

Title TBD

Anonymous authors

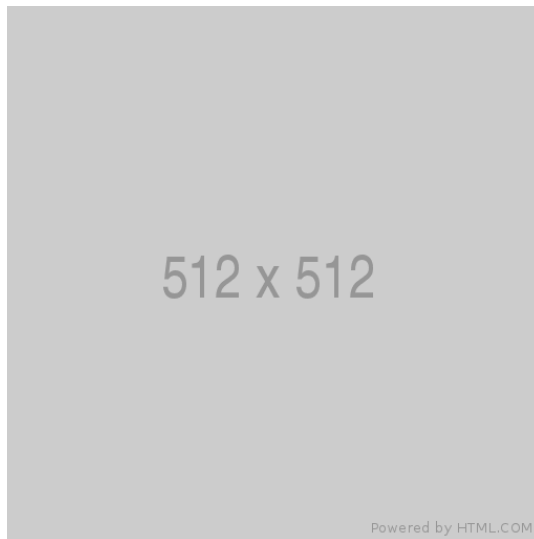


Figure 1: *this is the teaser*

Abstract

This is the ABSTRACT

CCS Concepts

• **Computing methodologies** → Collision detection; • **Hardware** → Sensors and actuators; PCB design and layout;

1. Introduction

THIS IS FROM SIGGRAPH PAPER.

Non-linear constraint solving is fundamental to graphics and scientific computing, with applications ranging from collision detection and minimal distance computation to element inversion and Boolean operations. A vast literature addresses this topic (Section ??). For example, [AMHH*18, AMHH*24] summarize methods for static collision detection between proxies, referencing over 100 algorithms tailored to different primitive pairs and accuracy/efficiency trade-offs. Similar per-primitive-pair specialization is required for minimal distance queries and, likewise, each finite element (FE) type and order necessitates custom code for positive Jacobian checks. This complexity doubles when considering time-dependent scenarios.

While real-time applications often restrict primitives to boxes due to limited computational resources, high-fidelity simulations

may require conservative, high-accuracy predicates [Sny92]. Testing the correctness and ensuring the efficient, accurate implementation of these algorithms is a major challenge [Wang et al. 2021]. The difficulty of generalizing theoretical improvements across different cases hinders progress in this pervasive and crucial family of algorithms, essential to modern computing.

In contrast to algorithm specialization, snyder92 proposed a general framework, based on interval analysis, for conservative solutions to high-order constrained optimization. This framework offers two algorithms: SOLVE, which finds all solutions to a non-linear constraint system, and MINIMIZE, which finds the constrained global minimum of a function. For SOLVE, the conservative algorithm returns a region guaranteed to contain all solutions (if any), potentially including points near the feasible domain. For MINIMIZE, it returns a value less than or equal to the true minimum

and within a bounded distance of it. In both cases, this conservativeness accounts for numerical rounding errors.

Although often considered slower than methods like Newton's minimization, recent work [WFS*21, CYN*24] demonstrates the effectiveness and relative efficiency of this conservative approach, particularly when seeking guaranteed solutions.

Snyder's approach uses Natural Interval Extensions (NIE) to compute *inclusion functions* (??) that bound function ranges over domains, by composition of interval operators (??). Although general, NIE's convergence to the true range via domain decomposition can be slow. For the common case of polynomials, tighter bounds are achievable via their Bézier representation [LKB20, Sta95, JRG13]. However, Bézier representation can be computationally expensive for polynomials with many terms.

We employ a hybrid approach, blending Bézier inclusion functions and NIE. Decomposing polynomial expressions into simpler forms (fewer variables or lower degree) allows us to construct a spectrum of inclusion functions that ranges from fully NIE-based (expanded expressions) to fully Bézier-based (collapsed expressions). Hybrid solutions can dramatically improve efficiency. For example, our hybrid solver for continuous collision detection between high-order polynomial patches is orders of magnitude faster than purely NIE-based and purely Bézier-based solutions (??).

We developed MiSo on top of such a hybrid approach. MiSo is a Python-based domain-specific language (DSL) for the specification of SOLVE and MINIMIZE problems. MiSo enables the user to quickly explore possible hybrid approaches by changing a few lines of code. From a simple specification, the MiSo compiler produces a numerically robust C++ solver for the given problem, automatically generating all the necessary representations of the functions involved, the related transformations required for domain subdivision, and the evaluation of inclusion functions.

Domain decomposition and interval arithmetic are used to guarantee conservative results. Setting a compile-time flag switches to a faster, non-conservative computation mode based on standard floating-point arithmetic. A known limitation of subdivision-based methods is that they suffer from a curse of dimensionality; hence, our method may become impractical for problems in many dimensions. However, we show that we are able to achieve competitive performance for a number of fundamental geometric problems, especially those involving high-order geometry.

We demonstrate competitive performance against hand-optimized code for key computer graphics problems, including linear and high-order continuous collision detection, and finite element validity checks.

MiSo is available as an open-source project at <https://gitlab.com/fsichetti/miso>.

References

- [AMHH*24] AKENINE-MÖLLER T., HAINES E., HOFFMAN N., PESCE A., IWANICKI M., HILLAIRES S.: Real-time rendering - ray tracing resources page, 2024. URL: <https://www.realtimerendering.com/intersections.html>. 1
- [CYN*24] CHEN X., YU C., NI X., CHU M., WANG B., CHEN B.: A time-dependent inclusion-based method for continuous collision detection between parametric surfaces. *ACM SIGGRAPH Computer Graphics* 43, 6 (2024). 2
- [JRG13] JOHNEN A., REMACLE J.-F., GEUZAIN C.: Geometrical validity of curvilinear finite elements. *Journal of Computational Physics* 233 (2013), 359–372. 2
- [LKB20] LENGAGNE S., KALAWOUN R., BOUCHON F., MEZOUAR Y.: Reducing pessimism in interval analysis using b-splines properties: Application to robotics. *Reliable Computing* 27 (2020), 63–87. 2
- [Sny92] SNYDER J.: Interval analysis for computer graphics. *ACM SIGGRAPH* (1992), 121–130. 1
- [Sta95] STAHL V.: *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations*. PhD thesis, Sept. 1995. 2
- [WFS*21] WANG B., FERGUSON Z., SCHNEIDER T., JIANG X., ATTENE M., PANOZZO D.: A large-scale benchmark and an inclusion-based algorithm for continuous collision detection. *ACM Trans. Graph.* 40, 5 (2021), 1–16. 2
- [AMHH*18] AKENINE-MÖLLER T., HAINES E., HOFFMAN N., PESCE A., IWANICKI M., HILLAIRES S.: *Real-Time Rendering 4th Edition*. A.K. Peters/CRC Press, Boca Raton, FL, USA, 2018. 1