Differential ray tracing for an efficient computation of the Jacobian in the damped least-squares algorithm

Tobias Seger[1,*], *Christoph* Menke[1], *Matthias* Sonntag[1], and *Karsten* Urban[2]

[1]Carl Zeiss AG, 73446 Oberkochen, Germany
[2]Institute of Numerical Mathematics, Ulm University, 89081 Ulm, Germany

Abstract. We present a framework for the optimization of optical systems based up on the damped least-squares method and differential ray tracing. First, a mathematical description for the differentiation of the ray-surface intersection is introduced. It is observed that the structure of the optimization variables and operands can be exploited such that the Jacobian for the damped least-squares algorithm can be computed in the same computational complexity as the primal ray tracing for both forward and reverse mode algorithmic differentiation. To demonstrate the advantages of the method a freeform mirror system is analyzed.

1 Introduction

For many years the damped least-squares (DLS) algorithm has been the method of choice for the optimization of optical systems. DLS requires the evaluation of the Jacobian of the optimization operands which is usually done by finite differences. Despite the simplicity of the finite difference method it has some major drawbacks, namely the need for many function evaluations and its limited stability and precision. As an alternative algorithmic differentiation (AD) [1] has been used in many disciplines including lens design [2], where it is often referred to as differential ray tracing and mainly used in the context of end-to-end design [3]. The basic idea of AD is to describe the computation of the optimization operands in terms of a computational graph that can be differentiated using the chain rule. Depending on the direction in which the chain rule is applied the method is called AD forward mode or AD reverse mode.

In this contribution, we present a method to efficiently compute the Jacobian using AD in both forward and reverse mode. This allows us to use pseudo Newton methods such as DLS instead of first-order gradient based methods for optimization. A mathematical analysis for the differentiation of the ray-surface intersection enables an improvement of the performance. This is demonstrated for a freeform design with many optimization parameters since these systems are known to be particularly challenging [4].

2 Differential ray tracing

Ray tracing as a part of geometrical optics describes the propagation of light through an optical system from the object to the image surface. The task in differential ray tracing is to compute the derivative of some raydata with respect to the variable parameters of the optimization, e.g., surface curvatures or thicknesses. Many of the ray tracing operations are straightforward to differentiate using basic differentiation rules. Particularly expensive is the computation of the intersection point of a ray with an optical surface and the derivative of this operation.

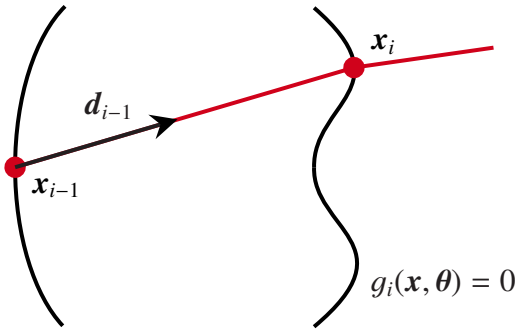

Figure 1. Intersection of a ray with an optical surface

A ray at surface i is given by its intersection point $\boldsymbol{x}_{i-1}$ with the previous surface (index $i-1$) and its direction $\boldsymbol{d}_{i-1}$. The optical surface is implicitly described by a nonlinear function g_i depending on the spatial coordinate $\boldsymbol{x}$ and the variable parameters $\boldsymbol{\theta}$ of the optimization. Now the task is to find the geometrical path length t_i such that

$$g_i(\boldsymbol{x}_{i-1} + t_i\, \boldsymbol{d}_{i-1}, \boldsymbol{\theta}) = 0 \tag{1}$$

as it is illustrated in Figure 1. For the special case of a spherical surface the problem (1) can be solved analytically. In general the solution has to be determined numerically, e.g., by a one-dimensional Newton method. However, differentiating the Newton iteration to obtain the required derivatives is expensive, memory intensive and inefficient [5]. A solution for this problem that is well known in the AD community is the implicit function theorem [6]. We apply this concept to optical design in a general framework. First, a solution of the nonlinear equation is computed without considering any derivatives. The derivatives $\partial t_i/\partial \boldsymbol{\theta}$, $\partial t_i/\partial \boldsymbol{x}_{i-1}$, $\partial t_i/\partial \boldsymbol{d}_{i-1}$ exist locally by the implicit function theorem if the ray direction and the surface normal are not orthogonal at the intersection point.

*e-mail: tobias.seger@zeiss.com

The procedure will be demonstrated for the AD forward mode, where the tangent of the geometrical path length can be computed by the implicit function theorem

$$\dot{t}_i := \frac{\mathrm{d}t_i}{\mathrm{d}\boldsymbol{\theta}}\,\dot{\boldsymbol{\theta}} = -\frac{(\boldsymbol{\nabla}_{\boldsymbol{x}} g_i)^\top(\dot{\boldsymbol{x}}_{i-1} + t_i\,\dot{\boldsymbol{d}}_{i-1}) + (\boldsymbol{\nabla}_{\boldsymbol{\theta}} g_i)^\top\dot{\boldsymbol{\theta}}}{(\boldsymbol{\nabla}_{\boldsymbol{x}} g_i)^\top \boldsymbol{d}_{i-1}}, \quad (2)$$

where $\boldsymbol{\nabla}_{\boldsymbol{x}}$ denotes the spatial gradient of a function and $\boldsymbol{\nabla}_{\boldsymbol{\theta}}$ the derivative with respect to the variable parameters. By a proper vectorization over the variable parameters by array programming it is achieved that each surface only needs to be evaluated once for each ray to obtain all the derivatives. In contrast to this, the use of finite differences requires $n + 1$ (n is the number of parameters) evaluations of each surface for a single ray, which is particularly expensive for surfaces of high order as are frequently used in freeform systems.

3 Efficient Jacobian computation

Let m denote the number of operands in the merit function of the optical system and n the number of variable parameters. Then, the computational cost for the evaluation of all operands is $O(m \cdot n)$. Computing the Jacobian with finite differences yields computational cost of $O(m \cdot n^2)$. The AD forward mode performs Jacobian-vector products, i.e., the differential ray tracer has to be invoked with unit vectors $\mathbf{e}_k \in \mathbb{R}^n$ for $k = 1, \ldots, n$, each computing a column of the Jacobian, resulting in a complexity of $O(m \cdot n^2)$. On the other hand, the reverse mode performs vector-Jacobian products, i.e., the differential ray tracer receives $\mathbf{e}_j \in \mathbb{R}^m$ for $j = 1, \ldots, m$ to compute the Jacobian row-wise with a complexity of $O(m^2 \cdot n)$. The aim of our work is to eliminate the quadratic dependencies as far as possible.

Optical surfaces are often described as conic sections and a term describing the deviation from it as a linear combination of basis polynomials. A key insight for an efficient differential ray tracer is that the derivative of the second term with respect to a coefficient is simply the corresponding basis polynomial, which allows a very fast computation. The efficiency of the forward mode relies on pure local effects of the variable parameters, meaning that the inner product $(\boldsymbol{\nabla}_{\boldsymbol{\theta}} g_i)^\top\dot{\boldsymbol{\theta}}$ in (2) is always performed with a unit vector or zero. For the reverse mode, note that each operand only depends on very few rays (e.g., for transverse aberrations on two rays). Only these few rays need to be considered for the vector-Jacobian product and consequently the computational cost for computing a row of the Jacobian reduces from $O(m \cdot n)$ to $O(n)$.

4 Results

The developed methods are tested for an unobscured three-mirror anastigmat, where the deviation of the mirror surfaces from a conic section is described by Zernike polynomials. Here, the merit function controls transverse aberration, distortion and some boundary conditions ensure that the reflected rays do not intersect. Figure 2 shows a cross section of an optimized system.

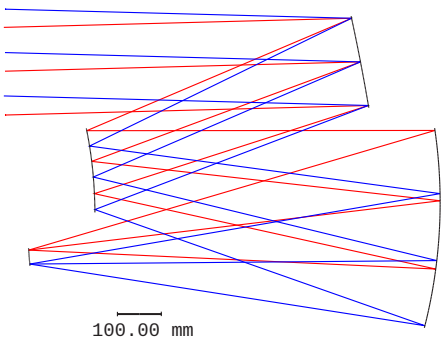

Figure 2. Layout of the freeform three-mirror anastigmat

For the numerical experiments the number of operands is fixed and the polynomial order of the surfaces is gradually increased from third order up to order 20. As can be seen in Figure 3 the finite differences behave as expected with a runtime that increases quadratically with the number of parameters. On the other hand, both reverse and forward mode only have linear runtime. This significant reduction of runtime facilitates the DLS optimization of optical systems, which are challenging due to their long computing times.

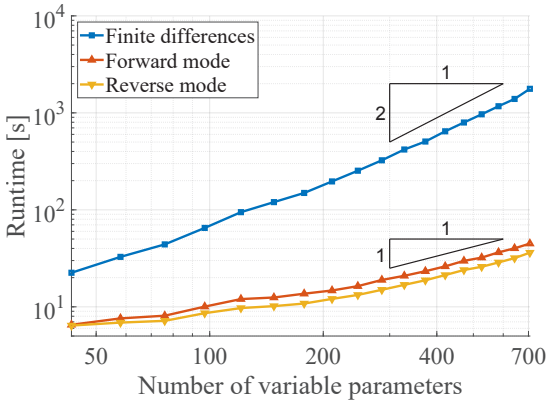

Figure 3. Runtime of 10 DLS iterations for the three-mirror anastigmat on a single CPU. The Jacobian was either computed using finite differences or algorithmic differentiation.

References

[1] A. Griewank, A. Walther, *Evaluating Derivatives*, 2nd edn. (Society for Industrial and Applied Mathematics, 2008)
[2] J. Werner, M. Hillenbrand, A. Hoffmann, S. Sinzinger, Schedae Informaticae **21**, 169 (2012)
[3] Z. Li, Q. Hou, Z. Wang, F. Tan, J. Liu, W. Zhang, Optics Letters **46**, 5453 (2021)
[4] Y. Nie, J. Zhang, R. Su, H. Ottevaere, Optics Express **31**, 7450 (2023)
[5] C. Wang, N. Chen, W. Heidrich, IEEE Transactions on Computational Imaging **8**, 905 (2022)
[6] C.C. Margossian, WIREs Data Mining and Knowledge Discovery **9**, e1305 (2019)