Challenge 03:
Celestial CFT Bootstrap

Comprehensive Technical Report

Domain:	Quantum Gravity & Particle Physics
Difficulty:	High
Timeline:	6–12 months
Prerequisites:	Scattering amplitudes, conformal field theory, Mellin transforms, representation theory

Pure Thought AI Challenges
January 19, 2026

Contents

1 Executive Summary

Celestial holography represents a revolutionary approach to understanding quantum gravity in flat spacetime. It reformulates 4D flat-space scattering amplitudes as correlation functions in a 2D conformal field theory living on the **celestial sphere** at null infinity. This challenge aims to apply the powerful **conformal bootstrap** methodology to constrain and classify consistent celestial CFTs compatible with graviton scattering.

Analysis Note

Unlike AdS/CFT, which relates gravity in anti-de Sitter space to CFT on its boundary, celestial holography provides a holographic description for *asymptotically flat* spacetimes—the relevant arena for actual gravitational wave observations and particle physics experiments.

The key innovation is the **Mellin transform**, which converts momentum-space amplitudes into celestial amplitudes carrying definite conformal weights. These celestial amplitudes satisfy conformal Ward identities, crossing symmetry, and soft theorems—providing a rich set of constraints amenable to bootstrap analysis.

2 Scientific Context and Motivation

2.1 From AdS/CFT to Flat Space Holography

The AdS/CFT correspondence has been spectacularly successful, but the real universe is not Anti-de Sitter—it is asymptotically flat (or de Sitter, accounting for dark energy). The quest for a **flat space holographic principle** has led to several approaches:

- BMS symmetry and soft theorems (Strominger et al.)
- Carrollian CFT on null infinity
- **Celestial holography:** 2D CFT on the celestial sphere

Physical Insight

The Celestial Sphere: Consider 4D Minkowski space $\mathbb{R}^{1,3}$. Null infinity $I^{\pm}$ has topology $\mathbb{R} \times S^2$. The S^2 factor is the **celestial sphere**—the sphere of directions from which massless particles can arrive or depart. Celestial holography posits that 4D gravity is equivalent to a 2D CFT on this S^2.

2.2 Celestial Amplitudes via Mellin Transform

The key mathematical operation is the **Mellin transform** from momentum space to the "celestial basis":

Definition 2.1 (Celestial Amplitude)**.** Given a momentum-space amplitude $A(\omega_i, z_i, \bar{z}_i)$ for n massless particles, the celestial amplitude is:

$$\boxed{\tilde{A}(\Delta_i, z_i, \bar{z}_i) = \int_0^\infty \prod_{i=1}^{n} d\omega_i \, \omega_i^{\Delta_i - 1} \, A(\omega_i, z_i, \bar{z}_i)} \tag{1}$$

where:

- ω_i are energy variables (with momenta parametrized as $p_i^\mu = \omega_i(1, \hat{n}_i)$)

- $(z_i, \bar{z}_i)$ are stereographic coordinates on the celestial sphere
- Δ_i are **conformal weights** (generally complex)

Analysis Note

The Mellin transform trades energy ω for conformal weight Δ. Particles with definite energy become superpositions over the principal continuous series of $\mathrm{SL}(2,\mathbb{C})$ representations. The resulting celestial amplitudes transform as CFT correlators under Lorentz transformations.

2.3 The Core Question

Central Research Question

What is the consistent space of celestial CFTs compatible with graviton scattering amplitudes?
The celestial CFT must satisfy:

1. $\mathrm{SL}(2,\mathbb{C})$ covariance (celestial conformal symmetry)
2. Crossing symmetry (OPE associativity)
3. Unitarity (positive norms / positive spectral density)
4. Weinberg soft graviton theorem
5. Regge boundedness at high energies

2.4 Why This Matters

(1) New Holographic Paradigm: Connects 4D gravity to 2D CFT without requiring AdS geometry—directly applicable to the real universe.

(2) Rigorous Constraints: The bootstrap approach carves out the space of consistent theories using only fundamental principles.

(3) Testable Predictions: Produces "islands" (allowed regions) and "no-go" exclusions that can be verified by explicit amplitude calculations.

(4) Unification: May connect to the broader program of flat-space holography, including BMS symmetry and memory effects.

3 Mathematical Formulation

3.1 Kinematics: Celestial Sphere Coordinates

A null 4-momentum in Minkowski space can be parametrized as:

$$p^\mu = \omega\, q^\mu(z,\bar{z}), \quad q^\mu = (1 + z\bar{z}, z + \bar{z}, -i(z - \bar{z}), 1 - z\bar{z}) \tag{2}$$

where $(z, \bar{z}) \in \mathbb{C}$ are stereographic coordinates on $S^2 \cong \mathbb{CP}^1$.

Mathematical Structure

Lorentz Group Action: The Lorentz group $\mathrm{SL}(2,\mathbb{C})$ acts on the celestial sphere via Möbius transformations:

$$z \mapsto \frac{az+b}{cz+d}, \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}(2,\mathbb{C}) \tag{3}$$

This is precisely the global conformal group of a 2D CFT!

3.2 Conformal Primary Wavefunctions

Instead of plane waves $e^{ip\cdot x}$, celestial holography uses **conformal primary wavefunctions**:

$$\phi^{\pm}_{\Delta}(X; z, \bar{z}) = \int_0^\infty d\omega\, \omega^{\Delta-1} e^{\pm i\omega q\cdot X} \tag{4}$$

These transform under Lorentz transformations as:

$$\phi^{\pm}_{\Delta}(X; z, \bar{z}) \to |cz+d|^{-2\Delta} \phi^{\pm}_{\Delta}(X; z', \bar{z}') \tag{5}$$

which is precisely the transformation of a **CFT primary operator** with weights $(\Delta, \bar{\Delta}) = (\Delta/2, \Delta/2)$ for scalars.

3.3 Celestial Amplitudes as CFT Correlators

The celestial amplitude $\tilde{A}(\Delta_i, z_i, \bar{z}_i)$ transforms as an n-point correlator in 2D CFT:

$$\tilde{A}(\Delta_i, z_i, \bar{z}_i) \sim \langle \mathcal{O}_{\Delta_1}(z_1, \bar{z}_1) \cdots \mathcal{O}_{\Delta_n}(z_n, \bar{z}_n) \rangle_{\mathrm{CFT}} \tag{6}$$

Analysis Note

Key Insight: This is not a metaphor—celestial amplitudes *are* CFT correlators. They satisfy conformal Ward identities, have OPE expansions, and (conjecturally) arise from some underlying 2D CFT on the celestial sphere.

3.4 Constraints on Celestial CFT

3.4.1 Constraint 1: $\mathrm{SL}(2,\mathbb{C})$ Covariance

Celestial amplitudes must transform covariantly under Lorentz (= conformal) transformations:

$$\tilde{A}(\Delta_i, z_i, \bar{z}_i) \to \prod_i |cz_i+d|^{-2\Delta_i} \tilde{A}(\Delta_i, z'_i, \bar{z}'_i) \tag{7}$$

This determines the kinematic structure of n-point functions up to conformally invariant cross-ratios.

3.4.2 Constraint 2: Crossing Symmetry (OPE Associativity)

The **Operator Product Expansion (OPE)** in the celestial CFT:

$$\mathcal{O}_{\Delta_1}(z_1)\mathcal{O}_{\Delta_2}(z_2) = \sum_{\Delta} C^{\Delta}_{\Delta_1\Delta_2} \frac{\mathcal{O}_{\Delta}(z_2)}{(z_1-z_2)^{\Delta_1+\Delta_2-\Delta}} \tag{8}$$

Crossing symmetry requires that the OPE is *associative*:

$$\boxed{(\mathcal{O}_1\mathcal{O}_2)\mathcal{O}_3 = \mathcal{O}_1(\mathcal{O}_2\mathcal{O}_3)} \tag{9}$$

For 4-point functions, this gives the **crossing equation**:

$$\sum_{\Delta_s} C_s^2(\Delta_s) G_{\Delta_s}(z, \bar{z}) = \sum_{\Delta_t} C_t^2(\Delta_t) G_{\Delta_t}(1 - z, 1 - \bar{z}) \tag{10}$$

where G_Δ are conformal blocks.

3.4.3 Constraint 3: Unitarity

For a unitary CFT, the OPE coefficients squared must be non-negative:

$$\boxed{|C^{\Delta}_{\Delta_1 \Delta_2}|^2 \geq 0} \tag{11}$$

This is the **positivity condition** that makes the bootstrap a convex optimization problem.

Critical Consideration

Subtlety: Celestial CFT involves the **principal continuous series** of $\mathrm{SL}(2, \mathbb{C})$, where $\Delta = 1 + i\lambda$ with $\lambda \in \mathbb{R}$. Unitarity must be properly defined for this non-compact situation.

3.4.4 Constraint 4: Soft Graviton Theorem

Weinberg's **soft graviton theorem** states that in the limit where one graviton becomes soft ($\omega \to 0$):

$$A_{n+1}(\omega \to 0) \sim \frac{S^{(0)}}{\omega} A_n + S^{(1)} A_n + O(\omega) \tag{12}$$

In the celestial basis, this becomes a **pole structure**:

$$\boxed{\tilde{A}(\Delta \to 0) \sim \frac{S^{(0)}}{\Delta} + \frac{S^{(1)}}{\Delta^2} + \cdots} \tag{13}$$

where $S^{(0)}$ is the Weinberg soft factor (sum of $\varepsilon \cdot p_i / p \cdot p_i$ terms).

3.4.5 Constraint 5: Regge Boundedness

At high energies, the amplitude must be bounded:

$$|\tilde{A}(\Delta)| \lesssim |\Delta|^N \quad \text{as } |\Delta| \to \infty \tag{14}$$

This ensures convergence of OPE sums and dispersion relations.

3.5 Bootstrap Formulation

Celestial Bootstrap Problem

Find: Celestial OPE data $\{C_{ijk}, \Delta_i\}$
Satisfying:

1. $\mathrm{SL}(2,\mathbb{C})$ covariance
2. Crossing symmetry (OPE associativity)
3. Unitarity ($C^2 \geq 0$ or appropriate positivity for continuous spectrum)
4. Soft theorems (pole structure at $\Delta \to 0, 1$)
5. Regge boundedness

Goal: Either construct explicit OPE data, or prove no solution exists for given assumptions (e.g., minimal gap Δ_{gap}).

4 Implementation Approach

4.1 Phase 1: Celestial Amplitude Calculator (Months 1–2)

Goal: Build a Mellin transform engine to compute celestial amplitudes from momentum-space expressions.

4.1.1 Mellin Transform Implementation

```python
from mpmath import mp
import numpy as np
from scipy.integrate import quad

mp.dps = 50  # High precision arithmetic

def mellin_transform(amplitude_func, omega_vars: list,
                     delta_vars: list, z_vars: list) -> complex:
    """
    Compute celestial amplitude via Mellin transform

    A_tilde(Delta_i) = integral d^n omega prod_i omega_i^{Delta_i-1} A(omega_i)

    Args:
        amplitude_func: Momentum-space amplitude A(omega, z, zbar)
        omega_vars: List of energy integration variables
        delta_vars: List of conformal weights (can be complex!)
        z_vars: List of celestial coordinates (z, zbar) pairs

    Returns:
        Celestial amplitude value (complex)
    """
    n_particles = len(omega_vars)

    def integrand(*omegas):
        # Momentum-space amplitude
        A = amplitude_func(omegas, z_vars)

        # Mellin weight factors
        weight = 1.0
```

```python
        for i, (omega, delta) in enumerate(zip(omegas, delta_vars)):
            weight *= omega**(delta - 1)

        return A * weight

    # Multi-dimensional integration
    # For principal series: Delta = 1 + i*lambda, integrate along contour
    result = multi_integrate(integrand, [(0, np.inf)] * n_particles)

    return result


def multi_integrate(func, limits, method='adaptive'):
    """
    Multi-dimensional numerical integration

    Uses nested quadrature for moderate dimensions,
    Monte Carlo for high dimensions.
    """
    from scipy.integrate import nquad

    result, error = nquad(func, limits)
    return result
```

Listing 1: Mellin transform for celestial amplitudes

4.1.2 Three-Point Graviton Amplitude

```python
def graviton_3pt_momentum(omega1, omega2, omega3, z1, z2, z3,
                          zbar1, zbar2, zbar3, helicities):
    """
    Tree-level 3-graviton amplitude in momentum space

    For (+++) or (---) helicity configurations, this vanishes.
    For mixed helicity, involves spinor-helicity brackets.

    Args:
        omega_i: Energies
        z_i, zbar_i: Celestial coordinates
        helicities: Tuple of helicities (+1 or -1)

    Returns:
        Amplitude (includes momentum-conserving delta function)
    """
    # Spinor-helicity brackets
    # <ij> = sqrt(omega_i * omega_j) * (zbar_i - zbar_j)
    # [ij] = sqrt(omega_i * omega_j) * (z_i - z_j)

    h1, h2, h3 = helicities

    # MHV amplitude: (++-) configuration
    if h1 == 1 and h2 == 1 and h3 == -1:
        bracket_12 = np.sqrt(omega1 * omega2) * (zbar1 - zbar2)
        bracket_23 = np.sqrt(omega2 * omega3) * (zbar2 - zbar3)
        bracket_31 = np.sqrt(omega3 * omega1) * (zbar3 - zbar1)

        # Parke-Taylor-like structure
        amplitude = bracket_12**4 / (bracket_12 * bracket_23 * bracket_31)

    else:
        amplitude = 0  # Other helicity configurations

```

```
    # Momentum conservation (delta function)
    # Handled by integration measure

    return amplitude


def celestial_3pt_graviton(Delta1, Delta2, Delta3, z1, z2, z3,
                           zbar1, zbar2, zbar3, helicities):
    """
    Celestial 3-point graviton amplitude

    After Mellin transform, this should have pure conformal structure:
    A_tilde ~ C_123 / ((z12)^{a} (z23)^{b} (z31)^{c})
    with exponents determined by conformal weights.
    """
    def amplitude_func(omegas, z_data):
        omega1, omega2, omega3 = omegas
        return graviton_3pt_momentum(
            omega1, omega2, omega3,
            z1, z2, z3, zbar1, zbar2, zbar3,
            helicities
        )

    return mellin_transform(
        amplitude_func,
        [omega1, omega2, omega3],
        [Delta1, Delta2, Delta3],
        [(z1, zbar1), (z2, zbar2), (z3, zbar3)]
    )
```

Listing 2: Celestial 3-point graviton amplitude

Analysis Note

Expected Structure: The celestial 3-point amplitude should take the form:

$$\tilde{A}_3 = C(\Delta_i) \cdot \frac{1}{z_{12}^{\Delta_1+\Delta_2-\Delta_3} z_{23}^{\Delta_2+\Delta_3-\Delta_1} z_{31}^{\Delta_3+\Delta_1-\Delta_2}} \tag{15}$$

Verify this structure numerically as a consistency check.

4.2 Phase 2: Conformal Block Decomposition (Months 2–4)

Goal: Implement celestial conformal blocks and OPE decomposition.

4.2.1 Celestial Conformal Blocks

```
from mpmath import hyp2f1

def celestial_conformal_block(Delta, z, zbar, Delta_ext):
    """
    Conformal block for celestial CFT

    For 2D CFT on the sphere, conformal blocks are hypergeometric:
    G_Delta(z, zbar) = z^{Delta/2} zbar^{Delta/2} F(Delta, ..., z) F(..., zbar)

    For celestial CFT with continuous spectrum, use principal series.

    Args:
        Delta: Internal conformal dimension (can be complex: 1 + i*lambda)
```

```python
        z, zbar: Cross-ratio coordinates
        Delta_ext: External dimensions [Delta_1, Delta_2, Delta_3, Delta_4]

    Returns:
        Conformal block value
    """
    Delta1, Delta2, Delta3, Delta4 = Delta_ext

    # Holomorphic and anti-holomorphic parts
    h = Delta / 2  # Holomorphic weight
    hbar = Delta / 2  # Anti-holomorphic weight

    # For scalar external operators
    a = (Delta1 - Delta2) / 2
    b = (Delta3 - Delta4) / 2

    # Hypergeometric function for holomorphic block
    # g_h(z) = z^h * 2F1(h-a, h+b; 2h; z)
    g_h = z**h * float(hyp2f1(h - a, h + b, 2*h, z))
    g_hbar = zbar**hbar * float(hyp2f1(hbar - a, hbar + b, 2*hbar, zbar))

    return g_h * g_hbar


def conformal_block_series(Delta, z, zbar, Delta_ext, order=20):
    """
    Conformal block via series expansion for better numerics

    Uses the Zamolodchikov recursion relation for efficiency.
    """
    # Implementation of recursion
    # G_Delta = z^Delta * sum_n c_n(Delta) z^n
    pass
```

Listing 3: Celestial conformal blocks

4.2.2 OPE Decomposition

```python
import numpy as np
from scipy.optimize import minimize

def ope_decomposition(celestial_4pt, z_grid, zbar_grid,
                      Delta_spectrum, Delta_ext):
    """
    Decompose celestial 4-point function into conformal blocks

    A_tilde_4 = sum_Delta C^2(Delta) G_Delta(z, zbar)

    Args:
        celestial_4pt: Function or array of 4-point amplitude values
        z_grid, zbar_grid: Grid of cross-ratio values
        Delta_spectrum: List of conformal dimensions to include
        Delta_ext: External operator dimensions

    Returns:
        Dictionary {Delta: C^2} of OPE coefficients squared
    """
    n_points = len(z_grid)
    n_ops = len(Delta_spectrum)

    # Build matrix of conformal blocks
    G_matrix = np.zeros((n_points, n_ops), dtype=complex)
```

```python
    for i, (z, zbar) in enumerate(zip(z_grid, zbar_grid)):
        for j, Delta in enumerate(Delta_spectrum):
            G_matrix[i, j] = celestial_conformal_block(
                Delta, z, zbar, Delta_ext
            )

    # Amplitude values at grid points
    A_values = np.array([celestial_4pt(z, zbar)
                         for z, zbar in zip(z_grid, zbar_grid)])

    # Solve for OPE coefficients: A = G @ C^2
    # Use non-negative least squares for unitarity
    from scipy.optimize import nnls
    C_squared, residual = nnls(G_matrix.real, A_values.real)

    return {Delta: c2 for Delta, c2 in zip(Delta_spectrum, C_squared)}
```

Listing 4: OPE decomposition of celestial amplitudes

4.3 Phase 3: Crossing Equations and Soft Theorems (Months 4–6)

Goal: Formulate crossing symmetry and soft theorem constraints.

4.3.1 Crossing Equation

```python
def crossing_equation(ope_data_s, ope_data_t, z, zbar, Delta_ext):
    """
    Verify/impose s-channel = t-channel OPE

    sum_{Delta_s} C_s^2(Delta_s) G_{Delta_s}(z, zbar)
        = sum_{Delta_t} C_t^2(Delta_t) G_{Delta_t}(1-z, 1-zbar)

    Args:
        ope_data_s: Dict {Delta: C^2} for s-channel
        ope_data_t: Dict {Delta: C^2} for t-channel
        z, zbar: Cross-ratio point
        Delta_ext: External dimensions

    Returns:
        Crossing residual (should be ~ 0)
    """
    # s-channel sum
    lhs = sum(C2 * celestial_conformal_block(Delta, z, zbar, Delta_ext)
              for Delta, C2 in ope_data_s.items())

    # t-channel sum (note: z -> 1-z)
    rhs = sum(C2 * celestial_conformal_block(Delta, 1-z, 1-zbar, Delta_ext)
              for Delta, C2 in ope_data_t.items())

    return abs(lhs - rhs)


def verify_crossing_symmetry(ope_data, z_grid, tol=1e-6):
    """
    Verify crossing symmetry at multiple points
    """
    violations = []
    for z, zbar in z_grid:
        residual = crossing_equation(ope_data, ope_data, z, zbar, ...)
        if residual > tol:
            violations.append((z, zbar, residual))
```

```python

    return len(violations) == 0, violations
```

Listing 5: Crossing symmetry verification

4.3.2 Soft Theorem Constraints

```python
def impose_soft_theorem(ope_data, Delta_soft=0):
    """
    Impose Weinberg soft graviton theorem

    In celestial basis, as Delta -> 0:
    A_tilde(Delta) ~ S^(0)/Delta + S^(1)/Delta^2 + ...

    S^(0) is the leading soft factor (sum of epsilon.p_i / p.p_i)

    Args:
        ope_data: Current OPE data
        Delta_soft: Small conformal weight (approaching soft limit)

    Returns:
        Constraint residual
    """
    # Extract behavior near Delta = 0
    def celestial_amplitude_near_soft(Delta, z, zbar):
        # Reconstruct amplitude from OPE
        return sum(C2 * celestial_conformal_block(Delta_int, z, zbar, ...)
                   for Delta_int, C2 in ope_data.items())

    # Check pole structure
    epsilon = 1e-4
    A_eps = celestial_amplitude_near_soft(epsilon, z_test, zbar_test)

    # Leading pole: should go like 1/Delta
    residue_0 = A_eps * epsilon

    # Compare to Weinberg soft factor
    weinberg_factor = compute_weinberg_soft(z_test, zbar_test, ...)

    return abs(residue_0 - weinberg_factor)


def compute_weinberg_soft(z, zbar, external_data):
    """
    Compute Weinberg soft graviton factor

    S^(0) = sum_i (epsilon.p_i) / (p_soft.p_i)

    For celestial coordinates, this becomes a sum over external
    operators weighted by their positions on the celestial sphere.
    """
    S0 = 0
    for i, (z_i, zbar_i, Delta_i) in enumerate(external_data):
        # Soft factor contribution from particle i
        S0 += (zbar - zbar_i) / ((z - z_i) * (zbar - zbar_i))

    return S0
```

Listing 6: Soft graviton theorem implementation

4.4 Phase 4: Bootstrap SDP (Months 6–9)

Goal: Formulate and solve the celestial bootstrap as a semidefinite program.

```python
import cvxpy as cp
import numpy as np

def setup_celestial_bootstrap_sdp(Delta_gap, Delta_max,
                                   z_grid, Delta_ext):
    """
    Set up SDP for celestial CFT bootstrap

    Variables: OPE coefficients C^2(Delta) for Delta in spectrum
    Constraints:
        1. Positivity: C^2(Delta) >= 0 (unitarity)
        2. Crossing symmetry: sum C^2 G_s = sum C^2 G_t
        3. Soft theorem: correct pole structure
        4. Normalization

    Args:
        Delta_gap: Minimum conformal dimension (gap assumption)
        Delta_max: Maximum dimension to include
        z_grid: Cross-ratio grid for sampling constraints
        Delta_ext: External operator dimensions

    Returns:
        cvxpy Problem object
    """
    # Discretize spectrum above the gap
    # For continuous spectrum: use Gauss quadrature points
    Delta_spectrum = np.linspace(Delta_gap, Delta_max, num=100)
    n_ops = len(Delta_spectrum)

    # Variables: OPE coefficients squared
    C_squared = cp.Variable(n_ops, nonneg=True)

    constraints = []

    # 1. Positivity: automatic from nonneg=True

    # 2. Crossing symmetry at each z point
    for z, zbar in z_grid:
        # s-channel blocks
        G_s = np.array([celestial_conformal_block(D, z, zbar, Delta_ext)
                        for D in Delta_spectrum])
        # t-channel blocks
        G_t = np.array([celestial_conformal_block(D, 1-z, 1-zbar, Delta_ext)
                        for D in Delta_spectrum])

        # Crossing: sum C^2 G_s = sum C^2 G_t
        constraints.append(G_s.real @ C_squared == G_t.real @ C_squared)
        constraints.append(G_s.imag @ C_squared == G_t.imag @ C_squared)

    # 3. Soft theorem (simplified: fix coefficient at Delta near 0)
    # constraints.append(soft_theorem_constraint(C_squared, ...))

    # 4. Normalization: fix identity contribution
    constraints.append(C_squared[0] == 1)  # Identity operator

    return C_squared, constraints, Delta_spectrum


def solve_celestial_bootstrap(Delta_gap):
    """
```

```
    Solve for consistent celestial CFT with given gap

    Returns:
        Dictionary with status and OPE data (if feasible)
    """
    C_squared, constraints, spectrum = setup_celestial_bootstrap_sdp(
        Delta_gap, Delta_max=10, z_grid=generate_z_grid(20),
        Delta_ext=[1, 1, 1, 1]  # External gravitons
    )

    # Feasibility problem
    problem = cp.Problem(cp.Minimize(0), constraints)

    try:
        problem.solve(solver=cp.SCS, verbose=True)

        if problem.status == cp.OPTIMAL:
            ope_data = {D: c2 for D, c2 in zip(spectrum, C_squared.value)
                        if c2 > 1e-6}
            return {'status': 'feasible', 'ope_data': ope_data}
        else:
            return {'status': 'infeasible',
                    'dual_certificate': extract_dual(constraints)}

    except Exception as e:
        return {'status': 'error', 'message': str(e)}
```

Listing 7: Celestial bootstrap SDP formulation

Critical Consideration

Continuous Spectrum Challenge: Unlike typical CFT bootstrap where the spectrum is discrete, celestial CFT has a *continuous* spectrum (principal series). The SDP must be discretized carefully, and results should be checked for sensitivity to discretization.

4.5 Phase 5: Extract Results and Certificates (Months 9–12)

Goal: Map allowed regions and generate verifiable certificates.

```
def scan_celestial_cft_space(Delta_gap_range, Delta_ext):
    """
    Scan over gap assumptions and map allowed vs. forbidden regions

    Args:
        Delta_gap_range: Range of gap values to test
        Delta_ext: External operator dimensions

    Returns:
        Results dictionary with phase diagram data
    """
    results = {}

    for Delta_gap in Delta_gap_range:
        print(f"\nTesting Delta_gap = {Delta_gap}")

        result = solve_celestial_bootstrap(Delta_gap)
        results[Delta_gap] = result

        if result['status'] == 'feasible':
            print(f"  FEASIBLE: Found consistent OPE data")
            # Verify and export
            verify_celestial_cft(result['ope_data'])
```

```
            export_ope_data(result['ope_data'], f'celestial_gap{Delta_gap}.json
    ')

        elif result['status'] == 'infeasible':
            print(f"  INFEASIBLE: No solution exists")
            # Export extremal functional
            export_extremal_functional(
                result['dual_certificate'],
                f'extremal_gap{Delta_gap}.json'
            )

    # Generate phase diagram
    plot_celestial_phase_diagram(results)

    return results
```

Listing 8: Scanning celestial CFT space

5 Detailed Research Directions

5.1 Direction 1: MHV Sector Bootstrap

Research Direction

Simplification: Focus on the MHV (Maximally Helicity Violating) sector where graviton amplitudes take the simplest form.
Approach:

1. Compute celestial MHV amplitudes explicitly for 3, 4, 5 particles
2. Extract OPE data from these known amplitudes
3. Bootstrap: can we *derive* the MHV amplitudes from crossing + soft theorems alone?

Expected Outcome: Demonstrate that celestial bootstrap reproduces known graviton scattering in the MHV sector.

5.2 Direction 2: Subleading Soft Theorems

Beyond the leading Weinberg soft factor, there are **subleading** soft theorems:

$$\tilde{A}(\Delta) = \frac{S^{(0)}}{\Delta} + S^{(1)} + \Delta\, S^{(2)} + O(\Delta^2) \tag{16}$$

Research Direction

Investigation: How do subleading soft theorems constrain the celestial OPE?
These are related to:

- BMS supertranslations ($S^{(0)}$)
- Superrotations ($S^{(1)}$)
- Higher memory effects ($S^{(2)}$)

5.3 Direction 3: Celestial OPE and Collinear Limits

When two particles become collinear, the celestial amplitude should factorize:

$$\lim_{z_1 \to z_2} \tilde{A}_n \sim \sum_{\Delta} \frac{C^{\Delta}_{\Delta_1 \Delta_2}}{(z_1 - z_2)^{\Delta_1 + \Delta_2 - \Delta}} \tilde{A}_{n-1} \tag{17}$$

Research Direction

Study: Extract celestial OPE coefficients from known collinear limits of graviton amplitudes. Compare with bootstrap predictions.

5.4 Direction 4: Gluon Amplitudes and Yang-Mills

Celestial holography applies beyond gravity:

- **Gluon amplitudes** map to a celestial CFT with different structure
- **Color structure** introduces new OPE channels
- **Soft gluon theorem** differs from graviton case

Research Direction

Extension: Develop celestial bootstrap for Yang-Mills theory. Compare and contrast with gravity.

5.5 Direction 5: Loop Corrections and Infrared Divergences

Loop amplitudes have **infrared divergences** that affect celestial amplitudes:

- Soft and collinear divergences modify the Mellin transform
- May require IR regularization (dimensional regularization, mass regulator)
- Celestial interpretation of IR physics is an active research area

6 Success Criteria

6.1 Minimum Viable Result (6 months)

✓ Mellin transform engine working for tree-level graviton amplitudes

✓ 3-point and 4-point celestial amplitudes computed

✓ $\mathrm{SL}(2, \mathbb{C})$ covariance verified numerically

✓ Soft theorems checked

✓ **First bootstrap result:** Crossing equation setup for MHV sector

✓ Either: Allowed OPE data found, or: no-go region certified

6.2 Strong Result (9 months)

- ✓ Multi-channel bootstrap including all helicity sectors
- ✓ Soft + subleading soft theorems incorporated
- ✓ Rigorous allowed region in OPE space determined
- ✓ Or: Exclusion of certain conformal weight ranges proved
- ✓ Extremal functionals extracted and verified

6.3 Publication-Quality Result (12 months)

- ✓ Comprehensive celestial CFT space mapped for graviton scattering
- ✓ Phase diagram of consistent theories
- ✓ Novel predictions or no-go theorems
- ✓ Formal verification of key results
- ✓ Comparison with explicit amplitude calculations

7 Verification Protocol

```python
def verify_celestial_cft(ope_data):
    """
    Comprehensive verification of celestial CFT solution

    Checks:
    1. SL(2,C) covariance of reconstructed amplitudes
    2. Crossing symmetry
    3. Soft graviton theorem
    4. Unitarity (positivity)

    Args:
        ope_data: Dictionary {Delta: C^2} of OPE coefficients

    Returns:
        "VERIFIED" if all checks pass
    """
    # 1. Verify SL(2,C) covariance
    for transformation in sl2c_generators():
        transformed = apply_lorentz_transformation(ope_data, transformation)
        if not is_equivalent(transformed, ope_data):
            return "FAILED: SL(2,C) covariance violated"

    # 2. Check crossing symmetry at multiple points
    z_test_points = generate_crossing_test_points(n=50)
    for z, zbar in z_test_points:
        s_channel = ope_sum(ope_data, 's', z, zbar)
        t_channel = ope_sum(ope_data, 't', z, zbar)
        if abs(s_channel - t_channel) > 1e-8:
            return f"FAILED: Crossing violated at z={z}"

    # 3. Verify soft theorem
    soft_behavior = extract_soft_limit(ope_data, Delta_soft=1e-3)
    weinberg_soft = compute_weinberg_soft(...)
    if not is_close(soft_behavior, weinberg_soft, rtol=1e-6):
        return "FAILED: Soft theorem violated"
```

```python

    # 4. Unitarity (positive spectral density)
    for Delta, C2 in ope_data.items():
        if C2 < -1e-10:
            return f"FAILED: Negative OPE coefficient at Delta={Delta}"

    return "VERIFIED"
```

Listing 9: Comprehensive celestial CFT verification

8 Common Pitfalls and Mitigations

8.1 Mellin Transform Convergence

Critical Consideration

Problem: The Mellin integral $\int_0^\infty d\omega\, \omega^{\Delta-1} A(\omega)$ may not converge for all Δ.
Solutions:

- Work on the principal series: $\Delta = 1 + i\lambda$ with $\lambda \in \mathbb{R}$
- Use analytic continuation from convergent region
- Regularize with exponential damping: $\omega^{\Delta-1} e^{-\epsilon\omega}$

8.2 Continuous vs. Discrete Spectrum

Critical Consideration

Problem: Standard bootstrap techniques assume discrete spectrum; celestial CFT has continuous principal series.
Solutions:

- Discretize carefully using Gauss quadrature adapted to the measure
- Check results are stable under discretization refinement
- Use functional analysis techniques for continuous bootstrap

8.3 Shadow Operators and Redundancy

Critical Consideration

Problem: Celestial operators $\mathcal{O}_\Delta$ and their shadows $\mathcal{O}_{2-\Delta}$ are related, leading to redundancy in the OPE.
Solution:

- Impose shadow symmetry as an additional constraint
- Or restrict to $\mathrm{Re}(\Delta) \geq 1$ (half the spectrum)
- Document the convention clearly

9 Milestone Checklist

9.1 Phase 1: Amplitude Infrastructure (Months 1–2)

- ☐ Mellin transform calculator implemented
- ☐ 3-point celestial amplitude computed
- ☐ 4-point MHV celestial amplitude computed
- ☐ $\mathrm{SL}(2, \mathbb{C})$ covariance verified numerically
- ☐ Spinor-helicity formalism working

9.2 Phase 2: Conformal Blocks (Months 2–4)

- ☐ Celestial conformal blocks implemented
- ☐ Principal series representations handled correctly
- ☐ OPE decomposition working for test cases
- ☐ Cross-ratio dependence verified

9.3 Phase 3: Constraints (Months 4–6)

- ☐ Crossing equations formulated
- ☐ Soft theorem constraints implemented
- ☐ Subleading soft theorems added
- ☐ Combined constraint system tested

9.4 Phase 4: Bootstrap Solver (Months 6–9)

- ☐ SDP formulation complete
- ☐ Solver running and converging
- ☐ First feasibility/infeasibility result
- ☐ Gap scan initiated

9.5 Phase 5: Results & Verification (Months 9–12)

- ☐ Allowed/forbidden regions mapped
- ☐ Extremal functionals extracted
- ☐ Results verified against known amplitudes
- ☐ Publication draft prepared

10 Resources and References

10.1 Foundational Papers

1. Pasterski, Shao, Strominger (2017): "Flat Space Amplitudes and Conformal Symmetry of the Celestial Sphere" [arXiv:1701.00049]

2. Pasterski, Shao (2017): "Conformal Basis for Flat Space Amplitudes" [arXiv:1705.01027]

3. Strominger (2018): "Lectures on the Infrared Structure of Gravity and Gauge Theory" [arXiv:1703.05448]

4. Raclariu (2021): "Lectures on Celestial Holography" [arXiv:2107.02075]

5. Pasterski (2021): "Celestial Amplitudes" [TASI lectures]

10.2 Bootstrap References

1. Simmons-Duffin (2016): "The Conformal Bootstrap" [arXiv:1602.07982]

2. Poland, Rychkov, Vichi (2019): "The Conformal Bootstrap: Theory, Numerical Techniques, and Applications" [arXiv:1805.04405]

10.3 Software

- **mpmath:** Arbitrary precision — `pip install mpmath`
- **CVXPY:** Convex optimization — `pip install cvxpy`
- **SymPy:** Symbolic computation — `pip install sympy`
- **Spinor-Helicity packages:** Various implementations available

11 Conclusion

Celestial holography represents a bold new paradigm for understanding quantum gravity in the real, asymptotically flat universe. The celestial bootstrap approach—constraining the space of consistent celestial CFTs using crossing symmetry, unitarity, and soft theorems—offers a rigorous path toward classifying possible theories.

Success in this challenge would:

1. Establish the celestial bootstrap as a viable tool for constraining gravity

2. Produce novel predictions for graviton scattering (or prove certain structures impossible)

3. Connect the S-matrix bootstrap to the holographic program

4. Open new directions in flat-space holography

The interplay between scattering amplitudes, conformal field theory, and optimization provides a rich mathematical structure amenable to rigorous analysis and machine verification.