SOFTWARE & ACCESSORIES

Next-gen optical design software

Multisequential, object-based architecture for modeling optical systems improves sequential optical design, analysis, optimization, and tolerancing capabilities with one program.

Akash Arora

With the advent of personal computers, nearly all engineering disciplines benefited from computer-aided design (CAD) software—and the field of optics is no exception. Optical design workflow was revolutionized via programs capable of doing advanced analysis, optimization, and tolerancing.

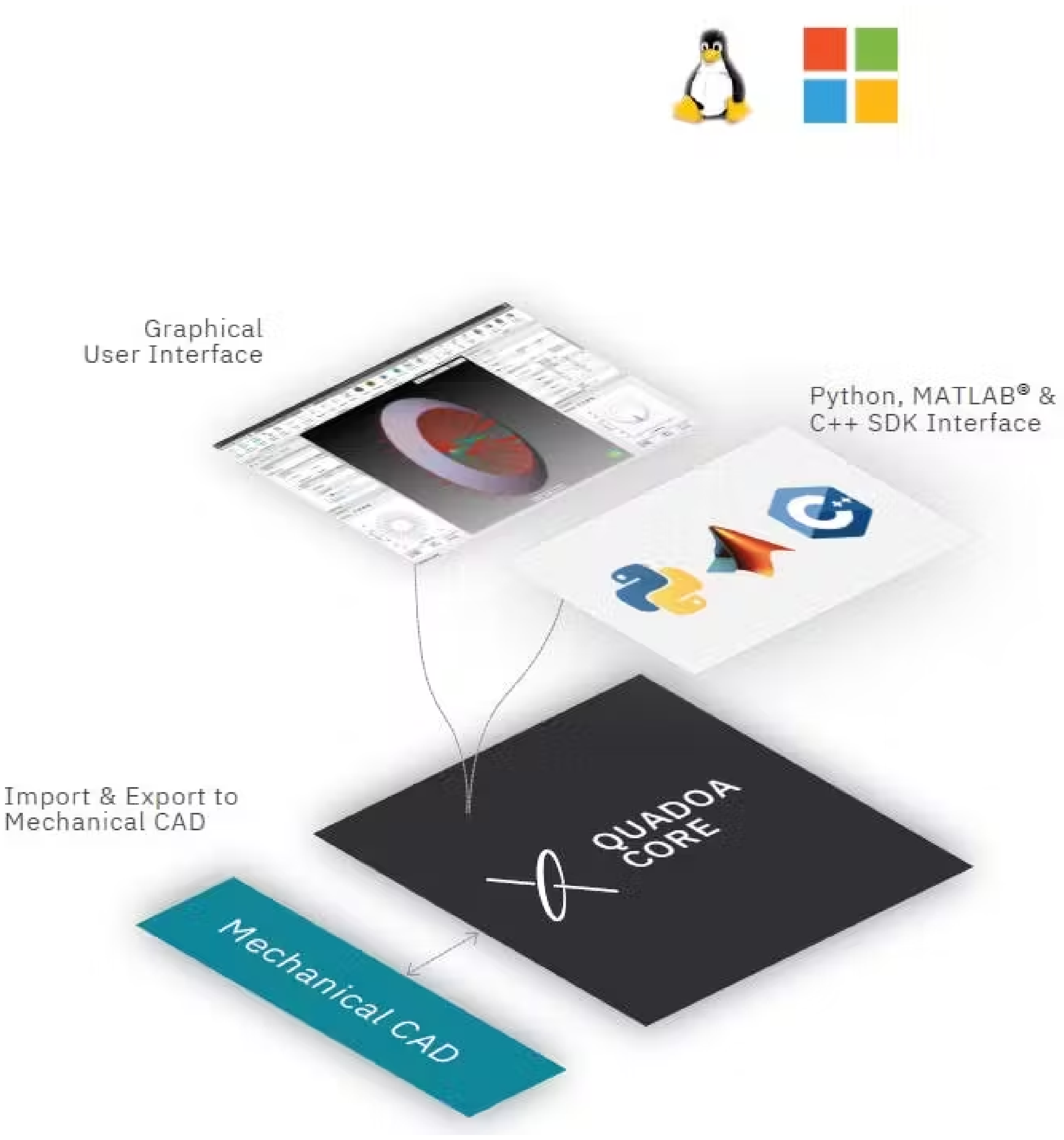
Many optical design programs that originated during the 20th century use a surface spreadsheet listing to define optical components and ray paths. This method was understandable due to the sequential nature of light propagation through traditional optical systems, such as cameras and telescopes.

Eventually, the sequential listing of optical surfaces became the paradigm for optical design programs. As applications with nonsequential light propagation became more common—including lighting and stray light—nonsequential optical design programs were developed to simulate light propagating through components in any order. These programs typically used object-based models that organize components into three-dimensional (3D) objects and assemblies, like mechanical CAD programs.

While nonsequential simulation is more general, sequential simulation is more efficient for a wide range of optical design problems. These legacy optical design programs, whose foundations were written long ago, continue to evolve within the limitations of their fundamental architectures. If you were to develop a modern optical design program, what would its key features be?

Modern programming

For starters, any modern program should exploit the latest software development languages and tools for its architecture. This means using efficient, robust, highly functional, pre-written libraries where available, and directly integrating these with efficient optical physics calculations.



The Quadoa software architecture.

Many powerful user interface (UI), graphics, and plotting libraries are now available that didn't exist decades ago when the most widely used software was written. Legacy optical design programs often use a software translation layer to interface older code with modern libraries rather than rewriting them. Many frustrating instabilities and inefficiencies of these programs are based upon this fact. A better method would be to write code from the ground up to exploit this modern software architecture. This comprehensive approach would prevent the need for architecture tradeoffs necessary to incorporate legacy code. Another powerful advantage is comprehensive and efficient access to the core algorithms via an API. It would even be possible to integrate the core algorithms into other software or equipment, completely independent of the user-interface.

Object-based architecture

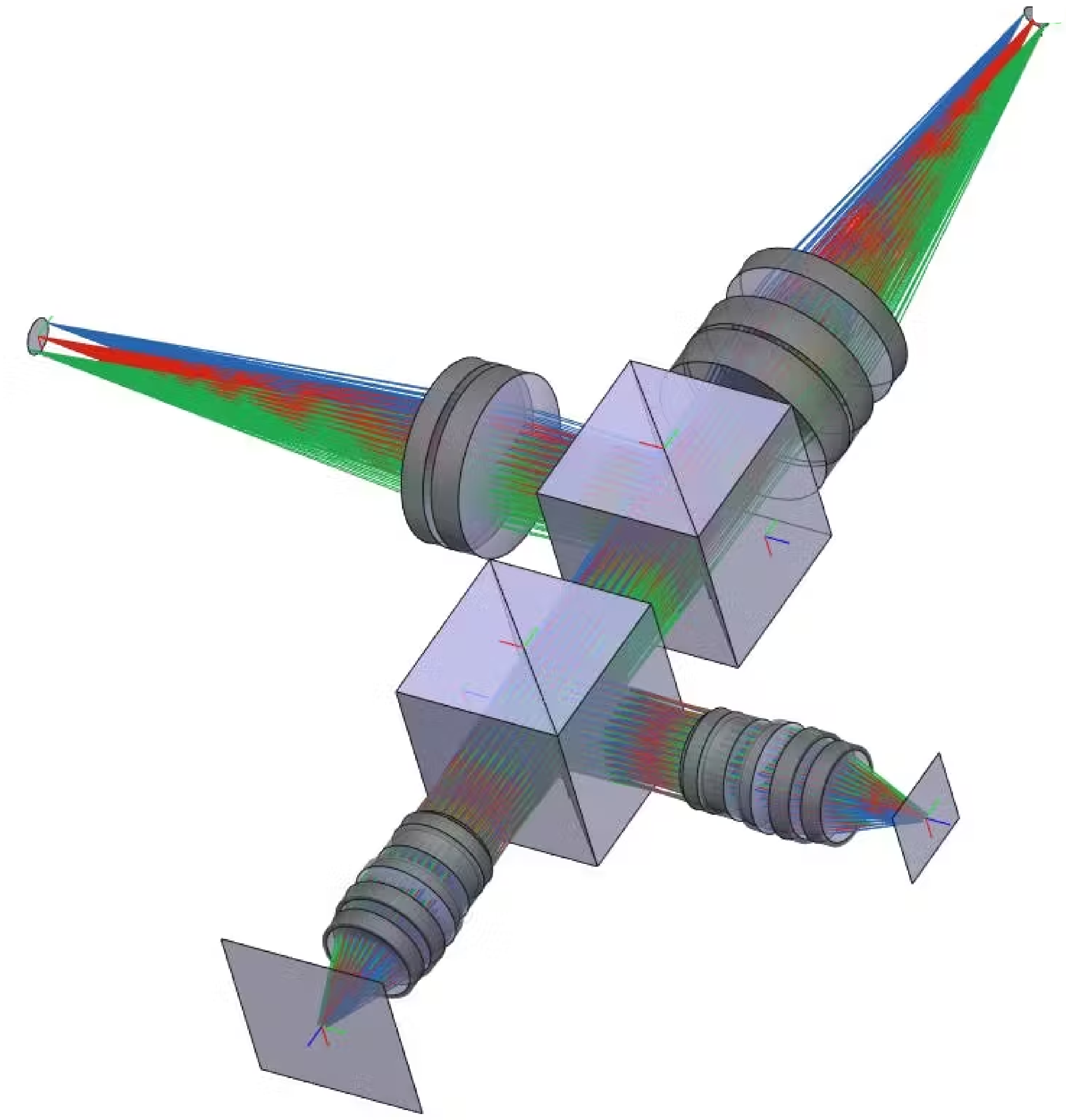
With the surface listing methodology of legacy optical design programs, significant complexity is needed to model non-rotationally symmetric systems or component dependencies. The fundamental rule in this methodology is that all surfaces are positioned relative to the prior surface. Positioning a surface globally or referenced to an assembly requires numerous coordinate-shifting surfaces and complicated solves.

An object-based architecture is a better method to model optical components, and its primary advantage is the intuitive definition of the system that most closely matches the nature of the real optical components. This enables 3D components to be easily defined in global coordinates or relative to others. They can be grouped into assemblies and subassemblies that support pivots and shifts about arbitrary points.

When implemented with modern graphical libraries, modifying component parameters can be done via a 3D viewer or a more traditional object editor. The benefits of this model extend to tolerancing and simulating complex tolerance dependencies. Mechanical CAD components can be readily incorporated into such an optical model.

Multisequential ray tracing

The sequential paradigm for optical simulation is based on valid reasoning, because many optical systems have well-defined ray paths and many optical physics calculations require aperture stops and unique reference rays to properly compute. Sequential ray tracing is the fastest and most efficient way to simulate these systems. It is important to be able to define a sequence of surfaces through which light traces. But using the same surface listing to define the optical model and the ray sequence necessitates complexity in all but the simplest systems. While the real optical system only has one set of components, the surface list optical model must duplicate and ignore surfaces using multiple configurations to simulate multiple optical paths.



Interferometer with multiple optical paths.

Multiple configurations are a powerful modeling tool that should be reserved for components moving or changing shape. A better method is to define the optical components and separately define any number of ray sequences that interact with these surfaces, decoupling the optical component definition and ray sequence definition. This multisequential paradigm has clear advantages over the current sequential paradigm—it ensures the model defining the optical components is as simple as possible, while supporting any number of discrete ray paths through these optics.

Double-pass systems only require optical components to be defined once. The separately defined ray sequence dictates multiple interactions with the same surfaces. In simpler systems, the program automatically predicts the sequence based upon the component listing order. Sequential analyses can be set to evaluate any defined sequential ray path.

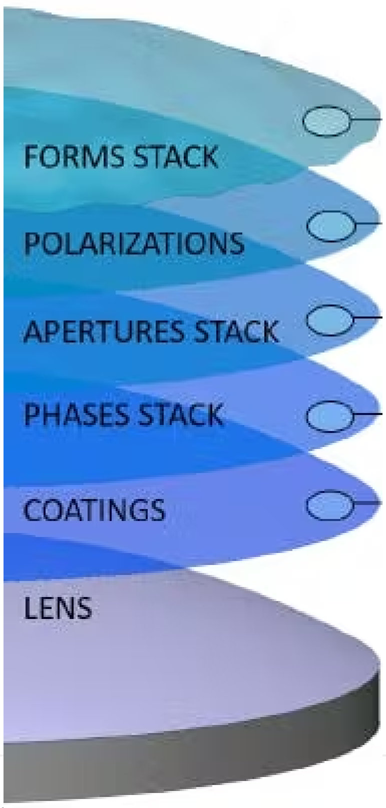
Unified nonsequential ray tracing

While sequential ray tracing is efficient, many optical systems include complex light sources that cannot be accurately defined as field points, or ray paths that aren't known in advance. These systems benefit from the more general non-sequential ray tracing methods available. It is often necessary to run sequential and non-sequential simulations on the same optical system. With legacy optical design programs, these are either distinct modes of operation—including at the level of component definition—or completely separate programs.

There are tools for converting a system from one mode/program to the other, but having two distinct models of the same system is inefficient and error-prone. A better method for a fundamentally object-based optical model is to support predefined ray sequences and more general non-sequential ray tracing—all in the same model. Some analyses require sequential ray tracing, while others support non-sequential ray tracing. This would be a valuable capability for an optical system such as a digital projector with optics needing to be simultaneously optimized for illumination and imaging targets.

Surface definition evolution

Optical components come in many shapes and with varying characteristics. On the simpler side, there are rotationally symmetric, spherical lenses and mirrors. Other components are more complex, such as freeform optics, lenslet arrays, or light modulators (such as polarization, position, and angle). With so many types of optical components, coding distinct surface types with the exact combination of needed parameters is problematic.



Source URL: https://www.laserfocusworld.com/software-accessories/article/14304881/next-gen-optical-design-software

Surface stack layer definition.

Legacy optical design programs have used this method and developed complicated workarounds, such as combining adjacent surfaces, for modeling complex surface types. A better method is to define a base surface and allow various parametric layers to be superimposed. These layers can comprise any combination of form/sag, aperture/obscuration, polarization, phase, coating, and replication/arrays. This will make the fundamental model simpler for traditional optics and vastly more extensible for novel optics.

This new paradigm will be simpler, more intuitive, more extensible, and more powerful than legacy optical design software. Quadoa Optical CAD software was written with these characteristics from the ground up. It has fully featured sequential optical design, analysis, optimization and tolerancing capabilities. We offer modules that interface to C++, MATLAB, and Python, enable wave optics simulation, and integrate mechanical CAD. And non-sequential simulation is under active development. The software can run on Windows or Linux natively, and on MacOS via a virtual machine. Licensing is perpetual or subscription, with local, network, or cloud options depending upon the user's needs.