



Research article**Sharp joint Carathéodory compatibility and fifth-order lifting obstructions in bi-univalent function theory****Zainab H. Mahmood^{1,*}, Reem O. Rasheed², Bassim K. Mihsin³ and Waggas Galib Atshan⁴**¹ Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq² Department of Mathematics, College of Education Tuzkhurmatu, Tikrit University, Tikrit, Iraq³ General Directorate of Education in Karbala, Karbala, Iraq⁴ Department of Mathematics, College of Science, University of Al-Qadisiyah, Diwaniyah, Iraq*** Correspondence:** Email: zainab.mahmood@sc.uobaghdad.edu.iq.

Abstract: Let $\Psi(z) = 1 + L_1z + L_2z^2 + L_3z^3 + L_4z^4 + \cdots$ be analytic in $\mathbb{D}$ with $L_1, \dots, L_4 \in \mathbb{R}$ and $L_1 > 0$, and let $\mathcal{M}$ consist of the bi-univalent functions f for which a prescribed linear operator with coefficients $\Phi_2, \dots, \Phi_5 > 0$, applied to f and to f^{-1} , yields two functions subordinate to Ψ . Put $\kappa = L_1\Phi_3/\Phi_2^2 - L_2/L_1$. Coupling the two subordinations through the second-order Carathéodory parametrization gives $(4-t^2)(x+y) = 2\kappa t^2$, and hence $|c_1|^2 \leq 4/(1+|\kappa|)$ on the second-order compatible coefficient body V_2 . This envelope is sharp on V_2 and it is not asserted to be sharp for $\mathcal{M}$. Its equality set is a single rotation orbit of an explicit finite Blaschke pair; for $\kappa \geq 0$, the envelope coincides with the bound produced by the usual summation argument; for $\kappa < 0$, it improves it. We then ask whether this equality orbit can be the low-order truncation of a member of $\mathcal{M}$. The inverse-compatibility residuals R_1, R_2 , and R_3 vanish identically on it, and the residual R_4 , attached to the fifth coefficient, is $R_4 = -2P_\sigma[\Phi_4(L_1\Phi_2^2 + \sigma(L_1^2\Phi_3 - L_2\Phi_2^2))^2]^{-1}$, $\sigma = \text{sgn } \kappa$, where P_σ is an explicit polynomial that is affine in L_4 with positive leading coefficient. Thus $P_\sigma \neq 0$ means precisely that the equality orbit violates fifth-order inverse compatibility; it gives no information about other points of V_2 . Combining this with the rigidity of the equality orbit and normality of the Schwarz class, we prove that $P_\sigma \neq 0$ implies the strict separation $\sup_{f \in \mathcal{M}} |c_1|^2 < 4/(1+|\kappa|)$. The gap is not quantified, and the exact value of the supremum is left open. For the Janowski generator $(1 + Az)/(1 + Bz)$, one has $\kappa = A$ and $P_\sigma = A(A-B)^3(A+\sigma)(2A+\sigma-B)$, which is nonzero for $A \neq 0$; for $A = 0$, the corresponding polynomial vanishes and no separation follows. For $\Psi(z) = e^z$, one has $P_{+1} = 23/12$, whence $\sup_{f \in \mathcal{M}} |a_2| < \sqrt{2/3}$.

Keywords: bi-univalent functions; subordination; Schwarz functions; Carathéodory functions; inverse coefficients; coefficient body; joint compatibility; lifting obstruction

Mathematics Subject Classification: 30C45, 30C50, 30J10

Notation, standing hypotheses and equation numbers are those of the main text. Supplementary File S2 consists of the Python script `Math-30790_Supplementary_S2.py` and the present document, which describes the script, lists its checks, reproduces its output and gives its full source code (Section S2.5). The script is a second check of the displayed algebra, independent of the Wolfram computation of Supplementary File S1. Neither computation is used as a substitute for any proof in the main text.

Supplementary File S2

S2.1. Scope and exact arithmetic

The script uses Python 3 with SymPy (tested with SymPy 1.14.0). Every check is carried out in exact arithmetic: integers, rational numbers, symbols and the algebraic number $\sqrt{2}$. Decimal values are printed at the end for comparison only and are not used in any check. Each check prints PASS or stops with an assertion error.

The symbolic variables are $L_1, \dots, L_4$ (real); $\Phi_2, \dots, \Phi_5$ (positive); K (positive, $K = |\kappa|$ on the branch $\kappa \neq 0$); the second-order Carathéodory parameters t, x and the free higher Schwarz coefficients u_3, u_4, v_3, v_4 ; and the Janowski parameters A, B (real). The sign σ is substituted as $+1$ and -1 separately.

The analytic ingredients of the main text (Schwarz–Pick rigidity, the maximum principle, normal families and Montel’s theorem, and Cauchy coefficient convergence) are not, and cannot be, checked by this script; they are proved in the main text.

S2.2. Running the script

`python3 Math-30790_Supplementary_S2.py`

Requirements: Python ≥ 3.9 and SymPy ≥ 1.12 . A complete run takes a few minutes on a standard laptop.

S2.3. List of checks

The labels printed by the script refer to the equation, lemma and section numbers of the main text.

No.	Check (as printed by the script)
1	(2.16) inverse coefficients b2..b5
2	Lemma 2.3 coefficients A1..A4
3	Section 4: $R_1 = R_2 = 0$ for every V_2 point (any x , any u_3, u_4, v_3, v_4)
4	(3.15) coefficients of u_0 , $\sigma=1$
5	(3.16) $v_j = (-1)^j u_j$, $\sigma=1$
6	Lemma 4.3 $R_1=R_2=R_3=0$ and $a_3=a_2^2$, $\sigma=1$
7	(4.13) bracket form of R_4 , $\sigma=1$
8	(4.14) $R_4 = -2 P_\sigma / [\Phi^4 (L_1 \Phi^2 + \sigma M)^2]$, $\sigma=1$
9	(4.15)=(4.16) compact form $T_1+T_2+T_3 = P_\sigma$, $\sigma=1$

No.	Check (as printed by the script)
10	(4.16) P_{sigma} has 16 monomials, $\text{sigma}=1$
11	(4.27) $dP/dL4 = L1^2 \text{Phi}2^4 \text{Phi}4$ and $\text{deg}_{L4} = 1$, $\text{sigma}=1$
12	(3.15) coefficients of $u0$, $\text{sigma}=-1$
13	(3.16) $v_j = (-1)^j u_j$, $\text{sigma}=-1$
14	Lemma 4.3 $R1=R2=R3=0$ and $a3=a2^2$, $\text{sigma}=-1$
15	(4.13) bracket form of $R4$, $\text{sigma}=-1$
16	(4.14) $R4 = -2 P_{\text{sigma}}/[\text{Phi}4 (L1 \text{Phi}2^2 + \text{sigma} M)^2]$, $\text{sigma}=-1$
17	(4.15)=(4.16) compact form $T1+T2+T3 = P_{\text{sigma}}$, $\text{sigma}=-1$
18	(4.16) P_{sigma} has 16 monomials, $\text{sigma}=-1$
19	(4.27) $dP/dL4 = L1^2 \text{Phi}2^4 \text{Phi}4$ and $\text{deg}_{L4} = 1$, $\text{sigma}=-1$
20	Remark 4.8 / (4.30): $\text{kappa}=0$ gives $R1=R2=R3=0$ and $R4 = -2P0/(\text{Phi}4 L1^2 \text{Phi}2^4)$
21	P_{sigma} restricted to $M=0$ equals $P0$ (both signs)
22	(6.4)-(6.5) Janowski $L_n = (-B)^{(n-1)}(A-B)$
23	(6.6) Janowski $\text{kappa} = A$
24	(6.8) Janowski $P_{\text{sigma}} = A(A-B)^3(A+s)(2A+s-B)$, $s=1$
25	(6.8) Janowski $P_{\text{sigma}} = A(A-B)^3(A+s)(2A+s-B)$, $s=-1$
26	Janowski $A=0$: $P0 = 0$
27	(6.14) $P_{\{+1\}} = 6 - 11L2 + 6L2^2 + L2^3 - 3L2L3 + L4$
28	(6.11) exponential $\text{kappa} = 1/2$
29	(6.12) exponential $P_{\{+1\}} = 23/12$
30	(6.12) exponential $L4_{\text{lift}} = -15/8$
31	(6.17)-(6.20) brackets and Phi_n
32	(6.21) $\text{kappa} = 4043 \sqrt{2}/40500 > 0$
33	(6.22) $L4_{\text{lift}}$ in $Q(\sqrt{2})$ and negative; (4.28) evaluated
34	App. C: $L1^2 \text{Phi}2^4 \text{Phi}4 = 7381125/234256$

S2.4. Output of a complete run

PASS (2.16) inverse coefficients $b2..b5$
 PASS Lemma 2.3 coefficients $A1..A4$
 PASS Section 4: $R1 = R2 = 0$ for every $V2$ point (any x , any $u3, u4, v3, v4$)
 PASS (3.15) coefficients of $u0$, $\text{sigma}=1$
 PASS (3.16) $v_j = (-1)^j u_j$, $\text{sigma}=1$
 PASS Lemma 4.3 $R1=R2=R3=0$ and $a3=a2^2$, $\text{sigma}=1$
 PASS (4.13) bracket form of $R4$, $\text{sigma}=1$
 PASS (4.14) $R4 = -2 P_{\text{sigma}}/[\text{Phi}4 (L1 \text{Phi}2^2 + \text{sigma} M)^2]$, $\text{sigma}=1$
 PASS (4.15)=(4.16) compact form $T1+T2+T3 = P_{\text{sigma}}$, $\text{sigma}=1$
 PASS (4.16) P_{sigma} has 16 monomials, $\text{sigma}=1$
 PASS (4.27) $dP/dL4 = L1^2 \text{Phi}2^4 \text{Phi}4$ and $\text{deg}_{L4} = 1$, $\text{sigma}=1$
 PASS (3.15) coefficients of $u0$, $\text{sigma}=-1$
 PASS (3.16) $v_j = (-1)^j u_j$, $\text{sigma}=-1$

PASS Lemma 4.3 $R_1=R_2=R_3=0$ and $a_3=a^2$, $\sigma=-1$
 PASS (4.13) bracket form of R_4 , $\sigma=-1$
 PASS (4.14) $R_4 = -2 P_\sigma / [\Phi^4 (L^2 \Phi^2 + \sigma M)^2]$, $\sigma=-1$
 PASS (4.15)=(4.16) compact form $T_1+T_2+T_3 = P_\sigma$, $\sigma=-1$
 PASS (4.16) P_σ has 16 monomials, $\sigma=-1$
 PASS (4.27) $dP/dL = L^2 \Phi^2 \Phi^4$ and $\deg_L = 1$, $\sigma=-1$
 PASS Remark 4.8 / (4.30): $\kappa=0$ gives $R_1=R_2=R_3=0$ and $R_4 = -2P_0/(\Phi^4 L^2 \Phi^2)$
 PASS P_σ restricted to $M=0$ equals P_0 (both signs)
 PASS (6.4)-(6.5) Janowski $L_n = (-B)^{n-1}(A-B)$
 PASS (6.6) Janowski $\kappa = A$
 PASS (6.8) Janowski $P_\sigma = A(A-B)^3(A+s)(2A+s-B)$, $s=1$
 PASS (6.8) Janowski $P_\sigma = A(A-B)^3(A+s)(2A+s-B)$, $s=-1$
 PASS Janowski $A=0$: $P_0 = 0$
 PASS (6.14) $P_{+1} = 6 - 11L^2 + 6L^2 + L^3 - 3L^2L_3 + L_4$
 PASS (6.11) exponential $\kappa = 1/2$
 PASS (6.12) exponential $P_{+1} = 23/12$
 PASS (6.12) exponential $L_4\text{-lift} = -15/8$
 PASS (6.17)-(6.20) brackets and Φ_n
 PASS (6.21) $\kappa = 4043 \sqrt{2}/40500 > 0$
 PASS (6.22) $L_4\text{-lift}$ in $Q(\sqrt{2})$ and negative; (4.28) evaluated
 PASS App. C: $L^2 \Phi^2 \Phi^4 = 7381125/234256$

Decimal values (comparison only):

$\kappa = 0.14117692426356600697$
 $L_4\text{-lift} = -0.14093869699366430893$
 $\text{envelope} = 0.32360757469871602903$

SymPy version: 1.14.0

All checks passed.

S2.5. Source code

The listing below is identical to the file `Math-30790_Supplementary_S2.py`.

```
#!/usr/bin/env python3
```

```
"""
```

```
Supplementary File S2 for Math-30790
```

```
"Sharp joint Caratheodory compatibility and fifth-order lifting obstructions
in bi-univalent function theory"
```

Independent exact symbolic verification of the displayed algebra.

This script is NOT part of any proof in the paper; every identity it checks is derived by hand in the manuscript. It uses only exact arithmetic (SymPy rationals, symbols and the algebraic number $\sqrt{2}$); no floating-point value is used in any check. Decimal values are printed for comparison only.

Requirements: Python ≥ 3.9 , SymPy ≥ 1.12 . Run: `python3 Math-30790_Supplementary_S2.py`
 Each check prints PASS or raises AssertionError.

Equation numbers in the check labels refer to the revised manuscript (R2 version).
Typical run time: about 3-4 minutes on a laptop.

Symbolic variables

```

L1,L2,L3,L4      real      (Taylor coefficients of Psi)
Phi2,...,Phi5    positive  (operator coefficients)
K                positive  (K = |kappa| on the branch kappa != 0)
sigma            substituted separately by +1 and -1
t, x             free      (second-order Caratheodory parameters, check C0)
A, B            real      (Janowski parameters)
"""

import sympy as sp

def check(name, cond):
    assert cond, "FAIL: " + name
    print("PASS " + name)

z, w = sp.symbols('z w')
L1, L2, L3, L4 = sp.symbols('L1 L2 L3 L4', real=True)
P2, P3, P4, P5 = sp.symbols('Phi2 Phi3 Phi4 Phi5', positive=True)
K = sp.symbols('K', positive=True)
M = L1**2*P3 - L2*P2**2
kappa = L1*P3/P2**2 - L2/L1

# ---- (2.16) reversion b2..b5 -----
a = sp.symbols('a2:6'); b = sp.symbols('b2:6')
g = w + sum(b[i]*w**(i+2) for i in range(4))
fg = sp.expand(g + sum(a[i]*g**(i+2) for i in range(4)))
sol = sp.solve([fg.coeff(w, k) for k in range(2, 6)], b, dict=True)[0]
a2, a3, a4, a5 = a
rev = [-a2, 2*a2**2 - a3, -5*a2**3 + 5*a2*a3 - a4,
        14*a2**4 - 21*a2**2*a3 + 6*a2*a4 + 3*a3**2 - a5]
check("(2.16) inverse coefficients b2..b5",
      all(sp.expand(sol[b[i]] - rev[i]) == 0 for i in range(4)))

# ---- Lemma 2.3: A_j as polynomials in u_j -----
def Acoef(u, LL=(L1, L2, L3, L4)):
    l1, l2, l3, l4 = LL
    u1, u2, u3, u4 = u
    return [l1*u1, l1*u2 + l2*u1**2, l1*u3 + 2*l2*u1*u2 + l3*u1**3,
            l1*u4 + l2*(2*u1*u3 + u2**2) + 3*l3*u1**2*u2 + l4*u1**4]
us = sp.symbols('u1:5')
U = sum(us[i]*z**(i+1) for i in range(4))
PsiU = sp.expand(L1*U + L2*U**2 + L3*U**3 + L4*U**4)
check("Lemma 2.3 coefficients A1..A4",
      all(sp.expand(PsiU.coeff(z, j+1) - Acoef(us)[j]) == 0 for j in range(4)))

def residuals(u, v, LL=(L1, L2, L3, L4), PH=(P2, P3, P4, P5)):
    Aj, Bj = Acoef(u, LL), Acoef(v, LL)
    aa = [Aj[j]/PH[j] for j in range(4)] # forced a2..a5, (4.1)
    bb = [r.subs(dict(zip(a, aa)), simultaneous=True) for r in rev]
    return [PH[j]*bb[j] - Bj[j] for j in range(4)], aa, Aj

# ---- C0: R1 = R2 = 0 at every point of V2 (second-order data only) -----
t, x, u3, u4, v3, v4 = sp.symbols('t x u3 u4 v3 v4')
y = 2*kappa*t**2/(4 - t**2) - x # (3.8)
R0, _, _ = residuals([t/2, (4 - t**2)*x/4, u3, u4],
                     [-t/2, (4 - t**2)*y/4, v3, v4])
check("Section 4: R1 = R2 = 0 for every V2 point (any x, any u3,u4,v3,v4)",

```

```

sp.simplify(R0[0]) == 0 and sp.simplify(R0[1]) == 0)

# ---- equality orbit, both sign branches -----
Psig = {}
for s in (1, -1):
    aa0 = 1/sp.sqrt(1 + K)
    u0 = z*(aa0 + s*z)/(1 + aa0*s*z)
    v0 = z*(-aa0 + s*z)/(1 - aa0*s*z)
    su = sp.series(u0, z, 0, 5).remove0(); sv = sp.series(v0, z, 0, 5).remove0()
    uc = [sp.simplify(su.coeff(z, j)) for j in range(1, 5)]
    vc = [sp.simplify(sv.coeff(z, j)) for j in range(1, 5)]
    exp_u = [1/sp.sqrt(1+K), s*K/(1+K), -K/(1+K)**sp.Rational(3, 2), s*K/(1+K)**2]
    check(f"(3.15) coefficients of u0, sigma={s}",
          all(sp.simplify(uc[j] - exp_u[j]) == 0 for j in range(4)))
    check(f"(3.16) v_j = (-1)^j u_j, sigma={s}",
          all(sp.simplify(vc[j] - (-1)**(j+1)*uc[j]) == 0 for j in range(4)))
    # substitute K = s*kappa (branch sgn kappa = s)
    Kb = s*M/(L1*P2**2)
    ub = [e.subs(K, Kb) for e in exp_u]
    vb = [(-1)**(j+1)*ub[j] for j in range(4)]
    R, aa, Aj = residuals(ub, vb)
    check(f"Lemma 4.3 R1=R2=R3=0 and a3=a2^2, sigma={s}",
          all(sp.simplify(R[j]) == 0 for j in range(3))
          and sp.simplify(aa[1] - aa[0]**2) == 0)
    bracket = -2*(2*P5*Aj[0]**4/P2**4 - 3*P5*Aj[0]*Aj[2]/(P2*P4) + Aj[3])
    check(f"(4.13) bracket form of R4, sigma={s}", sp.simplify(R[3] - bracket) == 0)
    P = (-L1**2*L2*P2**4*P4 + 2*s*L1*L2**2*P2**4*P4 + L2**3*P2**4*P4
          - 3*L1*L2*L3*P2**4*P4 + L1**2*L4*P2**4*P4
          + L1**4*P2**2*P3*P4 - 2*s*L1**3*L2*P2**2*P3*P4 - 2*L1**2*L2**2*P2**2*P3*P4
          + 3*L1**3*L3*P2**2*P3*P4 + L1**4*L2*P3**2*P4
          - 3*s*L1**3*L2*P2**3*P5 + 6*L1**2*L2**2*P2**3*P5 - 3*L1**3*L3*P2**3*P5
          + 3*s*L1**5*P2*P3*P5 - 6*L1**4*L2*P2*P3*P5 + 2*L1**6*P4*P5)
    Psig[s] = P
    check(f"(4.14) R4 = -2 P_sigma/[Phi4 (L1 Phi2^2 + sigma M)^2], sigma={s}",
          sp.simplify(R[3] + 2*P/(P4*(L1*P2**2 + s*M)**2)) == 0)
    Ms = sp.symbols('M')
    T1 = 2*L1**6*P4*P5
    T2 = -3*L1**2*P2*P5*(-s*Ms*L1 + 2*Ms*L2 + L1*L3*P2**2)
    T3 = P4*(Ms*L1**2*P2**2 + Ms**2*L2 - 2*s*Ms*L1*L2*P2**2 + 3*Ms*L1*L3*P2**2 + L1**2*P2**4*L4)
    check(f"(4.15)=(4.16) compact form T1+T2+T3 = P_sigma, sigma={s}",
          sp.expand((T1 + T2 + T3).subs(Ms, M) - P) == 0)
    check(f"(4.16) P_sigma has 16 monomials, sigma={s}", len(sp.Add.make_args(sp.expand(P))) == 16)
    check(f"(4.27) dP/dL4 = L1^2 Phi2^4 Phi4 and deg_L4 = 1, sigma={s}",
          sp.expand(sp.diff(P, L4) - L1**2*P2**4*P4) == 0 and sp.degree(P, L4) == 1)

# ---- kappa = 0 branch -----
R, _, _ = residuals([1, 0, 0, 0], [-1, 0, 0, 0])
L2k0 = L1**2*P3/P2**2
P0 = 2*L1**6*P4*P5 - 3*L1**3*L3*P2**3*P5 + L1**2*L4*P2**4*P4
check("Remark 4.8 / (4.30): kappa=0 gives R1=R2=R3=0 and R4 = -2P0/(Phi4 L1^2 Phi2^4)",
      all(sp.simplify(R[j].subs(L2, L2k0)) == 0 for j in range(3))
      and sp.simplify(R[3].subs(L2, L2k0) + 2*P0/(P4*L1**2*P2**4)) == 0)
check("P_sigma restricted to M=0 equals P0 (both signs)",
      all(sp.simplify(Psig[s].subs(L2, L2k0) - P0) == 0 for s in (1, -1)))

# ---- Janowski -----
A, B = sp.symbols('A B', real=True)
ser = sp.series((1 + A*z)/(1 + B*z), z, 0, 5).remove0()
Ln = [(-B)**(n-1)*(A - B) for n in range(1, 5)]

```

```

check("(6.4)-(6.5) Janowski  $L_n = (-B)^{(n-1)}(A-B)$ ",
      all(sp.simplify(ser.coeff(z, n) - Ln[n-1]) == 0 for n in range(1, 5)))
check("(6.6) Janowski  $\kappa = A$ ", sp.simplify(Ln[0] - Ln[1]/Ln[0] - A) == 0)
for s in (1, -1):
    Pj = Psig[s].subs({L1: Ln[0], L2: Ln[1], L3: Ln[2], L4: Ln[3],
                      P2: 1, P3: 1, P4: 1, P5: 1}, simultaneous=True)
    check(f"(6.8) Janowski  $P_{\text{sigma}} = A(A-B)^3(A+s)(2A+s-B)$ ,  $s=\{s\}$ ",
          sp.expand(Pj - A*(A - B)**3*(A + s)*(2*A + s - B)) == 0)
P0j = P0.subs({L1: Ln[0], L3: Ln[2], L4: Ln[3], P2: 1, P3: 1, P4: 1, P5: 1},
             simultaneous=True).subs(A, 0)
check("Janowski  $A=0: P_0 = 0$ ", sp.expand(P0j) == 0)

# ---- exponential -----
Pe = sp.expand(Psig[1].subs({L1: 1, P2: 1, P3: 1, P4: 1, P5: 1}))
check("(6.14)  $P_{+1} = 6 - 11L_2 + 6L_2^2 + L_2^3 - 3L_2L_3 + L_4$ ",
      sp.expand(Pe - (6 - 11*L2 + 6*L2**2 + L2**3 - 3*L2*L3 + L4)) == 0)
ev = {L2: sp.Rational(1, 2), L3: sp.Rational(1, 6), L4: sp.Rational(1, 24)}
check("(6.11) exponential  $\kappa = 1/2$ ", (1 - sp.Rational(1, 2)) == sp.Rational(1, 2))
check("(6.12) exponential  $P_{+1} = 23/12$ ", Pe.subs(ev) == sp.Rational(23, 12))
lift_e = sp.solve(Pe.subs({L2: ev[L2], L3: ev[L3]}), L4)[0]
check("(6.12) exponential  $L_4\text{lift} = -15/8$ ", lift_e == sp.Rational(-15, 8))

# ---- (p,q) example -----
p, q, d, lam, gam = sp.Rational(9, 10), sp.Rational(1, 2), sp.Rational(1, 10), 1, sp.Rational(1, 2)
br = lambda n: (p**n - q**n)/(p - q)
Ph = {n: ((br(n) + d)/(1 + d))**lam*(1 + (n - 1)*gam) for n in range(2, 6)}
check("(6.17)-(6.20) brackets and  $\Phi_n$ ",
      [br(n) for n in range(2, 6)] == [sp.Rational(7, 5), sp.Rational(151, 100),
                                       sp.Rational(371, 250), sp.Rational(13981, 10000)]
      and [Ph[n] for n in range(2, 6)] == [sp.Rational(45, 22), sp.Rational(161, 55),
                                           sp.Rational(18, 5), sp.Rational(44943, 11000)])
l1, l2, l3 = 1/sp.sqrt(2), sp.Rational(1, 4), 1/(12*sp.sqrt(2))
kp = sp.radsimp(l1*Ph[3]/Ph[2]**2 - l2/l1)
check("(6.21)  $\kappa = 4043 \sqrt{2}/40500 > 0$ ",
      sp.simplify(kp - 4043*sp.sqrt(2)/40500) == 0 and kp > 0)
X = sp.symbols('X')
lift = sp.solve(Psig[1].subs({L1: l1, L2: l2, L3: l3, L4: X, P2: Ph[2], P3: Ph[3],
                             P4: Ph[4], P5: Ph[5]}), X)[0]
claimed = (-3084612892 - 1088153235*sp.sqrt(2))/32805000000
check("(6.22)  $L_4\text{lift}$  in  $Q(\sqrt{2})$  and negative; (4.28) evaluated",
      sp.simplify(lift - claimed) == 0 and claimed < 0)
check("App. C:  $L_1^2 \Phi_2^4 \Phi_4 = 7381125/234256$ ",
      sp.simplify(l1**2*Ph[2]**4*Ph[4] - sp.Rational(7381125, 234256)) == 0)
env = l1/(Ph[2]*sp.sqrt(1 + kp))
print("\nDecimal values (comparison only):")
print("   $\kappa$       =", sp.N(kp, 20))
print("   $L_4\text{lift}$    =", sp.N(claimed, 20))
print("  envelope    =", sp.N(env, 20))
print("\nSymPy version:", sp.__version__)
print("All checks passed.")

```

