Schedae Informaticae Vol. 21 (2012): 169–175
doi: 10.4467/20838476SI.12.011.0821

Automatic Differentiation in the Optimization of Imaging Optical Systems

Jürgen Werner[1], Matthias Hillenbrand[1], Armin Hoffmann[2], Stefan Sinzinger[1]
[1]Fachgebiet Technische Optik, Institut für Mikro- und Nanotechnologien, Technische Universität Ilmenau, PF 10 05 65, 98684 Ilmenau, Germany
[2]Institut für Mathematik, Technische Universität Ilmenau, PF 10 05 65, 98684 Ilmenau, Germany
e-mail: *juergen.werner@tu-ilmenau.de*

Abstract. Automatic differentiation is an often superior alternative to numerical differentiation that is yet unregarded for calculating derivatives in the optimization of imaging optical systems. We show that it is between 8% and 34% faster than numerical differentiation with central difference when optimizing various optical systems.

Keywords: automatic differentiation, optimization, lens design.

1. Introduction

The goal in lens design is to create an optical system for a specific application with optimum image quality while meeting constraints such as costs or dimensions. In the first step, the designer selects a starting system that approximately performs the imaging task. The number, type and position of optical surfaces are determined in this step.

The next step is to choose the remaining free parameters, such as radii and distances between surfaces, to maximize the image quality, which is measured by a function $f : X \to \mathbb{R}^+$ mapping parameters $x \in X$ to a nonnegative real value combining aberrations and deviations from desired system properties [1]. The minimization of the merit function f is a problem of mathematical optimization [2]. In lens design, mainly the Levenberg-Marquardt [4] algorithm is used. This and similar algorithms require at least first order derivatives of the merit function [2]. How to efficiently calculate these derivatives is the main topic of this manuscript. With freeform surfaces becoming more popular, speed again becomes an issue due to the heavily increased number of variables.

In general there are several ways to calculate derivatives. Consider the function

$$f(x, y) = (3x + xy)^2. \tag{1}$$

Symbolic differentiation directly manipulates this formula to produce formulas for the partial derivatives, e.g.

$$\frac{\partial}{\partial x} f(x, y) = 2(3x + xy)(3 + y). \tag{2}$$

Unfortunately merit functions cannot usually be expressed in an algebraic form like Eq. (1). They often contain components based on ray tracing, e.g. the root mean square (rms) spot size. For various surfaces like aspheres, the intersection point of a ray with a surface cannot be calculated directly. Instead, Newton's method is used. Therefore an algebraic representation of the merit function is not readily available and symbolic differentiation is not a viable option. The commonly used alternative is numerical differentiation [2, 3], e.g. by central differences

$$\frac{\partial}{\partial x} f(x, y) \approx \frac{f(x + h, y) - f(x - h, y)}{2h}. \tag{3}$$

Numerical differentiation is easy to implement but of course only yields approximations of the derivatives.

There is yet another method for calculating derivatives that is especially suitable for functions evaluated by a computer program: automatic differentiation [5, 6, 7]. It usually yields exact values for derivatives (to machine accuracy) [7]. The application of automatic differentiation in lens design has not been published previously.

2. Automatic differentiation

Consider again Eq. (1). The so called *forward mode* of automatic differentiation does *not* produce symbolic representations of the derivatives like Eq. (2) but calculates the values of the derivatives at only one specific point (x, y) during execution of the computer program evaluating f at point (x, y). To do so, for each program variable, the values of its derivatives with respect to x and y are calculated along with its value. The usual rules of differentiation like product rule and chain rule are applied to each atomic step of the computer program. Table 1 shows a program to evaluate formula (1) and the extension to also calculate the derivatives. $t_6 = f(x, y)$ is the desired function value and $\nabla t = (\frac{\partial}{\partial x} t, \frac{\partial}{\partial y} t)$ is the gradient vector. Columns in gray show the values during program execution when started with $x = 5$ and $y = 7$.

Table 1.: Program to evaluate formula (1) and its derivatives. Notation borrowed from Rall and Corliss [8]

$t_1 = x$	$= 5$	$\nabla t_1 = (1, 0)$	$= (1, 0)$
$t_2 = y$	$= 7$	$\nabla t_2 = (0, 1)$	$= (0, 1)$
$t_3 = 3t_1$	$= 15$	$\nabla t_3 = 3\nabla t_1$	$= (3, 0)$
$t_4 = t_1 t_2$	$= 35$	$\nabla t_4 = t_1 \nabla t_2 + t_2 \nabla t_1$	$= (7, 5)$
$t_5 = t_3 + t_4$	$= 50$	$\nabla t_5 = \nabla t_3 + \nabla t_4$	$= (10, 5)$
$t_6 = (t_5)^2$	$= 2500$	$\nabla t_6 = 2t_5 \nabla t_5$	$= (1000, 500)$

There are different tools available that create the extensions necessary for calculating derivatives. A detailed and excellent introduction to automatic differentiation is given by Rall and Corliss [8].

3. Evaluation

We compare the performance of automatic differentiation to numerical differentiation when optimizing various optical systems. We implemented our own framework for optimizing optical systems [9], which includes ray tracing and a merit function based on the rms spot size. Optimization algorithms used are the Levenberg-Marquardt (LM) implementation from C/C++ Minpack [10] and SLSQP from NLopt [11]. Derivatives are calculated by automatic differentiation (AD) with the tapeless forward mode of ADOL-C [12] and by numerical differentiation with forward difference (ND1) $\frac{\partial}{\partial x} f(x, y) \approx \frac{f(x+h,y)-f(x,y)}{h}$ and with central difference (ND2) as in Eq. (3). We choose $h = \max\{\sqrt{\epsilon_m} \cdot |x|, \sqrt{\epsilon_m}\} \cdot \operatorname{sgn} x$ with ϵ_m being the machine accuracy, which is $2^{-52} \approx 2.2 \cdot 10^{-16}$ for the data type *double*. This choice of h conforms to the recommendation by Press et al. [13]. Correctness of computations is verified on various levels [9]. Four optical systems are used for evaluation: An apochromat-like system with three cemented glasses and small field; a Double Gauss lens; a compact photographic lens from a U.S. patent; and a lens for a compact point-and-shoot camera [14] containing two aspherical surfaces. The systems have 5, 8, 11 and 14 parameters set as variables for optimization, respectively. The complete system parameters and definitions of the merit functions are listed in the appendix.

Table 2 shows the solutions found by the optimization algorithms in combination with the different modes of differentiation. In addition, it shows the solutions found by the commercial optical design program Zemax.

When forward difference is used, the algorithms often fail to converge to the proper local minimum. Therefore forward difference is not suitable for optimization of optical systems.

The Point & Shoot lens is a rare example where central difference with the Levenberg-Marquardt algorithm fails. In general we found that, using LM, numerical differentiation with central difference works as well as automatic differentiation. We do not know which method of differentiation Zemax uses. It may be possible that a different LM implementation also finds the proper local minimum using central difference.

To ensure that the data indeed only reflects the effects of differentiation and is not influenced by starting points lying in the fractal region of the solution space [15], the starting points have been randomly perturbed at relative scales of 10^{-9} to 10^{-4}. All optimizations with LM starting from the new points led to the same local minima.

Table 2.: Merit function values after optimization

		Apochromat	Double Gauss	U.S. 4223982	Point & Shoot
LM	ND1	$7.60394 \cdot 10^{-4}$	$1.04366 \cdot 10^{-2}$	$9.77262 \cdot 10^{-3}$	$1.00915 \cdot 10^{-2}$
	ND2	$6.70740 \cdot 10^{-4}$	$1.04316 \cdot 10^{-2}$	$9.76228 \cdot 10^{-3}$	$1.25302 \cdot 10^{-2}$
	AD	$6.70740 \cdot 10^{-4}$	$1.04316 \cdot 10^{-2}$	$9.76229 \cdot 10^{-3}$	$6.31551 \cdot 10^{-3}$
SLSQP	ND1	$5.07001 \cdot 10^{-3}$	$1.21951 \cdot 10^{-2}$	$9.81915 \cdot 10^{-3}$	$1.32293 \cdot 10^{-2}$
	ND2	$9.94167 \cdot 10^{-4}$	$1.04316 \cdot 10^{-2}$	$9.76227 \cdot 10^{-3}$	$1.16099 \cdot 10^{-2}$
	AD	$6.70740 \cdot 10^{-4}$	$1.04316 \cdot 10^{-2}$	$9.76227 \cdot 10^{-3}$	$6.31551 \cdot 10^{-3}$
Zemax		$6.70741 \cdot 10^{-4}$	$1.04316 \cdot 10^{-2}$	$9.76234 \cdot 10^{-3}$	$6.31553 \cdot 10^{-3}$

To explain the previous results, the accuracy of numerical differentiation is shown in Tab. 3. Derivatives of the merit function are calculated with respect to each variable at three different configurations of each system. Accuracy is defined as the minimum of the absolute and the relative difference between the value obtained by numerical differentiation and the value obtained by automatic differentiation. Relative difference of a and b is calculated as $\frac{|a-b|}{\max\{|a|,|b|\}}$. The worst cases are the derivatives with respect to the first radius for the Apochromat and the 6th order aspheric term for the Point & Shoot lens. The combinations that have the worst *worst cases* (Apochromat with ND1, Point & Shoot with ND1 and ND2) are also the combinations where optimization with LM fails.

Table 3.: Accuracy of the derivatives of the merit function

	Forward Diff.		Central Diff.	
	Worst	Median	Worst	Median
Apochromat	$2 \cdot 10^{-1}$	$7 \cdot 10^{-5}$	$2 \cdot 10^{-4}$	$1 \cdot 10^{-9}$
Double Gauss	$9 \cdot 10^{-4}$	$6 \cdot 10^{-7}$	$5 \cdot 10^{-8}$	$6 \cdot 10^{-10}$
U.S. 4223982	$1 \cdot 10^{-3}$	$1 \cdot 10^{-6}$	$4 \cdot 10^{-8}$	$2 \cdot 10^{-9}$
Point & Shoot	$5 \cdot 10^{-1}$	$1 \cdot 10^{-5}$	$8 \cdot 10^{-2}$	$3 \cdot 10^{-9}$

Table 4 shows the relative speed of the different modes of differentiation, i.e. the number of gradients of the merit function calculated per second by the respective method, divided by the number of gradients calculated by central difference. Automatic differentiation is consistently faster than numerical differentiation with central difference.

Table 4.: Relative speed of differentiation

	ND1	ND2	AD
Apochromat	1.82	1.00	1.14
Double Gauss	1.90	1.00	1.30
U.S. 4223982	1.92	1.00	1.34
Point & Shoot	1.93	1.00	1.08

4. Conclusions

When optimizing optical systems, numerical differentiation with forward difference is often not a viable option due to lack of accuracy. Automatic differentiation is more accurate and between 8% and 34% faster than numerical differentiation with central difference for the four optical systems analyzed. Therefore automatic differentiation can replace numerical differentiation in the optimization of optical systems with the respective gain in speed.

Acknowledgment. The authors would like to thank the German ministries TMBWK and BMBF for the financial support through the programs “Graduate Research School on Optical Microsystems Technology” (FKZ: PE104-1-1) and “Kompetenzdreieck Optische Mikrosysteme” (FKZ: 16SV5473).

5. References

[1] Shannon R.R.; *The Art and Science of Optical Design*, Cambridge University Press, 1997.

[2] Gross H. (ed.); *Aberration Theory and Correction of Optical Systems*, *Handbook of Optical Systems* vol. 3, Wiley-VCH, 2007.

[3] Vasiljevic D.; *Classical and Evolutionary Algorithms in the Optimization of Optical Systems*, Kluwer Academic Publishers, 2002.

[4] Kidger M.J.; *Use of the Levenberg-Marquardt (damped least-squares) optimization method in lens design*, Optical Engineering 32(8), 1993, pp. 1731–1739.

[5] Wengert R.; *A simple automatic derivative evaluation program*, Communications of the ACM 7(8), 1964, pp. 463–464.

[6] Rall L.B.; *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science 120, Springer 1981.

[7] Griewank A., Walther A.; *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, 2nd edition, 2008.

[8] Rall L.B., Corliss G.F.; *An introduction to automatic differentiation*, Computational Differentiation: Techniques, Applications, and Tools, SIAM, 1996, pp. 1–17.

[9] Werner J.; *Optimization of imaging optical systems.* Bachelors thesis, Ilmenau University of Technology, 2011.

[10] Devernay F.; *C/C++ Minpack.*

[11] Johnson S.G.; *The NLopt nonlinear-optimization package.*

[12] Walther A., Griewank A.; *ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++*, 2010.

[13] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.; *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007.

[14] Kidger M.J.; *Intermediate Optical Design*, SPIE Publications, 2004.

[15] van Turnhout M., Bociort F.; *Instabilities and fractal basins of attraction in optical system optimization*, Optics Express 17(1), 2009, pp. 314–328.

A. Systems

Table 5 shows the details of the systems used for evaluation. Variables set for optimization are printed in bold. Lens data for U.S. 4223982 is taken from example 1 of the patent. The stop has been placed at a distance of 2 mm from the fourth surface. Glasses have been replaced by SF53, LAF13, N-SK5, SF53, LAK10 and LAK13, respectively. The radius of the last surface is automatically chosen to achieve a focal length of 100 mm. All other radii and the distance to the image plane are set as variables for optimization. The fourth system is the *lens for compact point-and-shoot camera* from Kidger [14]. All radii and (pre-existing) conic constants and aspheric coefficients are set as variables for optimization.

For all systems the merit function is generated by the default merit function functionality of Zemax using the following parameters: rms spot radius centroid; Gaussian quadrature with 3 rings (7 rings for Point & Shoot) and 6 arms; and an overall weight of 1. For the Apochromat and the Double Gauss lens the Zemax operand EFFL with parameters wave 2, target 100 and weight 1 is added to the merit function to achieve a focal length of 100 mm. For the Point & Shoot lens the Zemax operand EFFL with parameters wave 2, target 35 and weight 1 is added.

Table 5.: System specifications. Angles are given in degree, wavelengths in μm and all other lengths in mm.

(a) Apochromat: lens data.

	Radius	Thickness	Glass
OBJ	Infinity	0.0000	
STO	**60.6053**	2.0000	F2
2	**-38.2839**	2.0000	KZFSN5
3	**28.5790**	2.0000	N-FK51A
4	**-74.2175**	**98.3228**	
IMA	Infinity	0.0000	

(b) Double Gauss: lens data.

	Radius	Thickness	Glass
OBJ	Infinity	0.0000	
1	**54.1532**	8.7467	SK2
2	**152.5219**	0.5000	
3	**35.9506**	14.0000	SK16
4	Infinity	3.7770	F5
5	**22.2699**	14.2531	
STO	Infinity	12.4281	
7	**-25.6850**	3.7770	F5
8	Infinity	10.8339	SK16
9	**-36.9802**	0.5000	
10	Infinity	6.8582	SK16
11	**-67.1476**	**57.3145**	
IMA	Infinity	0.0000	

(c) Apochromat: other properties.

Aperture	10; Entrance Pupil Diameter
Field	Angle; Radial
	$(0,0),(0,0.7),(0,1)$
Wavelengths	0.4861327, 0.5875618, 0.6562725

(d) Double Gauss: other properties.

Aperture	30; Entrance Pupil Diameter
Field	Angle; Radial
	$(0,0),(0,10),(0,14)$
Wavelengths	0.4861, 0.5876, 0.6563

(e) U.S. 4223982: other properties.

Aperture	Float By Stop Size
	Semi-Diameter 8.198
Field	Angle; Radial
	$(0,0),(0,16.8),(0,24)$
Wavelengths	0.4861, 0.5876, 0.6563

(f) Point & Shoot: other properties.

Aperture	Float By Stop Size
	Semi-Diameter 3.355
Field	Angle; Radial
	$(0,0),(0,18),(0,25),(0,31)$
Wavelengths	0.4861, 0.5876, 0.6563