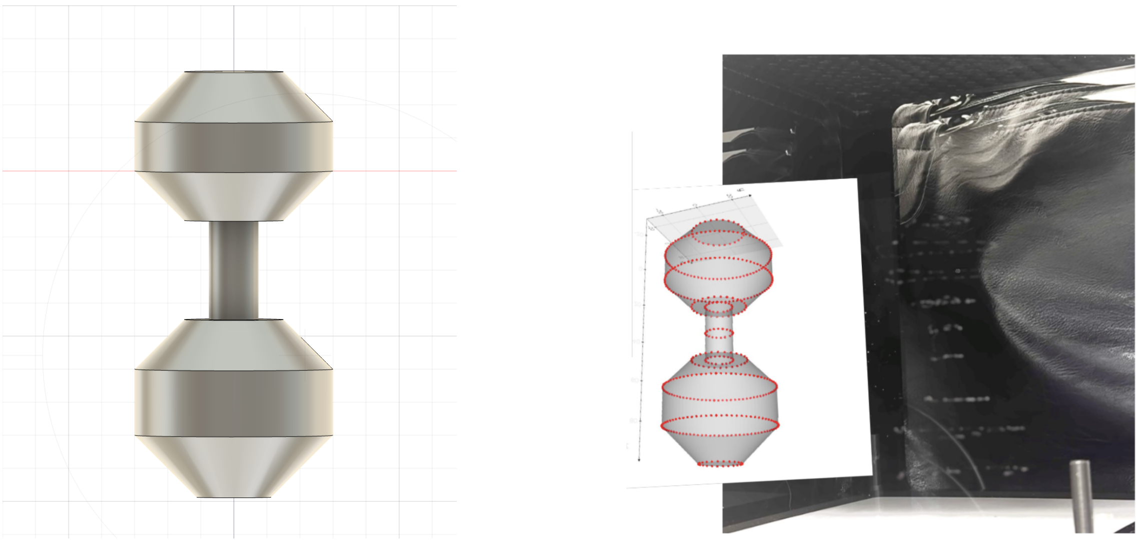
OptiSpline

August 2025

1 Abstract

Acoustic levitation uses high-frequency sound waves to trap and manipulate small particles in mid-air without any physical contact. By carefully shaping the pressure fields, particles can be suspended and moved along desired trajectories, enabling applications such as mid-air displays and contact-free manufacturing. Acoustic levitation enables ways to generate volumetric particle paths for mid-air display, however, designing trajectories that are efficient, smooth, and acoustically stable continues to be a challenge due to the trade-off between speed, smoothness and acoustic stability. Prior work such as OptiTrap introduced a numerical approach for nearly time-optimal trap trajectories from a reference shape, while StableLev applied machine learning to stabilise multiparticle levitation. However, existing methods don't always capture the role of complex acoustic environments and the Gor'kov potential, leading to limited path smoothness and levitation stability. We present OptiSpline, a computational framework that optimises B-spline control points while incorporating smoothness, curvature and acoustic forces via the Gor'kov potential. Our method employs mini-batch spline evaluation and a gradient-based solver, using derivatives of the Gor'kov potential with respect to spline control points, enabling optimisation over hundreds of sampled points per iteration. OptiSpline achieves smooth trajectories suitable for real-time levitation and compares its performance to linearly interpolated paths. Tested across multiple target shapes, OptiSpline outperforms linear interpolation by generating smoother trajectories with fewer oscillations, higher success rates, and higher execution speeds.



(a) Dumbbell STL

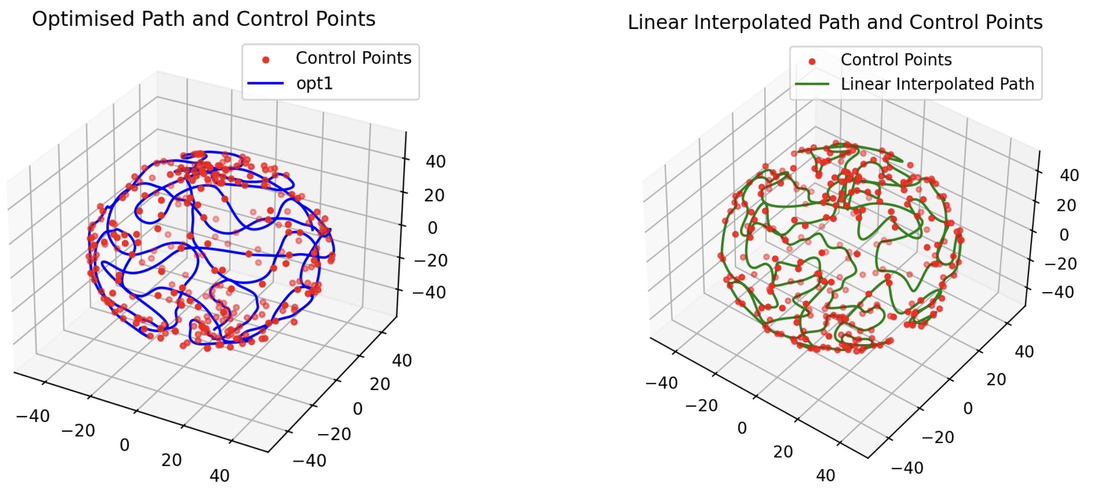
(b) Dumbbell levitated path

Figure 1: Final result of OptiSpline levitation of a dumbbell shape

2 Introduction

Acoustic levitation uses sound waves to trap and manipulate small polystyrene particles in mid-air, allowing the creation of dynamic, three-dimensional paths that are the basis for mid-air displays. Using the Persistence of Vision (PoV) effect, acoustic levitation can create volumetric content in mid-air. This technique rapidly moves acoustically trapped particles along periodic paths, forming shapes that are visible and perceptible to the human eye. For the most optimal result, these paths must not only follow a desired shape but also satisfy strict physical constraints. A key difficulty arises from the inherent trade-offs in trajectory design: paths that are very fast often involve sharp turns, which require rapid changes in acceleration. The acoustic trap can only exert limited forces on the particle, so if the particle needs more force than the trap can provide to follow the turn, it becomes destabilised. On the other hand, very smooth or slow paths preserve stability but limit display quality and thus the ability to accurately recognise the shape. Therefore, the main objective is to achieve trajectories that are relatively fast, smooth, and acoustically stable.

OptiSpline represents trajectories using B-splines, which are suited for generating smooth paths due to their piecewise polynomial nature and local control of curvature. Unlike linear interpolation, which produces jagged transitions, spline representations allow continuous control of both first and second derivatives, making them inherently smoother [Figure 2]. The control points are optimised with an objective that balances smoothness, curvature, and the acoustic forces derived from the Gor'kov potential. This integration ensures that both geometric and physical considerations are taken into account during trajectory design.



(a) Opti-B-Spline (b) Linear interpolation

Figure 2: Visual comparison of smoothness of linear and optimised paths

Instead of evaluating the full trajectory at every iteration, a mini-batch sampling strategy is employed in which only a subset of spline points is considered per optimisation step. This reduces computational time while still capturing the global structure of the path across iterations. Furthermore, a gradient-based solver computes how small adjustments to spline control points affect both trajectory smoothness and Gor'kov potential values. Using these derivatives, OptiSpline can update control points directly along directions, allowing us to process hundreds of sampled points per iteration.

The effectiveness of OptiSpline is demonstrated through comparative experiments with linear interpolation. While linear interpolation is simple and computationally uncomplicated, it produces paths with discontinuous derivatives that lead to sharp transitions in velocity and acceleration. These discontinuities reduce levitation stability and introduce oscillations in the particle trajectory. OptiSpline, on the other hand, produces continuous and responsive paths.

To validate this, OptiSpline and linear interpolation paths were tested across a range of target shapes with varying curvature and complexity. In most cases, OptiSpline produced trajectories that were smoother and exhibited fewer oscillations compared to those obtained by linear interpolation. The improved smoothness led to higher success rates, with particles remaining trapped and following the intended path more reliably.

3 Related work

Trajectory optimisation for acoustic levitation has been explored from multiple perspectives, each addressing certain aspects of the problem but revealing new challenges. Previous efforts, such as OptiTrap [Paneva(2022)], aimed to generate trajectories that closely follow a reference path while minimising traversal time. By treating the particle paths as a sequence of discrete points and optimising their positions, OptiTrap successfully produced smooth and nearly time-optimal trajectories. However, because it did not explicitly penalise curvature or acceleration, some of the resulting paths contained sharp turns. These sharp turns require rapid changes in acceleration that can exceed the forces the acoustic trap can apply, destabilising the levitated particles.

To tackle stability more directly, StableLev [Gao(2024)] introduced a reinforcement learning–based approach to actively stabilise multiparticle levitation. By continuously monitoring particle states and learning from past interactions, StableLev effectively mitigates issues such as drift, oscillations, and collisions, providing robustness in multiparticle scenarios. However, this method primarily focuses on controlling particle behaviour rather than explicitly optimising the smoothness or efficiency of single-particle trajectories. Consequently, while StableLev solves one aspect of the problem, which is robustness, it does not fully address the need for smooth, efficient paths suitable for high-quality mid-air displays.

Meanwhile, in fields such as robotics, computer graphics, and medicine, spline representations, particularly B-splines, have long been used for path design due to their inherent smoothness and local control [Chaudhuri(2021)]. B-splines allow designers to adjust trajectory segments independently, ensuring continuity in velocity and acceleration. This property directly addresses the sharp-turn problem observed in OptiTrap, as it enables smoother paths that reduce abrupt accelerations and improve stability. Spline-based optimisation has been successfully applied in robotics and UAV path planning, often minimising objectives such as curvature, snap, or energy consumption. But, despite their potential, spline methods have not been applied to acoustic levitation, where the nonlinear Gor'kov potential introduces additional constraints [Figure 3]. Ignoring these forces can produce trajectories that are efficient in theory but unstable in practice, as even small deviations can lead to particle oscillations or loss of trapping.

Overall, these studies illustrate an evolving landscape of trajectory design strategies. Time-optimal methods improve efficiency but risk destabilising particles with sharp turns. Reinforcement learning enhances robustness but does not prioritise smooth, efficient trajectories and spline-based approaches offer smooth, physically realistic paths, but have yet to incorporate the underlying acoustic physics. This progression motivates the development of OptiSpline, which combines spline optimisation with physics-aware constraints to generate trajectories that are fast, smooth, and acoustically stable.

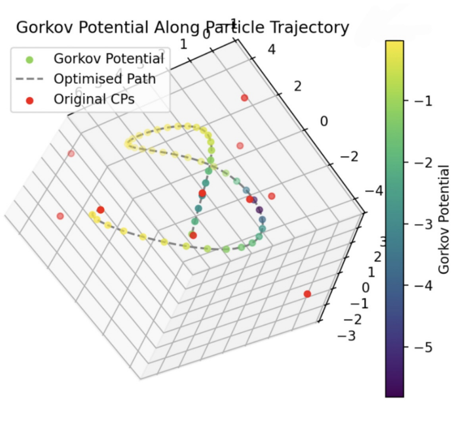


Figure 3: Gor'kov Potential Along an Arbitrary Spline

4 Methodology

4.1 Bézier curves

Splines are widely used in trajectory generation because they provide smooth, continuous paths with controllable curvature and derivative properties [Figure 4]. In the context of acoustic levitation, these properties are essential: abrupt changes in curvature or acceleration can make the particle less stable. Splines offer a good mathematical framework for constructing feasible paths. Bézier curves form mathematical foundation of spline theory.

A Bézier curve of degree n is defined parametrically as

$$B(t) = \sum_{i=0}^{n} b_i B_{i,n}(t), \quad t \in [0, 1]. \tag{1}$$

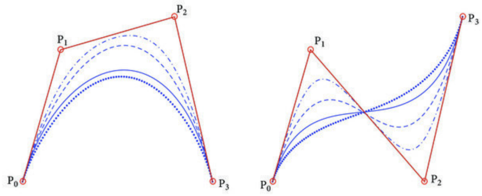


Figure 4: Cubic Bézier curve [Zhu and Liu(2019)]

where b are control points and B(t) are Bernstein polynomials [Farin(2001)]. The curve is strongly influenced by its control points but does not generally pass through them, except for the first and last points. This makes Bézier curves approximation-based rather than interpolating. They are globally smooth and provide predictable shape behaviour, but suffer from local control, which means that a small change to one control point can significantly alter the rest of the trajectory.

Since Bézier curves offer no local control, they are often regarded as a starting point that naturally leads to more flexible spline variants.

4.2 B-splines and Catmull-Rom splines

A B-spline is defined as a piecewise polynomial curve expressed in terms of basis functions over a knot vector:

$$S(t) = \sum_{i=0}^{n} N_{i,k}(t) b_i \tag{2}$$

where $N_{i,k}(t)$ are B-spline basis functions of degree k, and b_i are control points. Unlike Bézier curves, B-splines allow local control: moving one control point only affects a limited region of the curve.

Catmull-Rom splines are interpolating curves, meaning they pass directly through all control points. A segment between points P_i and P_{i+1} is defined as:

$$C(t) = 0.5\Big[(2P_i)+(-P_{i-1}+P_{i+1})t+(2P_{i-1}-5P_i+4P_{i+1}-P_{i+2})t^2+(-P_{i-1}+3P_i-3P_{i+1}+P_{i+2})t^3\Big]. \tag{3}$$

They are particularly useful when exact points must be visited, though this comes at the cost of reduced control over curvature.

For trajectory planning, the smoothness of the curve is crucial. Continuity is commonly classified as:

- C^0 continuity: the curve is connected continuously, with no gaps.
- C^1 continuity: the first derivative is continuous with smooth tangents, which prevents abrupt changes in velocity.
- C^2 continuity: the second derivative is continuous, providing smooth curvature and avoiding sudden acceleration changes.

B-splines naturally allow C^{k-1} continuity for degree-k polynomials, making them highly suitable for generating smooth trajectories for levitated particles. In contrast, Catmull-Rom splines guarantee interpolation through points (C^1 continuous by default) but may have less control over higher-order derivatives, which can lead to sharper curvature changes at control points.

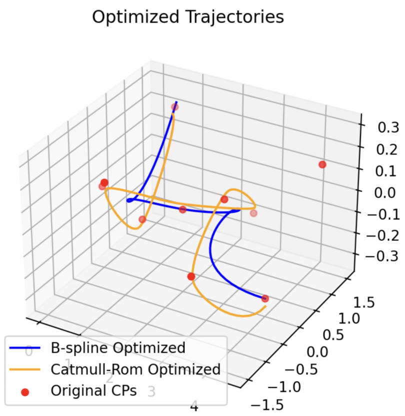


Figure 5: B-spline vs. Catmull-Rom spline

Bézier and B-splines provide global smoothness with C^1 or C^2 continuity, whereas Catmull-Rom splines guarantee interpolation but may suffer from sharper turns [Figure 5]. B-splines offer the best balance between smoothness and local control, making them suitable for adaptive trajectory generation. Bézier curves are simple but lack flexibility for complex paths. Catmull-Rom splines are useful for interpolating pre-defined waypoints but may introduce instability in levitation due to high curvature at sharp control point transitions. That is why B-splines was the final choice for the path generation.

4.3 Path generation

4.3.1 Control points

Control points for the trajectory were not manually selected; instead, they are automatically extracted from the target surface geometry, which is provided as an STL (Stereolithography) mesh (e.g., spheres, tori, cylinders) [Figure 6]. To obtain control points suitable for trajectory planning, we first assemble all the vertices of the mesh into a raw point cloud

$$\{q_j\}_{j=1}^{N} \subset R^3, \tag{4}$$

where N is the total number of vertices in the mesh. For typical STL files representing common geometric shapes such as spheres, tori, or cylinders, N can range from several thousand to tens of thousands. Using all vertices directly would be computationally expensive and unnecessary for spline-based trajectory optimisation, as adjacent vertices are often very close to each other and contain redundant information.

Therefore, we uniformly sample a subset of N_s points from the raw point cloud, where typical values are N_s = 300–500. Uniform sampling ensures that the selected points are spread evenly over the surface, preserving the overall shape while reducing computational complexity. These sampled points form the set of reference locations that the levitated particle is intended to follow. By fitting a smooth spline through this set of points, we can generate a continuous trajectory that both accurately represents the target surface and satisfies the physical constraints of the acoustic levitation system.

To impose a trajectory-like ordering on this unordered set, the principal component analysis (PCA) with $n_{\text{comp}} = 1$ was applied. Denoting the first principal component by $u \in R^3$, each point is projected as $s_j = u^\top q_j$. Points are first sorted by s_j and then refined into a continuous chain by a greedy nearest-neighbour (NN) traversal: starting at the first element, the next point is the unused point with minimum Euclidean distance from the current location. This produces an ordered 1D sequence $\{p_i\}_{i=1}^{N}$ that approximates a continuous path over the surface while remaining simple (the greedy NN has $O(N^2)$ time in the worst case and works well in practice for these sizes).

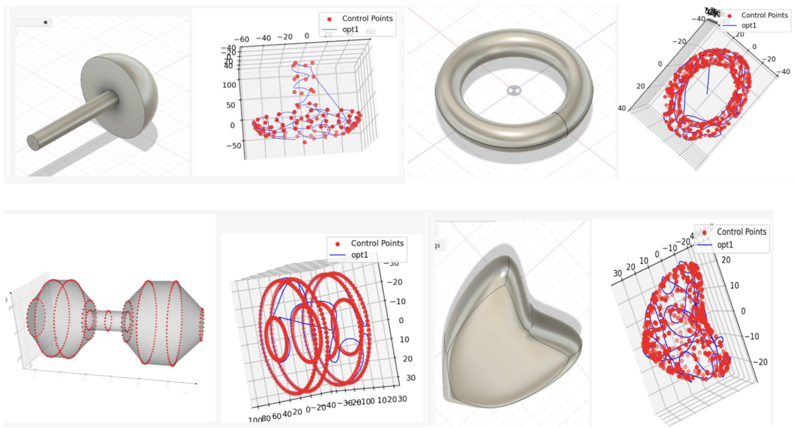


Figure 6: Control points generated from different STL shapes and approximated paths through them

Knot Vector Construction. Knot vectors determine how the curve transitions between segments. Given the ordered control points $\{p_i\}$ and a chosen spline degree k (we use $k = 3$), we construct a clamped, uniformly spaced knot vector

$$\boldsymbol{t} = \underbrace{(0, \dots, 0)}_{k+1}, \xi_1, \dots, \xi_M, \underbrace{(1, \dots, 1)}_{k+1}, \tag{5}$$

with $M = N - k - 1$ interior knots ξ_m placed uniformly in $(0, 1)$. Clamping ensures interpolation at the endpoints and provides C^{k-1} continuity at interior knots for B-splines without repeated knots.

B-spline Evaluation The B-spline curve is

$$S(t) = \sum_{i=1}^{N} N_{i,k}(t; \boldsymbol{t})\, p_i, \qquad t \in [t_k,\, t_{|\boldsymbol{t}|-k-1}], \tag{6}$$

where $N_{i,k}$ are degree-k basis functions on $\boldsymbol{t}$. We sample $S(t)$ at n_s points (n_s = 2000–3500) to obtain $\{x_\ell\}_{\ell=1}^{n_s}$.

4.3.2 Objective and Spline Optimisation

Control points are refined by minimising a composite objective

$$\mathcal{J} \;=\; \alpha\, \mathcal{L}_{\text{smooth}} \;+\; \beta\, \mathcal{L}_{\text{fit}} \;+\; \gamma\, \mathcal{L}_{\text{G}} \;+\; \delta\, \mathcal{L}_{\kappa}, \tag{7}$$

with scalar weights $\alpha, \beta, \gamma, \delta$. Two variants are used:

(1) a *geometry+Gor'kov* variant with α=100, β=10, γ=5×10^5, δ=1 (sparser sampling for $\mathcal{L}_{\text{G}}$), and

(2) a *geometry-only* variant with α=500, β=1, δ=1000 (and γ=0).

Each term in the objective serves a specific purpose:

- $\mathcal{L}_{\text{smooth}}$ (smoothness term): encourages the trajectory to be smooth by penalising large changes in velocity and acceleration between successive control points. This helps prevent sharp turns that could destabilise the levitated particle.
- $\mathcal{L}_{\text{fit}}$ (fidelity term): ensures that the trajectory follows the target geometry defined by the sampled points from the mesh. Minimising this term aligns the spline path closely with the desired 3D shape.
- $\mathcal{L}_{\text{G}}$ (Gor'kov term): accounts for the acoustic physics by penalising regions where the Gor'kov potential would lead to unstable levitation. Including this term helps maintain particle stability in the nonlinear acoustic field.
- $\mathcal{L}_{\kappa}$ (curvature term): penalises excessive curvature along the trajectory, limiting abrupt changes in direction that require high forces. This complements $\mathcal{L}_{\text{smooth}}$ by directly controlling geometric sharpness.

The weights $\alpha, \beta, \gamma, \delta$ allow the balancing these objectives.

Smoothness term $\mathcal{L}_{\text{smooth}}$. Let discrete velocity, acceleration, and jerk be

$$v_\ell = x_{\ell+1} - x_\ell, \qquad a_\ell = v_{\ell+1} - v_\ell, \qquad j_\ell = a_{\ell+1} - a_\ell.$$

We penalise dispersion in step lengths and higher-order differences:

$$\mathcal{L}_{\text{smooth}} \;=\; 0.1 \sum_\ell \left(\|v_\ell\| - \bar{\ell}\right)^2 \;+\; \sum_\ell \|a_\ell\|^2 \;+\; \sum_\ell \|j_\ell\|^2, \tag{8}$$

where $\bar{\ell} = \frac{1}{L}\sum_\ell \|v_\ell\|$ is the mean step length (computed with `no_grad` to stabilise optimisation). This promotes near-constant speed and low acceleration/jerk.

Geometric fit $\mathcal{L}_{\text{fit}}$. The purpose of this term is to encourage the spline trajectory to remain close to the target geometry specified by the mesh. Rather than requiring an explicit one-to-one correspondence between trajectory points and mesh vertices, we adopt a one-sided Chamfer-style distance. Specifically, each sampled point x_ℓ from the curve is penalised by its Euclidean distance to the nearest vertex in the target set $Y = \{y_m\}$. This gives

$$\mathcal{L}_{\text{fit}} \;=\; \frac{1}{n_s} \sum_{\ell=1}^{n_s} \min_{y \in Y} \|x_\ell - y\|_2, \tag{9}$$

where n_s is the number of spline samples.

Gor'kov term $\mathcal{L}_{\text{G}}$ To push the trajectory toward acoustically stable regions, we subsample the curve at points x_ℓ with stride $s = 10$ and evaluate the Gor'kov potential $G(x)$ using our acoustic model. The Gor'kov potential is computed from the acoustic pressure field, which we first obtain using a phase retrieval algorithm. To simulate the effect of levitation, we apply a levitation activation function to the pressure field. This activation highlights regions where the acoustic forces are strong enough to trap particles, effectively masking areas where levitation would be impossible. The activated field is then normalised so that the computed Gor'kov potential $G(x)$ reflects relative stability across the domain. Finally, we penalise the magnitude of $G(x)$ along the trajectory, discouraging paths that pass through regions of weak or unstable levitation.

$$\mathcal{L}_{\text{G}} \;=\; \sum_{\ell \in \mathcal{S}} \big|G(x_\ell)\big|. \tag{10}$$

In other words, the levitation activation ensures that only physically relevant regions, where the acoustic trap can actually hold a particle, contribute to the Gor'kov term.

Curvature term $\mathcal{L}_\kappa$ Discrete curvature is estimated via finite differences:

$$\kappa_\ell \;\approx\; \frac{\|a_\ell\|^2}{\|v_\ell\|^3 + \varepsilon}, \qquad \varepsilon = 10^{-6}, \qquad \mathcal{L}_\kappa = \sum_\ell \kappa_\ell. \tag{11}$$

This discourages tight bends that can destabilise levitation.

Solver Control points are optimised with a first-order method (learning rate 0.05–0.1, 30–100 iterations), using PyTorch autograd on the B-spline evaluation. The gradients are retained at curve samples when needed for diagnostics.

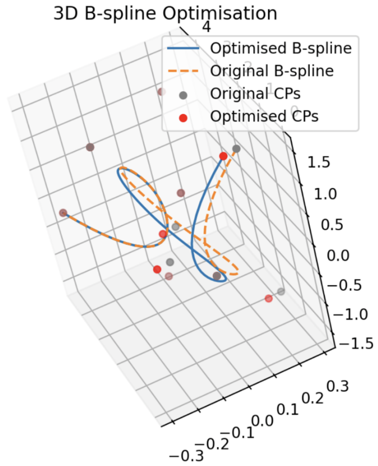


Figure 7: 3D B-Spline Optimisation

4.3.3 Smoothing

After optimisation, the sampling is densified where curvature is higher. This is needed for smoother and more stable turns. We compute κ_ℓ along the curve and define weights

$$w_\ell \;=\; 1 + \lambda\,\kappa_\ell, \qquad \lambda = 5.0,$$

then build a non-uniform parameterisation $t_\ell \propto \sum_{i\le\ell} w_i$ and resample each coordinate via 1D interpolation onto a uniform grid. Finally, a moving-average smoother is applied (window w=15) to reduce high-frequency jitter while preserving shape.

4.3.4 Prefix Motion

To prevent sudden jumps in the levitator, a smooth cosine-based ramp from the origin to the first spline point was added.

$$x_{\text{prefix}}(\tau) \;=\; (1-\eta(\tau))\,x_0 \;+\; \eta(\tau)\,x_1, \quad \eta(\tau) = \tfrac{1}{2}\big(1-\cos(\pi\tau)\big),\ \tau\in[0,1], \tag{12}$$

using $n_{\text{prefix}} = 1500$ samples. The final trajectory is formed by joining the cosine ramp with the optimised spline samples.

4.3.5 Discretisation and Controller Interface

Trajectories are produced as ordered 3D points on [mm] and scaled by a constant factor (1/1500) to match the levitator's workspace units used by the acoustic solver. This factor reflects the ratio between the typical physical size of the target meshes ($\sim$ 10−50 mm) and the unit domain used internally by the wave–geometry solver ($[-1,1]$ in each dimension). For example, a displacement of 1 mm in the mesh space becomes 1/1500 in solver units. This ensures that trajectories expressed in real-world millimetres are correctly mapped into the numerical domain of the acoustic field calculation, so that the subsequent activations correspond to physically realisable positions in the levitator. We set the frame rate to different values, which will later be used for testing the success of the levitated path, e.g. $f_{\text{cmd}} = 700$ Hz. For analysis, we also compute kinematic metrics: total path length, duration, average speed. Each trajectory point is mapped to an acoustic activation by (i) computing the WGS field at the query location, (ii) adding the levitation signal, (iii) normalising the drive, and (iv) streaming the transducer pattern to the hardware.

4.3.6 Continuity Considerations

The cubic ($k = 3$) B-splines with clamped, simple knots guarantee C^2 continuity at interior knots and C^0 continuity at the endpoints due to interpolation. Additional penalties on acceleration (a_ℓ) and jerk (j_ℓ) provide effective control over derivative continuity in the trajectory, preventing sudden velocity or acceleration changes that could destabilise the particle and push it out of the acoustic trap.

4.4 Assumptions and initial conditions

It was assumed a single particle is levitated, with no inter-particle interactions or multiparticle coupling. The levitation field is considered ideal and symmetric. The environmental disturbances such as airflow and temperature fluctuations are negligible. The acoustic pressure field is assumed to remain constant during trajectory execution. Experiments were conducted in still air, external forces such as airflow and vibration were neglected. Gravity was included in the model but assumed constant.

4.5 The particle and control objectives

The levitated test particle was a polystyrene bead with a radius of approximately 0.7-1 mm with particle density of $\rho \approx 1050$ kg/m^3. The main control objective is to move the particle smoothly between defined start and end positions along the generated spline trajectories while minimising destabilisation effects such as high curvature or rapid accelerations. Other objectives include maintaining particle within stable acoustic potential limits and avoiding excessive oscillations.

5 Experiments

The performance of the proposed spline-based trajectory optimisation was evaluated against a linear interpolation baseline across four representative shapes: sphere, double cylinder, torus, and four sphere configuration. When comparing both on the levitator, two main metrics were considered: maximum achievable frame rate (stability limit of path execution) and maximum possible speed (measured in mm/s). Each configuration was tested with 3,500 + 1,500 interpolation points.

5.1 Maximum Frame Rate

Each experiment for a given shape was repeated multiple times. The frame rate was initially set to 300 fps and then increased in increments of 100 fps until the trajectory could no longer be successfully levitated. Once failure was observed, the search was refined by adjusting the frame rate in smaller steps of 50 fps to identify the maximum feasible value.

Shape	**Linear Interpolation**	**Spline Optimisation**
Sphere	600	600
Double Cylinder	700	**1100**
Torus	700	**800**
Four spheres	Failed to stabilise	500

Table 1: Maximum achievable frame rate for each shape.

The spline-optimised trajectories showed higher stability limits in complex geometries. The four sphere configuration failed to stabilise under linear interpolation, whereas spline optimisation allowed stable execution at 500 fps.

5.2 Maximum Speed

The test for each shape was run 3 times and the mean number was then calculated.

Shape	**Linear (mm/s)**	**Spline (mm/s)**
Sphere	261.418	**283.418**
Double Cylinder	255.603	**265.603**
Torus	127.705	**159.467**
Four spheres	-	203.003

Table 2: Maximum achievable speed for each shape.

The speed was measured at a frame rate of 600 fps. For all shapes, spline optimisation produced higher achievable speeds, particularly in the complex torus case (**+25% improvement**). Since the four sphere configuration failed to stabilise at all under the frame rate of 600 fps, the results are missing.

5.3 Success Rate

Across repeated trials, spline optimisation demonstrated a higher overall success rate (**4/5**) compared to linear interpolation (**3/5**).

5.4 Trajectory Metrics Comparison

In addition to frame rate and speed, we compared the spline-optimised and linear interpolation trajectories on a single-sphere path in terms of distance travelled, maximum velocity, acceleration peaks, smoothness, curvature, and computation time. These metrics indicate trajectory quality and highlight changes that could destabilise the particle and push it out of the acoustic trap.

Metric	**Spline Optimisation**	**Linear Interpolation**
Distance travelled [mm]	1780.41	2181.64
Maximum velocity [mm/s]	2.75	3.09
Maximum acceleration [mm/s^2]	0.46	3.10
Smoothness (jerk^2)	3.71	43.80
Curvature measure	542.54	370.17
Computation time [s]	9.47	0.0017

Table 3: Comparison of trajectory metrics between spline-optimised and linear paths. Note that spline results do not include Gor'kov potential, which would further increase computation time.

The spline-optimised trajectory achieves significantly smoother motion, with lower maximum velocity and acceleration peaks compared to the linear path. This indicates reduced dynamic stress and a lower likelihood of particle ejection from the acoustic trap. The curvature is higher in the spline path, reflecting more flexible adaptation to the target shape, whereas the linear trajectory exhibits abrupt changes that increase the risk of destabilisation. Despite these advantages, the spline trajectory requires substantially longer computation time (9.47 s) compared to the almost instantaneous linear interpolation (0.0017 s). The linear path travels a longer distance and has higher velocity and acceleration, which may push the particle closer to the stability limits. Overall, spline optimisation offers better physical feasibility and smoother control, while linear interpolation remains valuable for quick real-time applications.

6 Discussion

The results suggest that OptiSpline generally offers advantages over simple linear interpolation, particularly in terms of particle stability and motion smoothness [Figure 8]. At first glance this might seem unsurprising, splines are designed to remove discontinuities, but in the context of acoustic levitation this becomes especially significant. Even small instabilities can cause a particle to drift out of the Gor'kov potential minimum, so the fact that spline-based trajectories reduce sudden accelerations directly translates into more reliable levitation. This reliability could make the difference between a mid-air display that appears seamless versus one that constantly "drops" particles. In that sense, OptiSpline is not just a minor incremental improvement but addresses a practical barrier for adoption of levitation in applications such as interactive graphics.

The use of B-splines proved particularly effective because of their ability to enforce smooth changes in velocity and acceleration. This property is well-known in robotics and UAV path planning, but here it takes on new importance because abrupt forces in acoustic levitation are critical, causing immediate trap loss. In this light, one could argue that splines are almost a natural fit for levitation-based displays.

However, if splines are so effective, why has their use in acoustic levitation been limited so far? The answer lies partly in the difficulty of coupling splines with the nonlinear physics of the Gor'kov potential, which has not been widely attempted before. While Gor'kov provides highly accurate identification of acoustic forces, its computation is extremely slow, particularly when repeated many times during trajectory optimisation. This makes this model accurate but not as useful in practice. For some applications such latency would be unacceptable. This suggests that future work should not only focus on new optimisation strategies, but also on developing faster approximations of the Gor'kov potential, perhaps using reduced-order physics or machine learning–based algorithms.

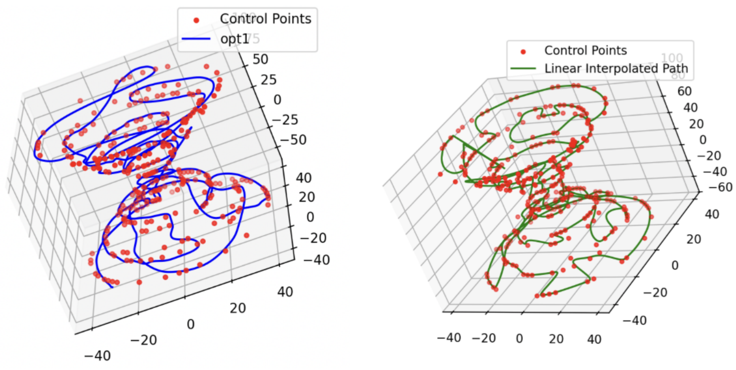


Figure 8: OptiSpline(left) vs. Linear Interpolation(right) paths on a double cylinder

Another important observation is that OptiSpline does not always outperform linear interpolation once computation time is factored in. For simple, nearly straight trajectories, linear methods remain competitive, since the risk of destabilisation is inherently lower. This nuance matters for practical deployment. A system designer might choose a hybrid approach, using OptiSpline for complex shapes with sharp curvature while falling back to interpolation for trivial cases. So, the true contribution of OptiSpline is not in universally replacing existing methods, but in extending the design space to reliably include more complex, high-curvature paths that were previously difficult to realise.

Shape complexity also emerged as a key factor. OptiSpline performed best on smooth, concave geometries, but highly intricate or acute-angled shapes were not systematically evaluated. Such cases remain a limitation of the current approach, and addressing them will be critical for future work. For instance, rendering letters, logos, or detailed symbols in mid-air requires handling of sharper turns and multiple connected components, which are not yet robustly supported. Successfully tackling these scenarios could open the door to richer volumetric displays, more expressive haptic interactions, and precise particle manipulation in scientific contexts.

These findings indicate that OptiSpline makes an important step toward practical trajectory design for acoustic levitation. Its current strengths lie in stability and smoothness, while its main limitations are computational cost and reduced generality for complex geometries. Future improvements in physics-aware optimisation and real-time Gor'kov modelling could make OptiSpline not only a useful research tool but also a foundation for next-generation levitation applications.

7 Conclusion

In this work, we presented the OptiSpline trajectory optimisation method for acoustic levitation, demonstrating that it generally outperforms linear interpolation in maintaining particle stability and producing smooth motion. B-splines were particularly effective in generating continuous trajectories that limit abrupt changes in velocity and acceleration. While the Gorkov potential remains the most accurate tool for determining quantity of a sound field, its high computational cost limits its practicality for iterative or real-time optimisation. OptiSpline performs best for smooth, concave trajectories, but its effectiveness for more complex shapes with multiple sections or acute angles has not yet been thoroughly tested, representing a current limitation of the method. Future work will focus on extending OptiSpline to handle these challenging geometries and further improving computational efficiency, making it a more robust solution for practical levitation applications.

References

[Chaudhuri(2021)] Arindam Chaudhuri. B-splines. arXiv preprint arXiv:2108.06617, 2021. Published in *Encyclopedia of Computer Graphics and Games*.

[Farin(2001)] Gerald Farin. *Curves and Surfaces for CAGD: A Practical Guide*. Morgan Kaufmann, San Francisco, CA, 5th edition, 2001.

[Gao(2024)] Lei Gao. Stablelev: Data-driven stability enhancement for multi-particle acoustic levitation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024. doi: 10.1145/3613904.3642286.

[Paneva(2022)] Viktorija Paneva. Optitrap: Optimal trap trajectories for acoustic levitation displays. *ACM Transactions on Graphics*, 41(5):Article No. 173, 1–14, 2022. doi: 10.1145/3517746.

[Zhu and Liu(2019)] Yuanpeng Zhu and Zhuo Liu. A class of trigonometric bernstein-type basis functions with four shape parameters. *Mathematical Problems in Engineering*, 2019. Figure comparing TB-type ("TB-like") curves with classical cubic Bézier curves.