would an attempted whole-relaxation certificate mean?

Solution. The two constants $+1$ and -1 are individually nonzero, while their convex hull contains zero. A certificate for the full convex relaxation would falsely claim that the zero constant is excluded. The safe choices are to keep two separate parameter components or use a representation that preserves nonconvexity. No sound half-plane checker should accept the full relaxation.

16.8.2 CP9: completeness of the closest-hull normal oracle

C12 gives a complete finite separation oracle for a rational polygonal hull: enumerate its vertices and orthogonal edge projections, test each rational candidate against all controls, and accept one with a strict positive margin. If zero is outside the hull, one candidate works. If zero is inside, no candidate can work. This is a completeness statement about the finite hull representation, not about all curves represented by those controls.

A proof boundary worth formalizing. The oracle theorem depends on convex hull correctness, the closest-point characterization and rational arithmetic. The winding checker additionally depends on partition coverage and homotopy. The source's argument principle additionally depends on the continuous map and its interior. A formalization that checks only the arithmetic sign of one sample does not close these layers. The v2 obligation ledger names them separately.

Practical search cost. Building the hull of M controls takes $O(M \log M)$ orientation comparisons in the standard monotone-chain implementation, and testing all edge candidates directly can take $O(M^2)$ dot products. That is not an optimal support-function implementation. On easy axis-separable arcs, four coordinate tests may be faster. The capability improvement therefore coexists with a possible generation-time penalty.

16.8.3 CP10: an independently checkable lower-bound portfolio

The precision model MP8 is deliberately small enough to support three proofs: convexity, dynamic programming and enumeration. Its dynamic program records $F_j(B)$, the minimum scaled error after j stages with total bit cost B:

$$F_{j+1}(B) = \min_{b \in \{2,\dots,14\}} \{F_j(B-b) + w_{j+1} 2^{14-b}\}.$$

The weights here are the scaled integers $1, 4, 16, 64$; the common scale is $64 \cdot 2^{14}$. The base is $F_0(0) = 0$ and $F_0(B) = +\infty$ otherwise. Induction on stages proves that the recurrence considers every admissible allocation exactly through its last coordinate, with no hidden greedy assumption.

Why this helps the corpus audit. It is an example of a legitimate maximal/minimal claim: a declared objective, a feasible candidate, an independently checked lower bound and a proof that the search domain is sufficient. The same standard is applied to document metrics. A count reaching its declared denominator proves saturation of that count, not absolute semantic completeness or learning efficacy.

Exercise 16.8 (CP10a)**.** A purported optimality certificate contains only a feasible vector. Which side of the optimum does it bound, and what is missing?

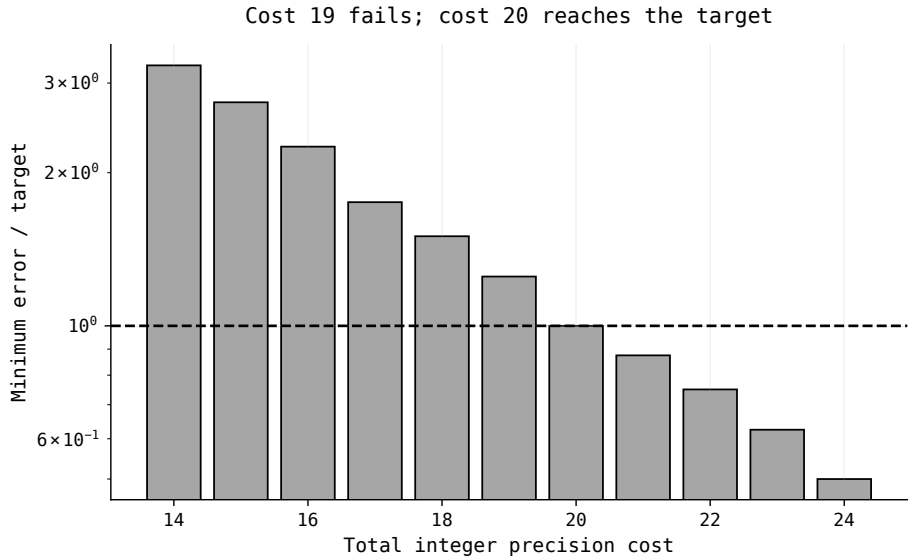


Figure 16.3: The exact minimum error at costs near the target. The first feasible cost is twenty, independently matching the analytic lower bound in MP8.

Solution. In a minimization problem, a feasible candidate gives an upper bound. It does not exclude a cheaper feasible candidate. A matching lower bound, an exhaustive justified search, a dual certificate or another valid impossibility argument is missing. Renaming the vector "optimal" does not add that argument.

16.8.4 CP11: regular-closure state compression needs an invariant

For degree d with monodromy group G, the regular closure has $|G|$ states. An observable invariant under a stabilizer can descend to fewer states. The quotient must preserve every operation used by that observable; otherwise the compressed transition system may produce the right count but the wrong label or continuation path.

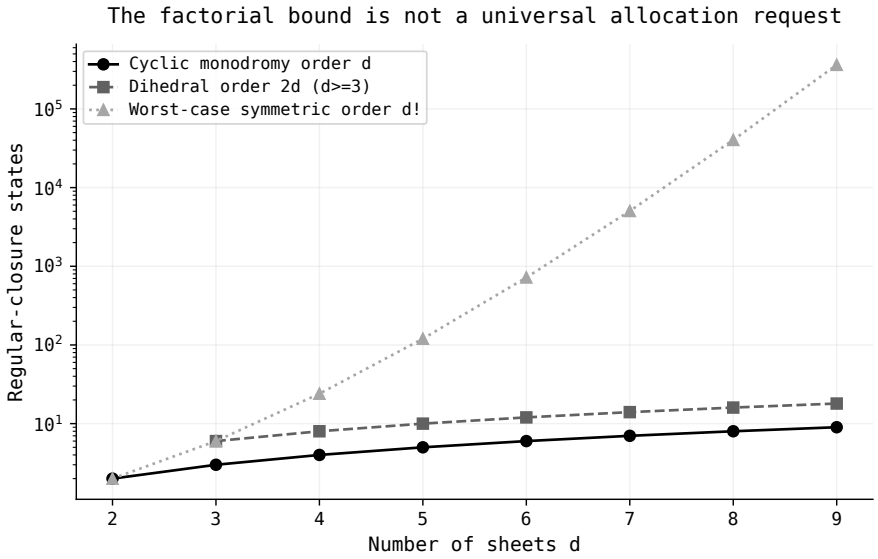


Figure 16.4: Cyclic, dihedral and full-symmetric possibilities. The dihedral line is interpreted only for degrees at least three; the factorial curve is a worst-case group order, not a mandatory storage allocation.

Comparison protocol. Report the actual generated subgroup, transitivity, kernel/stabilizer, quotient map and preserved observables. Benchmark full-ordering and quotient-state algorithms only on a task that genuinely factors through the quotient. A six-state root-ordering automaton and a three-state single-root automaton do not produce interchangeable outputs even when both descend from the same cover.

16.8.5 CP12: attack the corpus audit as an algorithm

The release audit has negative tests of its own. Removing a required witness must reduce coverage or fail validation. Duplicating a claim identifier must not create an additional theorem. Reusing the same figure in five documents must not create five original figures. A broken prerequisite/content pairing must fail the lecture gate. A stale metric from an earlier PDF must fail its file-hash binding.

These tests are necessary because a maximality metric can be gamed as easily as a software benchmark. Source obligations are frozen in a versioned list before the report is counted; pedagogical repetitions are labelled as navigational; prose-duplication diagnostics exclude title pages, source registries and intentional theorem references. The semantic correctness of each witness remains an expert-review obligation beyond these syntactic tests.

16.8.6 CP13: research acceptance is a conjunction, not a weighted slogan

A credible new algorithmic result needs mathematical soundness, a meaningful admissible class, honest failure reporting, reproducible implementation and a comparison against an adequate baseline. A huge speed ratio on one adversarial example cannot compensate for an invalid certificate. Conversely, a slower checker may be useful because it certifies a larger class or gives a stronger guarantee. The proposed evaluation uses a vector of outcomes and reports incomparability instead of manufacturing one prestige score.

Chapter 17

Analytic approximation and geometric computation

17.1 C01. Finite jets as a certified data structure

Let $f(z) = \sum_{j\geq 0} a_j z^j$ be holomorphic on a neighbourhood of $\overline{\mathbb{D}_R}$ and satisfy $|f| \leq M$ there. A degree-n jet is the residue class of its Taylor series in $\mathbb{C}[z]/(z^{n+1})$. The numerical representation stores $(a_0, \ldots, a_n)$ together with (R, M). Omitting the final two fields discards the approximation guarantee.

For $r < R$, Cauchy's estimate yields

$$\sup_{|z|\leq r} \left| f(z) - \sum_{j=0}^{n} a_j z^j \right| \leq M \frac{(r/R)^{n+1}}{1 - r/R}. \tag{17.1}$$

The proof is the geometric sum of the coefficient bounds $|a_j| \leq M/R^j$. If the stored coefficients have absolute errors η_j, the evaluation certificate becomes

$$E(r) \leq M \frac{(r/R)^{n+1}}{1 - r/R} + \sum_{j=0}^{n} \eta_j r^j + E_{\text{arithmetic}}.$$

The first term is analytic truncation; the second is data error; the third depends on the evaluation implementation. Their aggregation is valid by the triangle inequality. A small polynomial residual does not supply the missing analytic-radius bound.

17.1.1 Algebraic operations and truncation semantics

Addition is coordinatewise. Multiplication is truncated convolution:

$$(ab)_k = \sum_{j=0}^{k} a_j b_{k-j}, \qquad 0 \leq k \leq n.$$

This is exact in the quotient algebra. For inversion, $a_0 \neq 0$ is necessary and sufficient for invertibility in the quotient, with

$$b_0 = a_0^{-1}, \qquad b_k = -a_0^{-1} \sum_{j=1}^{k} a_j b_{k-j}.$$

A quotient inverse exists even when the represented analytic function has zeros elsewhere in the proposed evaluation domain. A domain-level inverse certificate must exclude those zeros. This distinction is a source of useful adversarial tests for series libraries.

Exercise 17.1. For $R = M = 1$, determine the smallest degree giving uniform truncation error at most 2^{-10} on $\overline{\mathbb{D}_{1/2}}$ under (17.1).

Solution. The bound is 2^{-n}. Hence $n = 10$ is sufficient and is the smallest degree allowed by this bound. Sharpness for the entire bounded-holomorphic class is not established by this calculation. □

Graphical reference. The shared construction is displayed in Volume I.

17.2 C02. Exact minimax complexity of linear restriction

Mathematical reference. Volume I, normal-family and mapping core M34–M35 This is an additional application of Hilbert-space approximation theory.

On the Hardy space $H^2(\mathbb{D})$, let $T_r f(z) = f(rz)$, $0 < r < 1$. The coefficient basis identifies T_r with $\operatorname{diag}(1, r, r^2, \ldots)$. The exact optimal error among bounded complex-linear operators of rank at most N is

$$\inf_{\operatorname{rank} A \le N} \|T_r - A\|_{H^2 \to H^2} = r^N. \tag{17.2}$$

The upper bound is attained by retaining the first N coefficients. For the lower bound, restrict an arbitrary rank-N operator to the $(N+1)$-dimensional space spanned by $1, z, \ldots, z^N$. Its kernel contains a unit vector v. Consequently $\|(T_r - A)v\| = \|T_r v\| \ge r^N$.

This proof covers adaptive choices of the rank-N linear operator, arbitrary linear measurements, and arbitrary linear reconstructions within the specified operator class. It does not cover unrestricted nonlinear encodings, randomized average-case objectives, or a prior distribution supported on a smaller class. The extremizing adversarial vector may depend on the proposed operator.

For tolerance $0 < \varepsilon < 1$, the smallest admissible rank is

$$N_{\min} = \left\lceil \frac{\log \varepsilon}{\log r} \right\rceil.$$

At $r = 1/2$, error 2^{-12} requires exactly twelve complex coefficients. An arbitrary thirteen-dimensional unresolved subspace would change the problem; the relevant finite-dimensional obstruction is already contained in the first thirteen monomials.

17.2.1 Storage accounting

Equation (17.2) counts complex coordinates of an exact linear representation. A machine implementation additionally chooses coefficient quantization. With N stored coefficients perturbed by errors e_j, the restriction error has the exact squared form $\sum_{j<N} r^{2j}|e_j|^2$. Uniform coordinate quantization can therefore be compared with weighted precision allocation without conflating coefficient count and bit count.

Exercise 17.2. At $r = 3/4$, give a proved rank requirement for error 10^{-3}.

Solution. $\log(10^{-3})/\log(3/4) \approx 24.012$, so $N = 25$. The adjacent exact tests are $(3/4)^{24} > 10^{-3}$ and $(3/4)^{25} < 10^{-3}$; clearing positive denominators reduces both to integer comparisons. □

Graphical reference. The shared construction is displayed in Volume I.

17.3 C03. Coefficient extraction and optimal sampling radius

Suppose the analytic radius is normalized to one, $|a_j| \le M$, and $N > n$. At nodes $z_k = qe^{2\pi ik/N}$ define

$$\widehat{a}_n = \frac{1}{Nq^n} \sum_{k=0}^{N-1} f(z_k) e^{-2\pi ink/N}.$$

Absolute convergence permits interchange of sums. Root-of-unity orthogonality gives

$$\widehat{a}_n = a_n + \sum_{\ell \ge 1} a_{n+\ell N} q^{\ell N}.$$

If each sample has absolute error at most η, then

$$|\widehat{a}_n - a_n| \le B(q) := \eta q^{-n} + M \frac{q^N}{1-q^N}, \qquad 0 < q < 1. \tag{17.3}$$

Arithmetic in the discrete transform is excluded from this display and must be bounded separately.

17.3.1 Existence and uniqueness of the optimal radius

Write $t = -\log q > 0$. Then

$$B(e^{-t}) = \eta e^{nt} + M \sum_{\ell \ge 1} e^{-\ell N t}.$$

For $n \ge 1$, $\eta, M > 0$, termwise differentiation on compact subintervals of $(0, \infty)$ gives

$$\frac{d^2}{dt^2} B(e^{-t}) = n^2 \eta e^{nt} + MN^2 \sum_{\ell \ge 1} \ell^2 e^{-\ell N t} > 0.$$

The function tends to infinity at both ends. It therefore has exactly one global minimizer. The stationary condition in q is

$$n\eta q^{-n-1} = \frac{MNq^{N-1}}{(1-q^N)^2}.$$

This theorem justifies bracketing a single root of the derivative. A local numerical optimizer has no additional global-minimum obligation for this model once the strict-convexity argument is accepted.

For $(n, N, M, \eta) = (8, 64, 2, 10^{-12})$, the stationary equation reduces, after multiplication by a positive factor, to

$$P(q) = 128q^{72} - 8 \cdot 10^{-12}(1-q^{64})^2 = 0.$$

The rational endpoints $q_- = 6555554687/10^{10}$ and $q_+ = 6555554688/10^{10}$ satisfy $P(q_-) < 0 < P(q_+)$. Strict convexity proves that the unique optimum lies in this interval of width 10^{-10}. Exact arithmetic also gives

$$3.2981998264 \cdot 10^{-11} < \min B < 3.2981998301 \cdot 10^{-11},$$

with the displayed decimals rounded outwards. The endpoint signs and the underlying rational error bounds are stored in `results/optimal_radius_certificate.json`. The floating-point optimization is retained as an independent diagnostic.

Exercise 17.3. Why can a smaller sample radius increase the error?

Solution. The sample-error contribution is ηq^{-n}. For $n > 0$ it diverges as $q \downarrow 0$, even while the aliasing contribution tends to zero. The quantity being reconstructed is a coefficient, which includes division by q^n. □

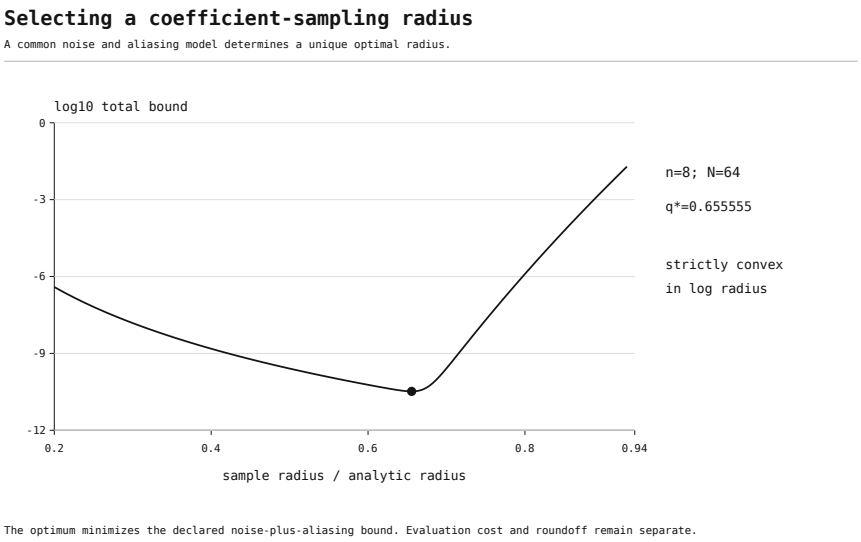


Figure 17.1: The unique optimum balances two explicitly derived terms. The graph records a numerical evaluation of the proved scalar objective.

17.4 C04. Finite sampling, regularity witnesses, and certificates

A boundary residual $g(\theta)$ sampled at N equally spaced angles admits the certificate

$$\sup_{\theta} |g(\theta)| \le \max_{k} |g(2\pi k/N)| + \frac{\pi L}{N} \tag{17.4}$$

provided $|g(\theta)-g(\vartheta)| \le L d_{\mathbb{T}}(\theta,\vartheta)$ in the shortest angular distance. Every angle lies within π/N of a node; the triangle inequality proves (17.4). For $g(\theta) = h(Re^{i\theta})$ the bound $L \le R\sup_{|z|=R}|h'(z)|$ follows by differentiating the parametrization.

There is an exact obstruction without a regularity bound. The nonzero polynomial $h(z) = \delta(z^N - r^N)$ vanishes at every node $re^{2\pi ik/N}$. Its inter-sample magnitude reaches $2|\delta|r^N$. Scaling δ makes that magnitude arbitrarily large while preserving every sampled value. This argument gives a constructive negative test for a purported sample-only supremum checker.

A sound checker receives the nodes, enclosed sample values, the angular Lipschitz bound, and the target tolerance. It validates the rational inequality obtained from (17.4) with $\pi < 22/7$. The analytic proof of L is a separate input obligation. The local SMT calculation validates the arithmetic implication; it does not infer differentiability or certify the supplied derivative bound.

Exercise 17.4. With sample maximum 10^{-3} and $L = 1/5$, give a sufficient sample count for boundary error at most $1/100$.

Solution. Using $\pi < 22/7$ requires $1/1000 + 22/(35N) \le 1/100$, equivalently $N \ge 4400/63$. Thus $N = 70$ suffices for this conservative rational certificate. □

17.5 C05. Certified conformal charts in computational geometry

Consider $\phi(z) = z + az^2$ on a convex disk $\overline{\mathbb{D}_r}$, with $2|a|r < 1$. The difference identity

$$\phi(z) - \phi(w) = (z-w)(1 + a(z+w))$$

gives the global estimate

$$(1 - 2|a|r)|z-w| \le |\phi(z) - \phi(w)| \le (1 + 2|a|r)|z-w|.$$

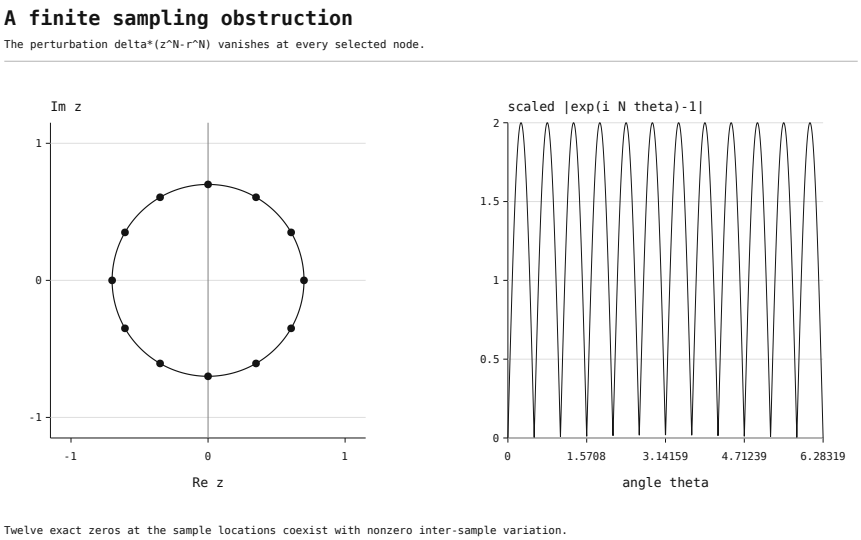


Figure 17.2: An exact finite-sampling adversary. The polynomial vanishes at every displayed sample while remaining nonzero between samples.

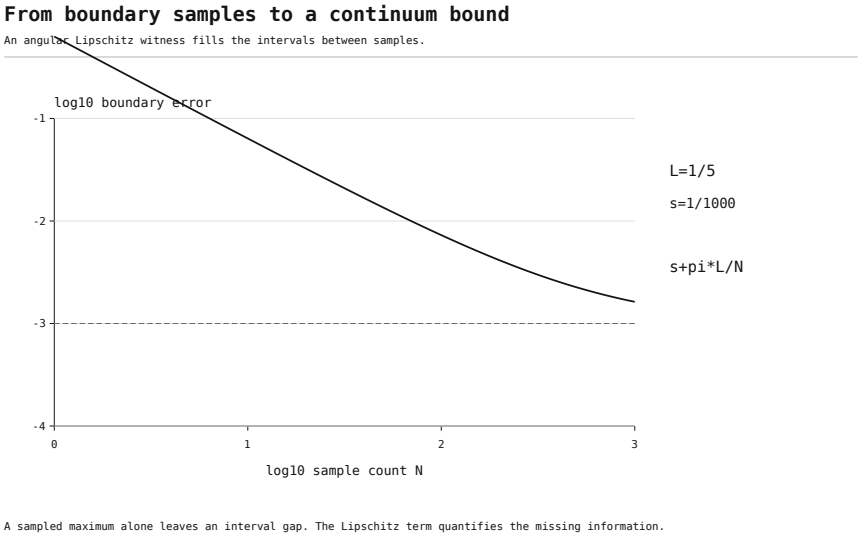


Figure 17.3: A verified regularity bound converts a finite sample maximum into a continuum estimate.

It proves injectivity, an inverse Lipschitz bound, and a derivative bound with the same algebraic witness. Checking the derivative at grid nodes supplies none of these global conclusions by itself.

Suppose a holomorphic approximation p satisfies $\sup_{\mathbb{D}_R} |p - \phi| \le \varepsilon$ for $r < R$. Cauchy's estimate yields

$$\sup_{\mathbb{D}_r} |p' - \phi'| \le \delta := \frac{\varepsilon}{R - r}.$$

Integrating the derivative error along the segment joining $z, w \in \mathbb{D}_r$ gives

$$|p(z) - p(w)| \ge (1 - 2|a|r - \delta)|z - w|.$$

The strict positivity of the final factor is an explicit sufficient condition for global injectivity of the approximation. It depends on convexity of the comparison domain. Replacing the segment by an arbitrary path requires a path-length bound.

For $a = 1/5$, $r = 3/5$, $R = 4/5$, $\varepsilon = 1/100$, the certified lower factor is $71/100$. The corresponding inverse Lipschitz constant is at most $100/71$. The universal normalized Koebe

derivative quotient at $r = 3/5$ is 256, whereas the exact quotient in this family is 31/19. Their ratio is 4864/31. These are enclosure constants, with no measured runtime interpretation.

Exercise 17.5. Can a positive Jacobian at every point guarantee global injectivity on every domain?

Solution. No. The exponential map has nonzero complex derivative everywhere but is periodic. The present global conclusion uses the difference inequality on the entire comparison domain. □

Graphical reference. The shared construction is displayed in Volume I.

17.6 C06. Inverse formulas and projective numerical contracts

For $\phi_a(z) = z + az^2$, the inverse branch normalized at zero is

$$z = \frac{\sqrt{1 + 4aw} - 1}{2a} = \frac{2w}{1 + \sqrt{1 + 4aw}}.$$

The second form also has a continuous interpretation at $a = 0$. The equality follows by multiplying numerator and denominator by $1 + \sqrt{1 + 4aw}$. The square-root branch is part of the contract: it is the branch equal to 1 at $w = 0$ on a simply connected zero-free discriminant region. A pointwise principal-square-root instruction is sufficient only when the entire chosen region respects that branch convention.

Subtracting 1 from a square root close to 1 loses relative accuracy in the first formula. The rationalized expression has a denominator close to 2. The included binary64 graph demonstrates the effect for positive real w and $a = 1/5$; the algebraic identity supplies the representation-level guarantee. A machine-level error theorem would additionally specify correctly rounded elementary operations and square-root implementation.

For meromorphic values, homogeneous coordinates $[u : v] \in \mathbb{P}^1(\mathbb{C})$ support the chordal comparison

$$\chi([u : v], [p : q]) = \frac{|uq - pv|}{\sqrt{|u|^2 + |v|^2}\sqrt{|p|^2 + |q|^2}}.$$

The formula is invariant under independent nonzero rescalings of the coordinate pairs. It permits a finite representation of infinity as $[1 : 0]$. Rescaling numerical pairs before taking products controls overflow; the resulting rounded arithmetic still needs an error analysis. A nonzero pair is a mandatory input invariant.

Exercise 17.6. What is the chordal error between $[n : 1]$ and $[1 : 0]$?

Solution. It is $1/\sqrt{1 + n^2}$ for real n. Euclidean distance to an infinite target is undefined, while this projective error tends to zero. □

17.7 C07. Continuation, identifiability, and attainable guarantees

On the class of polynomials, observation on $\overline{\mathbb{D}_r}$ determines every coefficient uniquely in exact arithmetic. Nevertheless, for $0 < r < R$, the monomials z^n satisfy

$$\frac{\|z^n\|_{\infty,\mathbb{D}_R}}{\|z^n\|_{\infty,\mathbb{D}_r}} = (R/r)^n.$$

There can be no uniform Lipschitz continuation bound over all polynomial degrees. A norm bound on a larger analytic domain, a finite-dimensional prior, or another regularity constraint changes the admissible class and can restore conditional stability.

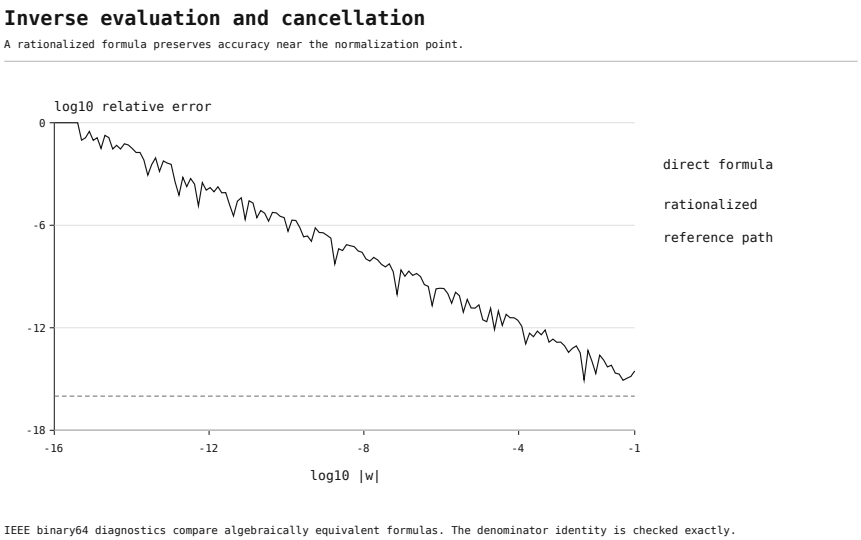


Figure 17.4: Cancellation in the direct inverse expression is visible across a range of input scales. This is a floating-point diagnostic.

A two-input indistinguishability argument is especially useful for software specifications. If two admissible functions differ by at most 2η in the observation norm, adversarial errors of size η can make their data identical. Any deterministic reconstruction must then have output error at least half their separation for one of the inputs. Taking appropriately scaled monomials produces explicit witnesses. This lower bound concerns information in the observations; an improved evaluation algorithm cannot recover information absent from those observations.

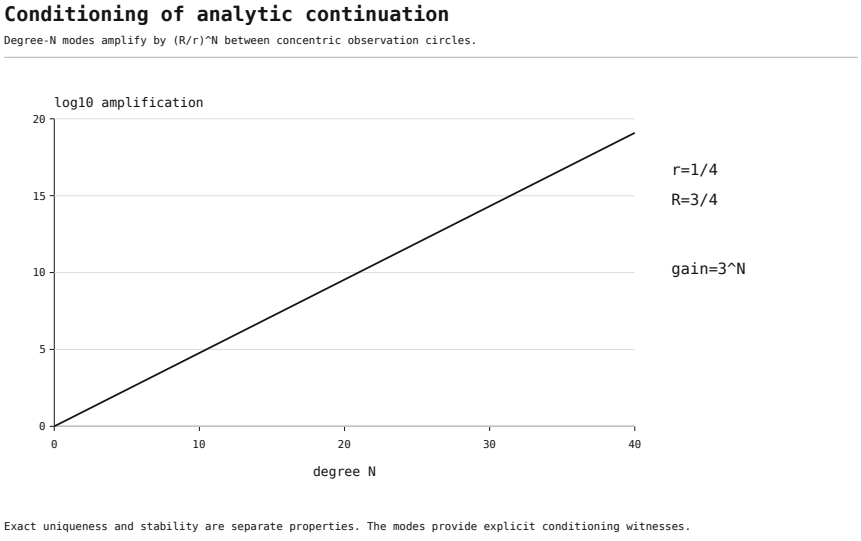


Figure 17.5: A constructive lower-bound family for continuation sensitivity. The vertical scale is logarithmic.

17.7.1 A complete verification record for this application family

Each reportable result includes the mathematical statement, domain assumptions, representation format, computation transcript, and discrepancy criterion. The code distinguishes exact rational identities, symbolic normalization, real-closed-field solver outcomes, and floating-point diagnostics. The prospective formalization requires a pinned project and an explicit theorem-coverage table.

Full Lean kernel verification of this section is incomplete.

Exercise 17.7. At $r = 1/4$, $R = 3/4$, what amplification is witnessed by degree twenty?

Solution. It is $3^{20} = 3\,486\,784\,401$. This exact ratio quantifies one mode; it is not a condition number estimated from a fitted data matrix. □

Chapter 18

Coefficient and chart calculations

18.1 CU01. Horner evaluation with a remainder certificate

Mathematical reference. Volume II, analytic-approximation chapter C01; Volume I, normal-family and mapping core M03

18.1.1 Prerequisites and governing statement

A polynomial can be evaluated by nested multiplication. The coefficient table and the function it approximates are distinct inputs.

$$p_N(z) = a_0 + z(a_1 + z(a_2 + \cdots + za_N))$$

Horner evaluation uses N multiplications and N additions after the coefficients are available.

$$E(r) \le M\frac{(r/R)^{N+1}}{1 - r/R} + \sum_{j=0}^{N} \eta_j r^j + E_{\text{arithmetic}}.$$

18.1.2 Derivation and worked instance

The displayed inequality separates three causes of error. The first comes from omitted analytic coefficients; the second comes from inaccuracies in stored coefficients; the third comes from arithmetic in the evaluation. Their sum follows from repeated triangle inequalities. Horner evaluation improves the arithmetic count relative to computing each power independently, but its stability still depends on coefficient magnitudes and cancellation.

For exact coefficients and R=M=1, r=1/2, degree ten gives analytic error at most 1/1024. Eleven complex coefficients are stored, including the constant coefficient. The evaluation still uses only ten nested multiplications.

18.1.3 Comparison and validity conditions

Baseline. Evaluate a polynomial and report its value without an approximation domain.

Proposed procedure. Attach an analytic tail certificate and a coefficient-error budget to the same evaluation.

Quantitative conclusion. The operation count remains explicit while the output acquires a uniform analytic error bound.

Validity conditions. A tail estimate obtained for one radius cannot be reused at a larger radius without recomputation.

Exercise 18.1. How many coefficients and Horner multiplications does degree seventeen require?

Solution. Eighteen coefficients and seventeen multiplications. An additional norm or remainder certificate is stored as metadata. □

Graphical reference. The shared construction is displayed in Volume I.

18.2 CU02. Weighted coefficient tables

Mathematical reference. Volume II, analytic-approximation chapter C02; Volume I, normal-family and mapping core M34–M35.

18.2.1 Prerequisites and governing statement

The squared Euclidean norm of a coefficient vector is the sum of squared magnitudes. Analytic restriction multiplies its j-th coordinate by r^j.

$$(a_0, a_1, a_2, \ldots) \mapsto (a_0, ra_1, r^2a_2, \ldots)$$

For r=1/2 the weights are 1,1/2,1/4,1/8 and continue geometrically.

$$\|T_r - T_r P_N\| = r^N.$$

18.2.2 Derivation and worked instance

The projection P_N retains indices zero through N-1. The first discarded coefficient has weight r^N and every subsequent weight is smaller. A unit vector concentrated at that first discarded coordinate attains the error. This establishes an exact bound for the coordinate truncation. The general rank-nullity argument in the graduate branch proves that no rank-N linear competitor improves the worst case.

At r=1/2 and N=8, every unit-norm input is approximated with error at most 1/256. The adversarial input z^8 attains that value.

18.2.3 Comparison and validity conditions

Baseline. Choose a storage length from a plot of one observed coefficient vector.

Proposed procedure. Use the exact worst-case restriction bound on a declared unit ball.

Quantitative conclusion. Eight retained coordinates give an attained worst-case error of 2^{-8}.

Validity conditions. The norm is an H^2 coefficient norm. It is not automatically a pointwise maximum or a physical peak-amplitude bound.

Exercise 18.2. For r=1/4 and N=5, compute the exact bound.

Solution. The bound is $(1/4)^5$=1/1024. □

18.3 CU03. Root-of-unity averaging and aliasing

Mathematical reference. Volume II, analytic-approximation chapter C03

18.3.1 Prerequisites and governing statement

The sum of N-th roots of unity vanishes except when the exponent is divisible by N.

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi i j k/N} = \begin{cases} 1 & N \mid j, \\ 0 & N \nmid j. \end{cases}$$

For N=4, the roots are 1,i,-1,-i.

$$\widehat{a}_n = a_n + \sum_{\ell \geq 1} a_{n+\ell N} q^{\ell N}.$$

18.3.2 Derivation and worked instance

Expand the analytic function at every sample node and exchange the absolutely convergent sums. The root-of-unity identity cancels all powers except those congruent to n modulo N. The surviving high-order powers are the aliasing error. A Fourier transform computes the same linear combinations efficiently; the analytic error estimate is independent of the particular transform implementation.

For f(z)=1/(1-z), every Taylor coefficient is one. With N=4 and q=1/2, the exact-arithmetic estimate is $1/(1\text{-}q^4)=16/15$ for each n<4. The coefficient error is 1/15.

18.3.3 Comparison and validity conditions

Baseline. Compute each Taylor coefficient by a separate numerical derivative.

Proposed procedure. Use one circular sample set for a discrete Fourier coefficient calculation with a derived aliasing term.

Quantitative conclusion. One set of N values supplies N coefficient combinations with an explicit aliasing identity.

Validity conditions. The sample circle must avoid poles and lie inside the analytic radius. Increasing N does not repair a violated domain assumption.

Exercise 18.3. For the same function with N=8 and q=1/2, compute the aliasing error.

Solution. The error is $q^8/(1\text{-}q^8)=1/255$. □

Graphical reference. See Figure 17.1 for the shared construction.

18.4 CU04. A test suite can miss an entire polynomial

18.4.1 Prerequisites and governing statement

A polynomial may vanish at a finite set of prescribed nodes without vanishing identically.

$$p(z) = \delta(z^N - r^N)$$

At z=r exp(2pi i k/N), the N-th power is r^N.

$$p(re^{2\pi i k/N}) = 0, \qquad \max_{|z|=r} |p(z)| = 2|\delta| r^N.$$

18.4.2 Derivation and worked instance

The sampled values are exactly zero for every integer k. At the halfway angle π/N, the N-th power equals -r^N, attaining the displayed maximum. Since δ is arbitrary, there is no universal supremum bound determined by these zero samples alone. A polynomial degree bound below N or an independent derivative bound changes the information available to the checker.

For N=12,r=1 and δ=3, twelve tests all return zero, but the true maximum is six. This is an exact algebraic counterexample, unaffected by numerical roundoff.

18.4.3 Comparison and validity conditions

Baseline. Accept a claimed zero function after it passes a fixed sample grid.

Proposed procedure. Construct a finite-sampling adversary and require a continuum regularity witness.

Quantitative conclusion. The counterexample demonstrates an unbounded sample-only error gap.

Validity conditions. Randomized testing can improve detection probabilities for a specified adversarial model, but finite success still differs from a universal analytic proof.

Exercise 18.4. What changes if the polynomial degree is known to be below N?

Solution. A polynomial of degree at most N-1 with N distinct zeros is identically zero. The degree bound is an essential added assumption. □

18.5 CU05. A global map invariant from one factorization

Mathematical reference. Volume II, analytic-approximation chapter C05; Volume I, normal-family and mapping core M31.

18.5.1 Prerequisites and governing statement

A lower distance bound with a positive constant proves injectivity. The triangle inequality bounds a perturbation of one.

$$\phi(z) - \phi(w) = (z - w)(1 + a(z + w))$$

If |z|,|w|≤r then |a(z+w)|≤2|a|r.

$$|\phi(z) - \phi(w)| \geq (1 - 2|a|r)|z - w|.$$

18.5.2 Derivation and worked instance

The reverse triangle inequality gives |1+a(z+w)|≥1-2|a|r. Positivity of this factor excludes equal images of distinct points. The same calculation supplies an inverse Lipschitz constant, which quantifies the propagation of target-coordinate error back to source coordinates. No mesh density enters the theorem.

For a=1/5 and r=1, the global lower factor is 3/5. A target error of at most 3/100 therefore produces source error at most 1/20 within the image.

18.5.3 Comparison and validity conditions

Baseline. Look for crossings in a drawn transformed grid.

Proposed procedure. Prove a difference-factor bound on the full comparison domain.

Quantitative conclusion. A single symbolic invariant covers infinitely many input pairs.

Validity conditions. The conclusion uses the quadratic model exactly. A fitted model with coefficient uncertainty must include that uncertainty in the bound.

Exercise 18.5. With |a|≤1/4 and r=1, give the certified inverse Lipschitz constant.

Solution. The lower factor is at least 1-1/2=1/2. Its reciprocal is two. □

Graphical reference. The shared construction is displayed in Volume I.

18.6 CU06. Projective representations in a numerical interface

Mathematical reference. Volume II, analytic-approximation chapter C06; Volume I, normal-family and mapping core M09.

18.6.1 Prerequisites and governing statement

A projective point [u:v] is unchanged by multiplying both coordinates by a nonzero scalar.

$$[u : v] = [\lambda u : \lambda v] \quad (\lambda \neq 0)$$

[1:0] represents ∞ and [0:1] represents zero.

$$\chi([u : v], [p : q]) = \frac{|uq - pv|}{\sqrt{|u|^2 + |v|^2}\sqrt{|p|^2 + |q|^2}}.$$

18.6.2 Derivation and worked instance

The numerator gains the product of the two scale magnitudes under independent rescaling; the denominator gains the same product. The ratio is therefore well defined. Before numerical products, coordinate pairs can be normalized to avoid unnecessary overflow. A representation invariant must exclude the zero pair and record the effect of rounding on the determinant.

The pairs [1000:1] and [1:0.001] describe the same value. Their chordal distance is exactly zero in real arithmetic.

18.6.3 Comparison and validity conditions

Baseline. Represent an infinite value by overflow in a floating-point affine coordinate.

Proposed procedure. Store a nonzero homogeneous pair and compare values with a scale-invariant metric.

Quantitative conclusion. Infinity becomes a finite algebraic state with an explicit error metric.

Validity conditions. Approximate normalization is a numerical operation. Scale invariance of the exact formula does not prove a floating-point equality test is reliable.

Exercise 18.6. What validation must precede conversion from [u:v] to u/v?

Solution. The pair must be nonzero. A finite affine conversion additionally requires v nonzero and a suitable bound if division error is to be controlled. □

18.7 CU07. Cancellation-free inverse evaluation

Mathematical reference. Volume II, analytic-approximation chapter C06; Volume I, normal-family and mapping core M31.

18.7.1 Prerequisites and governing statement

Expressions that are algebraically equal can have different floating-point cancellation patterns.

$$(s-1)(s+1) = s^2 - 1$$

When s is close to one, s-1 loses significant relative digits.

$$\frac{\sqrt{1+4aw}-1}{2a} = \frac{2w}{1+\sqrt{1+4aw}}.$$

18.7.2 Derivation and worked instance

Multiply the first numerator and denominator by sqrt(1+4aw)+1. The numerator becomes 4aw, which cancels the factor 2a. The rationalized formula has denominator near two at small w, avoiding the subtraction of nearly equal quantities. It also extends continuously to a=0, where it gives z=w.

For a=1/5 and positive w decreasing to machine precision, the direct formula can return zero while the rationalized expression stays proportional to w. The plot is an executed binary64 diagnostic; the branch identity is an exact symbolic result.

18.7.3 Comparison and validity conditions

Baseline. Evaluate the normalized quadratic root by subtracting one from a square root.

Proposed procedure. Rationalize the numerator and retain the chosen branch.

Quantitative conclusion. The improvement concerns numerical accuracy near the normalization point; no universal timing claim is made.

Validity conditions. A square-root branch crossing still invalidates the inverse selection. Rationalization repairs cancellation, not branch semantics.

Exercise 18.7. What value does the rationalized expression take at w=0?

Solution. The numerator is zero and the chosen square root equals one, so the value is zero. □

Graphical reference. See Figure 17.4 for the shared construction.

18.8 CU08. Cost and verification as separate outputs

Mathematical reference. 5; Volume II, analytic-approximation chapter C01–C07.

18.8.1 Prerequisites and governing statement

An algorithmic comparison requires equivalent inputs and equivalent requested outputs.

$$C_{\text{total}} = C_{\text{setup}} + mC_{\text{evaluation}} + C_{\text{verification}}$$

A one-time setup cost is amortized over m evaluations.

$$\text{gain} = (E_{\text{baseline}}/E_{\text{proposed}}) \quad \text{for the same norm and input class.}$$

18.8.2 Derivation and worked instance

A smaller enclosure is a valid improvement when both methods certify the same object. It is not itself a runtime ratio. A lower coordinate count is a storage improvement within its representation model; quantization bits and parsing overhead are additional costs. The experimental records preserve these distinctions and retain raw parameters for replay.

At r=3/5 the universal Koebe derivative quotient is 256 and the exact quadratic quotient is 31/19. Their enclosure ratio is 4864/31. A claim that the quadratic code runs 156.90 times faster would have no support from this calculation.

18.8.3 Comparison and validity conditions

Baseline. Compare a formula's length or one elapsed time with an unrelated method.

Proposed procedure. Record representation size, evaluation work, error guarantee, and setup cost under a common specification.

Quantitative conclusion. The comparison reports a proved enclosure improvement with its additional assumptions.

Validity conditions. The specialized bound requires verified membership in the quadratic family. The universal bound has a larger admissible class.

Exercise 18.8. Which extra datum would be needed to report a runtime improvement?

Solution. A matched executable benchmark with equivalent inputs, outputs, precision, hardware and timing protocol, including relevant setup and verification costs. □

Chapter 19

Approximation operators in computation

19.1 CG01. Optimality over arbitrary linear measurements

Mathematical reference. Volume I, normal-family and mapping core M35; Volume II, analytic-approximation chapter C02.

19.1.1 Prerequisites and governing statement

A bounded rank-N reconstruction factors through an N-dimensional range, regardless of its chosen coordinates.

$$\operatorname{rank} A \le N$$

The restriction to the first N+1 monomials has a nontrivial kernel.

$$\inf_{\operatorname{rank} A \le N} \|T_r - A\| = r^N.$$

19.1.2 Derivation and worked instance

The kernel argument does not assume diagonal structure of A. It therefore covers arbitrary measurements followed by linear reconstruction when the resulting operator is bounded and has rank at most N. Coordinate changes, dense mixing and a learned fixed linear encoder cannot improve the worst-case value under the same norm and input class. The conclusion is invariant under unitary reparameterizations of the Hilbert space.

For r=3/4 and N=25, the optimal error is exactly $(3/4)^{25}$. The integer comparison $1000*3^25<4^25$ verifies that it is below 10^{-3}.

19.1.3 Comparison and validity conditions

Baseline. Compare coefficient truncation with a particular basis transform.

Proposed procedure. Prove a lower bound covering every bounded complex-linear rank-N reconstruction.

Quantitative conclusion. The lower bound covers the complete stated linear class.

Validity conditions. Data-dependent nonlinear encoders need a separately specified information model.

Exercise 19.1. Why is rank alone insufficient without an input norm?

Solution. Unbounded scaling of the input would make any nonzero residual arbitrarily large. The unit ball defines the worst-case size being compared. □

Graphical reference. The shared construction is displayed in Volume I.

19.2 CG02. Strict convexity of the sampling-radius objective

Mathematical reference. Volume II, analytic-approximation chapter C03

19.2.1 Prerequisites and governing statement

A twice-differentiable strictly convex function has at most one stationary point. Divergence at both ends guarantees attainment of a minimum.

$$t = -\log q > 0$$

As q decreases to zero, t tends to ∞.

$$F(t) := B(e^{-t}) = \eta e^{nt} + M \sum_{\ell \geq 1} e^{-\ell N t}, \qquad F''(t) > 0.$$

19.2.2 Derivation and worked instance

Every term has a nonnegative second derivative and the noise term has a strictly positive second derivative for n≥1. Uniform convergence of the differentiated geometric series on compact t intervals justifies differentiation. The objective diverges as t tends to zero through aliasing and as t tends to ∞ through noise amplification. A root bracket for the first derivative therefore locates the unique global minimizer.

For n=8,N=64,M=2,$\eta=10^{-12}$, the numerical optimizer lies near q=0.6555554687. Exact rational signs of the cleared stationary polynomial can enclose this value without trusting a floating-point root finder.

19.2.3 Comparison and validity conditions

Baseline. Choose q=1/2 by convention or optimize a noisy sampled objective.

Proposed procedure. Derive an exact scalar objective whose global minimizer is unique.

Quantitative conclusion. The global optimum is certified by convexity once a stationary bracket is verified.

Validity conditions. The case n=0 has different endpoint behavior and is excluded from the stated strict interior-optimum argument.

Exercise 19.2. Write a denominator-free stationary equation.

Solution. MN q^{N+n}-n $\eta(1\text{-}q^N)^2$=0. All factors removed from the derivative equation are positive for q in (0,1). □

19.3 CG03. Derivative reconstruction from a larger-domain certificate

Mathematical reference. Volume I, normal-family and mapping core M01 and M32; Volume II, analytic-approximation chapter C05.

19.3.1 Prerequisites and governing statement

Cauchy estimates control an order-j derivative by a surrounding function bound and a positive radius.

$$\|e^{(j)}\|_{\mathbb{D}_r} \leq \frac{j!\,\|e\|_{\mathbb{D}_R}}{(R-r)^j}$$

The derivative order changes both the factorial and the power of the collar width.

$$E_j(r) \leq j!\varepsilon(R-r)^{-j}.$$

19.3.2 Derivation and worked instance

Every point in the smaller disk has a surrounding circle with radius approaching R-r inside the larger disk. Applying Cauchy and taking the radius limit gives the estimate. The same ε therefore supports several derivative certificates, each with its own growth as the collar narrows. A practical selection of R and r must account for the requested derivative order.

With ε=10^{-6} and R-r=1/10, the bounds for derivative orders one, two and three are 10^{-5},2*10^{-4},6*10^{-3}.

19.3.3 Comparison and validity conditions

Baseline. Differentiate an approximate polynomial and use its value error unchanged.

Proposed procedure. Propagate a whole-domain holomorphic error through a quantified Cauchy estimate.

Quantitative conclusion. An explicit derivative-error hierarchy replaces an unquantified differentiation step.

Validity conditions. The approximation error must be holomorphic on the larger disk. A piecewise interpolation with derivative jumps does not meet this hypothesis.

Exercise 19.3. Find the fourth-derivative bound for the same inputs.

Solution. 4!*10^{-6}/$(1/10)^4$=24/100=0.24. □

19.4 CG04. Exact sample certificates with uncertain data

Mathematical reference. Volume II, analytic-approximation chapter C04; source geometric bounds as certification inputs.

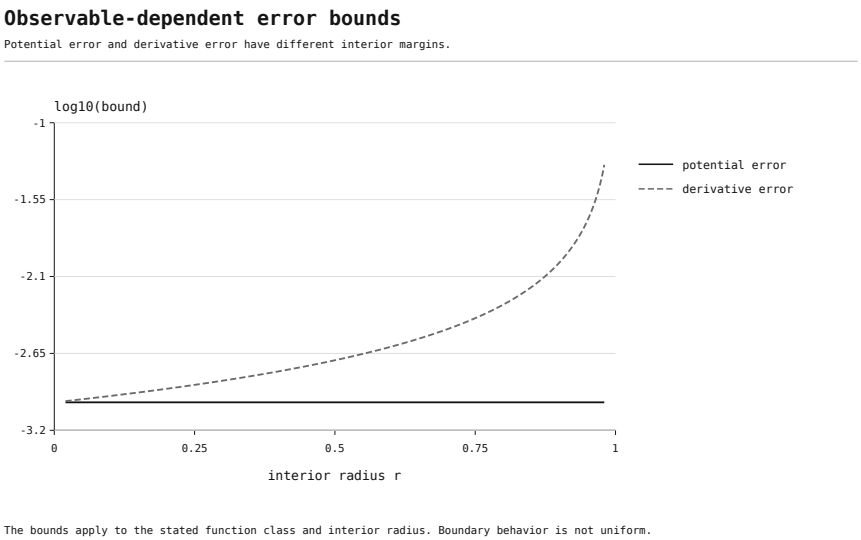


Figure 19.1: An explicit derivative-error hierarchy replaces an unquantified differentiation step. The figure is generated from the stated formula; its pixels have no independent proof status.

19.4.1 Prerequisites and governing statement

Interval enclosures bound every possible sample value. A Lipschitz constant covers the points between samples.

$$|g(\theta_k)| \le s_k, \qquad |g(\theta) - g(\vartheta)| \le L d_{\mathbb{T}}(\theta, \vartheta)$$

The worst mesh distance for N equally spaced samples is π/N.

$$\sup |g| \le \max_k s_k + \frac{22L}{7N}.$$

19.4.2 Derivation and worked instance

The geometric lemma uses the true π/N distance. Replacing π by $22/7$ preserves the upper-bound direction. If s_k and L are rational enclosures, the final comparison can be performed by clearing positive integer denominators. The checker needs no transcendental arithmetic. It nevertheless depends on a justified analytic bound for L.

For s=1/1000,L=1/5,N=70, the bound is 1/1000+11/1225=489/49000, which is below 1/100.

19.4.3 Comparison and validity conditions

Baseline. Round the largest sampled residual and declare a tolerance satisfied.

Proposed procedure. Check a rational upper bound that includes sample uncertainty and mesh distance.

Quantitative conclusion. A finite replayable arithmetic certificate closes the sample-to-continuum implication.

Validity conditions. Rational arithmetic prevents rounding in the final comparison but cannot validate a false regularity input.

Exercise 19.4. What exact margin remains below 1/100 in this example?

Solution. 1/100-489/49000=1/49000. □

19.5 CG05. Composition of approximate conformal charts

Mathematical reference. Volume I, normal-family and mapping core M32–M33; previous section telescope retained as a dependency.

19.5.1 Prerequisites and governing statement

For composable maps, a local error in an inner map is multiplied by an outer Lipschitz bound.

$$\|g \circ f - \widetilde{g} \circ \widetilde{f}\| \le L_g \|f - \widetilde{f}\| + \|g - \widetilde{g}\|$$

Both inner images must lie in the region on which the outer bounds are valid.

$$E_m \le \sum_{j=1}^{m} \varepsilon_j \prod_{k=j+1}^{m} L_k.$$

19.5.2 Derivation and worked instance

Insert one approximate stage at a time and telescope the difference of the two compositions. Apply the Lipschitz bound to each remaining outer composition. The finite formula follows by induction. An admissible-image enclosure must be checked at every stage; an accurate endpoint alone does not guarantee that all intermediate expressions were defined.

Let the three errors be 0.0001, 0.0002, and 0.0003. With outer derivative bounds two and three, their propagated contributions are 0.0006, 0.0006, and 0.0003. The total is 0.0015; the unweighted sum is 0.0006.

19.5.3 Comparison and validity conditions

Baseline. Sum per-stage error tolerances without accounting for amplification.

Proposed procedure. Propagate each error with all downstream Lipschitz factors and domain guards.

Quantitative conclusion. The certificate retains both accumulated numerical error and domain validity.

Validity conditions. The factors are valid only on the certified intermediate regions. Analyticity does not make them global constants.

Exercise 19.5. What happens to the first-stage error if L2=1/2 and L3=1/3?

Solution. Its contribution is multiplied by 1/6. Later errors still receive their own downstream factors. □

Graphical reference. The shared construction is displayed in Volume I.

19.6 CG06. Restriction in different output norms

Mathematical reference. Volume I, normal-family and mapping core M34–M35; Volume II, analytic-approximation chapter C02.

19.6.1 Prerequisites and governing statement

Point evaluation is bounded on H^2 at every interior point. Cauchy-Schwarz converts a coefficient tail to a pointwise tail.

$$\sum_{j\geq N} |a_j| r^j \leq \left(\sum_{j\geq N} |a_j|^2 \right)^{1/2} \frac{r^N}{\sqrt{1-r^2}}$$

The factor $(1\text{-}r^2)^{\text{-}1/2}$ comes from a geometric sum of squared weights.

$$\sup_{|z|\leq r} |f(z) - P_N f(z)| \leq \frac{r^N}{\sqrt{1-r^2}} \|f\|_{H^2}.$$

19.6.2 Derivation and worked instance

Apply Cauchy-Schwarz to the coefficient tail and then maximize the geometric factor over $|z|\leq r$. Equality at a fixed positive r is attained by a tail coefficient vector proportional to r^j. This gives the exact norm of the discarded-tail evaluation functional, although it does not by itself prove the minimax value for all rank-N operators into the supremum-norm target.

At r=1/2,N=4, the upper bound is (1/16)/(sqrt3/2)=1/(8sqrt3), exceeding the H^2 restriction error 1/16.

19.6.3 Comparison and validity conditions

Baseline. Use the H^2 truncation error as a pointwise maximum without conversion.

Proposed procedure. State the input and output norms and derive the appropriate evaluation factor.

Quantitative conclusion. The output guarantee includes the exact evaluation-functional factor.

Validity conditions. A sharp norm for one specified truncation operator is a different statement from a global approximation width in a new target norm.

Exercise 19.6. Why can the extremizing coefficients depend on the evaluation point?

Solution. Cauchy-Schwarz equality requires alignment with the geometric coefficient vector of that point. The supremum permits choosing the worst point and corresponding input. □

19.7 CG07. Analytic continuation as an inverse problem

Mathematical reference. Volume II, analytic-approximation chapter C07

19.7.1 Prerequisites and governing statement

Two admissible inputs that yield nearly identical observations cannot always be distinguished in bounded adversarial noise.

$$\|f - g\|_{\text{obs}} \leq 2\eta$$

The midpoint of their observation data lies within η of both.

$$\max\{\|A(y) - f\|_{\text{out}}, \|A(y) - g\|_{\text{out}}\} \geq \tfrac{1}{2}\|f - g\|_{\text{out}}.$$

19.7.2 Derivation and worked instance

The triangle inequality gives the displayed bound for any common observed datum y. Scale a high-degree monomial so its observation norm is 2eta. Its output norm on a larger disk is multiplied by $(R/r)^N$. This creates an explicit information-theoretic error floor. A bounded analytic prior limits admissible scaling and yields a different conditional stability problem.

For r=1/4,R=3/4,N=10, the amplification is 59049. Observation uncertainty η therefore permits a worst-case output error of at least 59049 η in the unrestricted polynomial construction.

19.7.3 Comparison and validity conditions

Baseline. Interpret uniqueness of analytic continuation as numerical stability.

Proposed procedure. Construct indistinguishable input pairs and lower-bound every deterministic reconstruction error.

Quantitative conclusion. The lower bound applies to every deterministic reconstruction with the stated observations.

Validity conditions. The pair must remain inside the declared prior class. Ignoring a global norm bound would overstate the lower bound.

Exercise 19.7. Which proof step fails if the observed data for f and g have disjoint noise balls?

Solution. There is no common datum y admissible for both, so the indistinguishability argument does not apply. □

Graphical reference. See Figure 17.5 for the shared construction.

19.8 CG08. Formal contracts for numerical certificates

Mathematical reference. Volume II, analytic-approximation chapter C04 Volume IV, formal proof specifications.

19.8.1 Prerequisites and governing statement

A certificate checker proves an implication from validated finite premises to a finite conclusion. Analytic premises require their own evidence.

$$\text{valid}(c) \Longrightarrow \text{bound}(c)$$

The meaning of valid includes every denominator and sign condition used in the implication.

$$\text{analytic premises} \Rightarrow \text{finite witness} \Rightarrow \text{checked inequality.}$$

19.8.2 Derivation and worked instance

The local Z3 runs check real-closed-field consequences, including positivity margins and rational inequalities. The symbolic normalizer checks polynomial identities after denominators are identified. A full Lean proof would connect these facts to complex differentiability, integration, compactness, and the specified function spaces. Those dependencies have explicit unresolved entries in the release manifest.

The inequality m-δ>0 is sufficient for the finite margin step. It cannot establish that a supplied m is a global co-Lipschitz constant; that requires core M31 or another model proof.

19.8.3 Comparison and validity conditions

Baseline. Treat successful numerical tests as proof of a theorem with universal analytic quantifiers.

Proposed procedure. Separate representation validation, exact arithmetic implications, and the analytic soundness theorem.

Quantitative conclusion. Every computational result has a declared statement and trust boundary.

Validity conditions. The solver is part of the trusted computing base for its results. No Lean kernel acceptance is established for these calculations.

Exercise 19.8. Why is a file containing a theorem followed by an assumption of that theorem not a completed formal proof?

Solution. The conclusion has been introduced as a premise. Kernel checking of the resulting implication would leave the original mathematical obligation unproved. □

Chapter 20

Optimal recovery and numerical specifications

20.1 CP01. The competitor class in an optimality theorem

Mathematical reference. Volume I, normal-family and mapping core M35; the finite-rank optimality argument.

20.1.1 Prerequisites and governing statement

An approximation width depends on the admissible algorithms, input set, information and output norm.

$$\inf_{A\in\mathcal{A}} \sup_{f\in\mathcal{F}} \|Tf - Af\|$$

Changing either A or F changes the mathematical optimization problem.

$$\mathcal{F} = \{f : \|f\|_{H^2} \le 1\}, \quad \mathcal{A} = \{A : \operatorname{rank} A \le N,\ A \text{ bounded linear}\}.$$

20.1.2 Derivation and worked instance

The attained value r^N survives arbitrary changes of linear measurement basis. It does not quantify nonlinear encoders, variable-length codes, restricted training distributions, or expected-error objectives. These extensions are mathematically meaningful, but the rank-nullity lower bound alone does not settle them. A research claim should specify the new class before presenting empirical compression ratios.

A two-dimensional prior makes rank two exact. The same rank on the full H^2 unit ball retains error r^2. The different answers are consequences of different input sets.

20.1.3 Comparison and validity conditions

Baseline. Infer universal algorithmic superiority from comparison with a selected truncation.

Proposed procedure. Prove the exact minimax value for a fully quantified linear operator class.

Quantitative conclusion. The present optimum is global inside an explicit, nontrivial infinite-dimensional linear model.

Validity conditions. Complex coordinate count has no direct equality with byte count or sample count in a nonlinear representation.

Exercise 20.1. Can a data-dependent choice among linear encoders remain a linear operator?

Solution. Generally it is nonlinear as a function of the input. A separate proof is required to show that the combined mapping belongs to the bounded linear class. □

Graphical reference. The shared construction is displayed in Volume I.

20.2 CP02. A rational bracket for the optimal sample radius

Mathematical reference. Volume II, analytic-approximation chapter C03; graduate dossier CG02; exact experiment register.

20.2.1 Prerequisites and governing statement

A unique stationary point can be enclosed by opposite signs at rational endpoints. Positive denominator clearing preserves those signs.

$$P(q) = MNq^{N+n} - n\eta(1 - q^N)^2$$

For rational M and η, P has rational coefficients.

$$q_- < q_* < q_+, \qquad P(q_-) < 0 < P(q_+).$$

20.2.2 Derivation and worked instance

The strict-convexity theorem identifies a unique global minimizer. Exact rational polynomial signs enclose its radius. On this interval the decreasing noise term is bounded using the right endpoint and the increasing aliasing term using the left endpoint, giving a lower bound for the optimum. Evaluating B at either endpoint gives an upper bound. The resulting certificate contains only rational integers and the previously proved uniqueness statement.

For n=8,N=64,M=2,η=10^{-12}, the release checks endpoints $6555554687/10^{10}$ and $6555554688/10^{10}$. The width is exactly 10^{-10}.

20.2.3 Comparison and validity conditions

Baseline. Report a floating-point optimizer without a verified enclosure.

Proposed procedure. Clear denominators, evaluate P exactly at rational endpoints, and combine the signs with the uniqueness theorem.

Quantitative conclusion. The optimizer and optimum bound receive exact finite enclosures.

Validity conditions. The rational bracket certifies the stated error model; omitted transform-roundoff terms would alter the optimization problem.

Exercise 20.2. Why can a minimum upper bound be obtained by evaluating B at q_?

Solution. The global minimum is no larger than the value at any admissible point, including the left bracket endpoint. □

20.3 CP03. Information-theoretic impossibility from a mode pair

Mathematical reference. Volume II, analytic-approximation chapter C07

20.3.1 Prerequisites and governing statement

Adversarial observation noise defines a set of inputs consistent with one datum.

$$y \in B_\eta(\mathcal{O}f) \cap B_\eta(\mathcal{O}g)$$

The intersection is nonempty when the observation difference is at most 2eta.

$$E_{\text{worst}} \geq \frac{1}{2}\|f - g\|_{\text{out}}.$$

20.3.2 Derivation and worked instance

The reconstruction output for the common datum is identical in the two possible worlds. If both errors were below half the output separation, the triangle inequality would contradict their separation. Scaled monomials supply a continuum of such pairs. The proof addresses every deterministic reconstruction algorithm, independent of arithmetic precision or runtime.

The output-to-observation ratio for degree N between radii 1/4 and 3/4 is 3^N. A global analytic norm restriction can eliminate a proposed pair, so the input-validity test must be included in each lower-bound certificate.

20.3.3 Comparison and validity conditions

Baseline. Attribute every failed continuation to an implementation defect.

Proposed procedure. Exhibit a pair of admissible functions that no deterministic observer can distinguish at the noise level.

Quantitative conclusion. The impossibility claim has an explicit witness and a quantified algorithm class.

Validity conditions. A statistical noise model permits average-risk statements with different constants and quantifiers. The deterministic lower bound should not be relabelled as a statistical effect size.

Exercise 20.3. What is the minimum additional information needed to invalidate one chosen pair witness?

Solution. Any valid prior or observation excluding at least one member of the pair. It need not solve the entire inverse problem to remove that particular witness. □

Graphical reference. See Figure 17.5 for the shared construction.

20.4 CP04. Complexity of exact sign certificates

Mathematical reference. Exact stationary bracket and sample certificates; Volume II, analytic-approximation chapter C03–C04.

20.4.1 Prerequisites and governing statement

Rational arithmetic avoids rounding but has an integer bit cost. Raising a rational p/q to a power grows numerator and denominator lengths.

$$(p/q)^n = p^n/q^n$$

The integer lengths grow proportionally to n log2(max(|p|,q)).

$$\text{bits}(q^n) = O(n \log q).$$

20.4.2 Derivation and worked instance

For the radius bracket, q has a ten-decimal-digit denominator and the highest relevant power is 72. Clearing denominators yields integers of several thousand bits, still modest in this example. Repeated squaring reduces the number of multiplication stages, while operand sizes determine the actual cost. A verifier can recompute powers from the rational endpoints, so the certificate need not store every intermediate integer.

A certificate consisting of two rational endpoints, model parameters and expected signs is smaller than a full multiplication trace. An independently replayed trace can improve auditability at a larger serialization cost.

20.4.3 Comparison and validity conditions

Baseline. Call an exact rational check constant-time because it uses one displayed inequality.

Proposed procedure. Count integer bit lengths, multiplication strategy, and certificate serialization.

Quantitative conclusion. The complexity record accounts for the size of exact arithmetic operands.

Validity conditions. Asymptotic bit estimates are not measured runtime data. Different arbitrary-precision implementations can have different overhead.

Exercise 20.4. How many bits does $(10^{10})^{72}$ require, approximately?

Solution. It has 721 decimal digits and approximately floor(720 log2 10)+1=2392 bits. □

20.5 CP05. Domain guards for analytic geometry code

Mathematical reference. Volume II, analytic-approximation chapter C05–C06; Volume I, normal-family and mapping core M07 and M27.

20.5.1 Prerequisites and governing statement

A formula can be algebraically valid while a selected branch or denominator is invalid on part of its requested domain.

$$\phi_a^{-1}(w) = 2w/(1 + \sqrt{1 + 4aw})$$

The square root is selected by analytic continuation from the normalization point.

$$\text{domain}(w) \ \wedge \ \text{branch}(s, w) \ \wedge \ s^2 = 1 + 4aw.$$

20.5.2 Derivation and worked instance

The squared equation alone accepts both roots. A branch witness must distinguish them globally, for example by positive real part on a certified slit domain. The denominator 1+s then stays nonzero. The forward-inverse identity supplies a useful additional check but does not distinguish a globally inconsistent branch sequence on its own.

For the Koebe inverse, the Cayley coordinate gives an intrinsic branch rule: sqrt(1+4w) has positive real part, and the corresponding source coordinate lies in the disk.

20.5.3 Comparison and validity conditions

Baseline. Validate a map by checking a finite set of forward-inverse residuals.

Proposed procedure. Carry zero-exclusion, branch normalization, and image-containment obligations through the interface.

Quantitative conclusion. The certificate includes the hypotheses needed to interpret an inverse as a single-valued function.

Validity conditions. A path crossing the branch cut leaves the specified interface domain. A numerical continuation procedure requires a different, path-dependent interface.

Exercise 20.5. Why does s squared equal the discriminant fail to select the branch?

Solution. Both s and -s satisfy the same polynomial equation whenever the discriminant is nonzero. Additional normalization or continuation data are necessary. □

Graphical reference. The shared construction is displayed in Volume I.

20.6 CP06. Semantic ownership of a proof corpus

Mathematical reference. Canonical claim identity and proof-dependency conventions.

20.6.1 Prerequisites and governing statement

A canonical mathematical claim can have several expository views without becoming several different theorems.

$$\text{owner} : Q \to \mathcal{D}$$

The owner function assigns one substantive derivation to each claim identifier.

$$\text{unique coverage} = |\{q \in Q : \exists \text{ qualifying witness for } q\}|.$$

20.6.2 Derivation and worked instance

A lecture slide can cite a parent theorem and supply a distinct worked instance. The theorem belongs to the parent owner; the instance belongs to the educational dossier. A finite claim ledger distinguishes these contributions. A lexical duplicate detector catches exact paragraph reuse, while semantic duplication requires a statement-level review. The two checks have different scopes.

The same Koebe bound can appear in a proof, a field certificate, and a slide. It contributes one canonical theorem and several application views. New quantitative hypotheses or a new lower bound can create additional obligations when explicitly stated.

20.6.3 Comparison and validity conditions

Baseline. Increase apparent coverage by repeating statements across educational levels.

Proposed procedure. Count canonical obligations once and identify branch-specific additions separately.

Quantitative conclusion. The accounting is invariant under duplicating a document or slide.

Validity conditions. Zero exact duplicate paragraphs does not establish mathematical novelty or semantic uniqueness.

Exercise 20.6. What should happen to the unique-claim count after copying a PDF?

Solution. It must remain unchanged, because no new canonical claim or qualifying mathematical witness has been introduced. □

20.7 CP07. Mutation testing of theorem interfaces

Mathematical reference. Formal report; Volume II, analytic-approximation chapter C04 and C07.

20.7.1 Prerequisites and governing statement

A checker should reject a deliberately invalid certificate. Passing positive tests alone cannot establish that rejection paths work.

$$\mathrm{accept}(c) = \mathrm{true}, \qquad \mathrm{accept}(\widetilde{c}) = \mathrm{false}$$

The mutation must change a mathematically relevant field, not only a filename.

$$m > \delta \quad \text{is changed to} \quad m \geq \delta \quad \text{at } m = \delta.$$

20.7.2 Derivation and worked instance

At equality the positive-margin conclusion is false. This supplies an exact boundary counterexample to a mutated contract. Similar controls set the normalized derivative to zero, let a quadratic coefficient exceed its univalent regime, or overcount a coverage denominator. Each mutation is tied to the specific finite conclusion it invalidates. A collection of successful mutations still does not prove completeness of the checker.

The release records solver answers for valid real-closed-field obligations and separate satisfiable counterexamples to altered assertions. Analytic theorem coverage is not inferred from the number of solver calls.

20.7.3 Comparison and validity conditions

Baseline. Run only known valid inequalities.

Proposed procedure. Perturb denominators, strict margins, normalization conditions and sample bounds, then retain counterexample outcomes.

Quantitative conclusion. The negative controls exercise strict assumptions that matter at the boundary of validity.

Validity conditions. Several parameter instances of one schema count as regression instances, not independent theorem schemas.

Exercise 20.7. Give a concrete mutation that destroys the quadratic injectivity margin.

Solution. Set a=1,r=1 in 1-2ar. The margin becomes -1, so the asserted strict positive bound fails. □

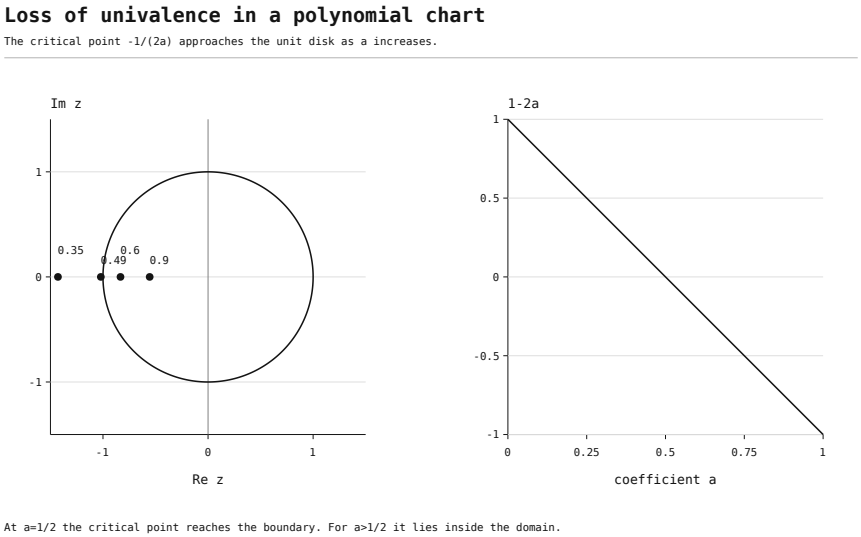


Figure 20.1: The negative controls exercise strict assumptions that matter at the boundary of validity. The figure is generated from the stated formula; its pixels have no independent proof status.

20.8 CP08. A layered trust record for machine-assisted mathematics

Mathematical reference. Formal verification manifest and core dependency audit.

20.8.1 Prerequisites and governing statement

A formal development has a trusted kernel, an environment, imported results and executable proof terms.

$$\text{statement} + \text{proof term} + \text{kernel acceptance}$$

A source script without a successful checker run has no recorded kernel acceptance.

$$\text{Lean acceptances} = 0 \quad \text{in this delivery.}$$

20.8.2 Derivation and worked instance

The recorded computational environment supplied a real-closed-field solver library and a symbolic algebra engine. Their accepted finite consequences are retained with explicit trust labels. The earlier formalization work produced candidate elementary lemmas and a theorem-coverage map. The analytic and topological formalization remains incomplete; no axiom restatement is counted as a completed proof.

A successful rational sign check can establish $P(q_-)<0$. To conclude that q_* lies to its right, one must also prove the relation between P and the optimization derivative and the uniqueness theorem.

20.8.3 Comparison and validity conditions

Baseline. Present generated theorem scripts as completed formal proofs.

Proposed procedure. Record the exact environment, checker outputs, assumptions and unresolved source-level obligations.

Quantitative conclusion. The trust record prevents a finite algebraic check from being promoted to a full-section theorem.

Validity conditions. Checksum integrity proves reproducible bytes. It supplies no missing mathematical implication.

Exercise 20.8. What is the mathematical role of a dependency audit?

Solution. It identifies each invoked theorem, matches its hypotheses to the application, and locates unresolved or circular proof obligations. □

References

[1] Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola. *Dive into Deep Learning*. Online source snapshot; book site 1.0.3.

[2] Oscar Levin. *Discrete Mathematics: An Open Introduction*. Fourth edition; September 2026 build (copyright 2025).

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms*. Fourth edition; 2022.

[4] Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Database System Concepts*. Seventh edition; 2020; retained conflicting sixth-edition title-page imprint.

[5] Steven Bird, Ewan Klein, Edward Loper. *Natural Language Processing with Python*. 2009; EPUB; Python 2-era examples retained.

[6] Torben Aegidius Mogensen. *Basics of Compiler Design*. Anniversary edition; 20 August 2010.

[7] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze. *Introduction to Information Retrieval*. Online draft; 1 April 2009.

[8] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. *Foundations of Machine Learning*. Second edition.

[9] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning*. Second edition; corrected twelfth printing; 13 January 2017.

[10] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor. *An Introduction to Statistical Learning: with Applications in Python*. First printing; 5 July 2023.

[11] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Second edition; corrected printing; 21 June 2023.

[12] Jacob Eisenstein. *Introduction to Natural Language Processing*. 2019; fixed-layout EPUB.